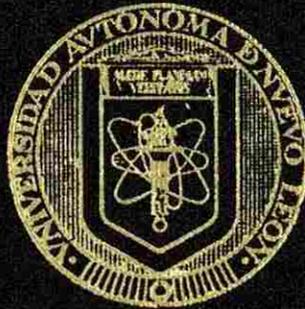


UNIVERSIDAD AUTONOMA DE NUEVO LEON

**FACULTAD DE INGENIERIA MECANICA
Y ELECTRICA
DIVISION DE ESTUDIOS DE POST-GRADO**



**PROYECTO PARA UN PROGRAMA DE
CAPACITACION PARA MAESTROS DE SISTEMAS
OPERATIVOS I DE LA FACULTAD DE INGENIERIA
MECANICA Y ELECTRICA**

POR

LIC. ARIADNE BEATRIZ SANCHEZ RUIZ

TESIS

**EN OPCION AL GRADO DE MAESTRO EN CIENCIAS
DE LA ADMINISTRACION CON ESPECIALIDAD
EN RELACIONES INDUSTRIALES**

CIUDAD UNIVERSITARIA

DICIEMBRE DE 1998

PROYECTO PARA UN PROGRAMA DE
CAPACITACION PARA MAESTROS DE SISTEMAS
OPERATIVOS DE LA FACULTAD DE INGENIERIA
MECANICA Y ELECTRICA

TM

Z5853

.M2

FIME

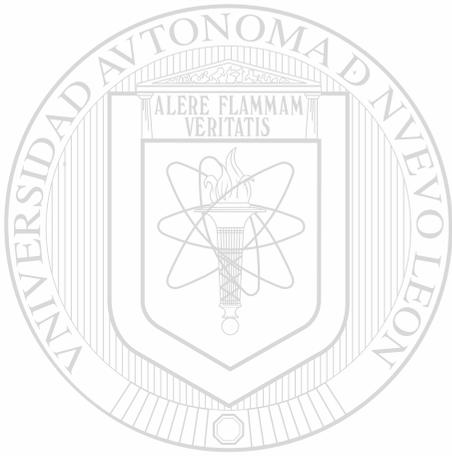
1998

S262

1998



1020124834



UANL

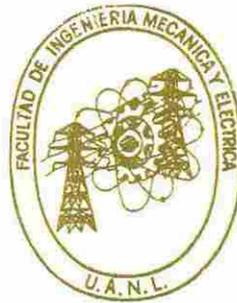
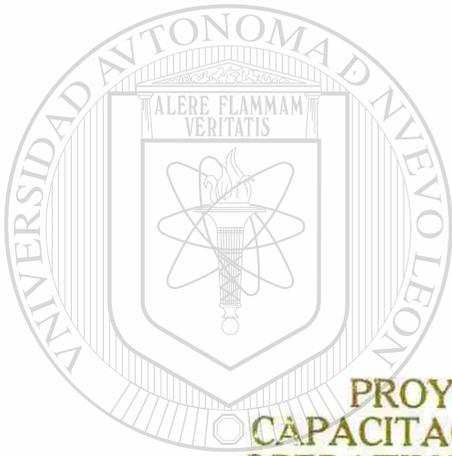
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE INGENIERIA MECANICA
Y ELECTRICA
DIVISION DE ESTUDIOS DE POST-GRADO



PROYECTO PARA UN PROGRAMA DE
CAPACITACION PARA MAESTROS DE SISTEMAS
OPERATIVOS I DE LA FACULTAD DE INGENIERIA
MECANICA Y ELECTRICA

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS
POR

LIC. ARIADNE BEATRIZ SANCHEZ RUIZ

TESIS

EN OPCION AL GRADO DE MAESTRO EN CIENCIAS
DE LA ADMINISTRACION CON ESPECIALIDAD
EN RELACIONES INDUSTRIALES



CIUDAD UNIVERSITARIA

DICIEMBRE DE 1998

TM
25853
M2
FINE
1998
5262

0129-68360



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

®

DIRECCIÓN GENERAL DE BIBLIOTECAS

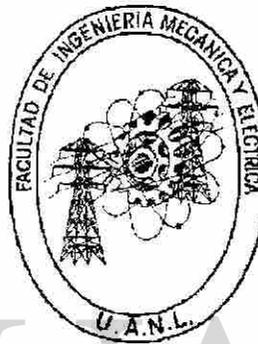


FONDO
TESIS

UNIVERSIDAD AUTONOMA DE NUEVO LEON

FACULTAD DE INGENIERIA MECANICA Y ELECTRICA

DIVISION DE ESTUDIOS DE POST-GRADO



PROYECTO PARA UN PROGRAMA DE CAPACITACION PARA MAESTROS
DE SISTEMAS OPERATIVOS I DE LA FACULTAD DE INGENIERIA
MECANICA Y ELECTRICA.

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
POR

DIRECCIÓN LIC. ARIADNE BEATRIZ SANCHEZ RUIZ

TESIS

EN OPCION AL GRADO DE MAESTRO EN CIENCIAS DE LA
ADMINISTRACION CON ESPECIALIDAD EN RELACIONES INDUSTRIALES

CIUDAD UNIVERSITARIA

DICIEMBRE DE 1998

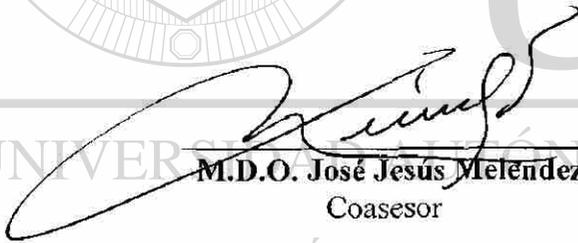
UNIVERSIDAD AUTONOMA DE NUEVO LEON
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA
DIVISION DE ESTUDIOS DE POST-GRADO

Los miembros del comité de tesis recomendamos que la tesis **“PROYECTO PARA UN PROGRAMA DE CAPACITACION PARA MAESTROS DE SISTEMAS OPERATIVOS I DE LA FACULTAD DE INGENIERIA MECANICA Y ELECTRICA”**, realizada por la Lic. Ariadne Beatriz Sánchez Ruíz, sea aceptada para su defensa como opción al grado de Maestro en Ciencias de la Administración con especialidad en Relaciones Industriales.

El Comité de Tesis



M.A. Matías A. Botello Treviño



M.D.O. José Jesús Meléndez Olivas
Coasesor



M.C. David Garza Garza
Coasesor

DIRECCIÓN GENERAL DE BIBLIOTECAS

No. Bo.


M.C. Roberto Villarreal Garza
División de Estudios de Post-grado

San Nicolás de los Garza, N.L., a Diciembre de 1998

Dedicatorias:

- A Dios sobre todas las cosas
 - A los seres del mundo celeste
 - A mi madre que se encuentra entre ellos
 - A mi padre y mis hermanos
 - A mis sobrinos
 - A Paco el amor de mi vida
 - A mis maestros los que partieron y a los que aun están en donde los conocí
 - A mis amigos los de ayer y los de siempre, los que se fueron a donde ya no puedo verlos y a los que aun puedo comunicarme con ellos
 - Al maestro Matías que me alentó a cursar la maestría y a preparar este trabajo para graduarme
 - A la maestra Olga Rodríguez que me abrió sus puertas para que tomara de su conocimiento y de su experiencia
-
- Al Ing. Antonio González quien me proporcionó la primer beca para estudiar la maestría
 - Al Ing. Jesús Guzmán quien me ha ayudado en lo que ha estado a su alcance

Agradecimientos:

- Al Ing. Cástulo Vela que me ha apoyado en mis proyectos
- Al Departamento Escolar que me proporcionó las cifras que necesitaba para hacer la proyección de este trabajo

- A la Secretaría Académica que me brindó su ayuda en cuanto a información y que conjuntamente con la Facultad nos dio a mí a otros maestros más la oportunidad de cursar un Diplomado en Docencia Universitaria.
- A mis asesores por tener paciencia para conmigo y dedicarme un poco de su tiempo
- A mis alumnos porque este trabajo ha sido pensando precisamente en ellos
- Un agradecimiento especial a todos aquellos que contribuyeron para que tuviera el espacio y el tiempo para hacer este trabajo



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

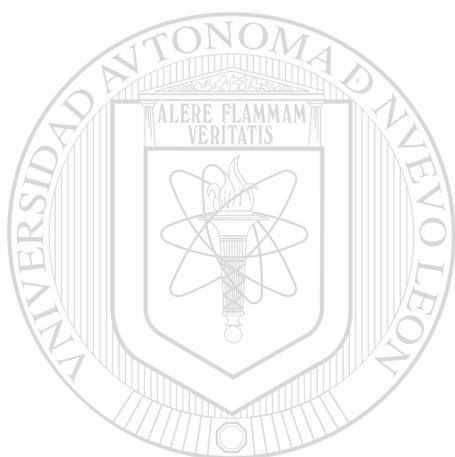


DIRECCIÓN GENERAL DE BIBLIOTECAS

Indice

Prólogo.....	7
Síntesis.....	10
1. Introducción.....	14
1.1. Antecedentes.....	15
1.2. Objetivo de la Tesis.....	17
1.3. Justificación.....	17
1.4. Planteamiento.....	19
1.5. Hipótesis.....	30
1.6. Metodología.....	30
2. Análisis de Información Bibliográfica.....	31
3. Diseño del Programa.....	35
3.1. Programa del Plan de Estudios o del Sistema Educativo (Programa Sintético).....	37
3.2. Programa de la Institución o de la Academia de Maestros (Programa Analítico).....	38
3.3. Programa del Docente (Programa Guía).....	47
3.4. Programa para el Profesor.....	61
3.5. Programa para el Alumno.....	65
3.6. Planeación Didáctica.....	68
4. Recursos Materiales.....	76
5. Conclusiones.....	77
6. Recomendaciones.....	79
Bibliografía.....	81
Listado de Tablas.....	83
Listado de Gráficas.....	86
Glosario de Términos.....	92
Anexo1 Apuntes.....	95

Anexo 2 Técnicas de Dinámicas de Grupo.....	237
Anexo 3 Configuración Mínima del Equipo Computacional.....	240
Anexo 4 Documentos.....	241
Anexo 5 Tipos de Evaluación.....	262
Resumen Autobiográfico.....	265



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

Prólogo.-

Si analizamos el pasado de la computación veremos al matemático inglés Charles Babbage con todo su entusiasmo en la manufactura de su "máquina analógica" ayudado por Lady Ada Augusta Lovelace quien escribió el primer programa que habría de funcionar en esta máquina, un sueño de siglos atrás, con el inconveniente de la tecnología que en su época no le fue de utilidad para producir las piezas con la precisión que se requería. Sin embargo, en este siglo XX, alrededor de 1940, en los albores de una guerra en la que la necesidad de hacer cálculos de grandes cantidades y de distribuir alimentos, ropa, armamento, municiones, etc, se echó mano de aquel viejo invento de Babbage y de una forma rudimentaria (con relación a nuestros días), se dio origen a las primeras computadoras funcionando primero como gigantescas máquinas calculadoras usando tubos al vacío, las cuales pesaban toneladas y ocupaban salas enormes. Se alimentaba la información mediante switch mecánicos, usando lenguaje maquinal. Era un trabajo descomunal, sin embargo la guerra dio origen a muchos descubrimientos y a la par con nuestras computadoras, los métodos de optimización nacieron y se utilizaron, por supuesto, estos han sido utilizados para beneficio de la humanidad.

En la posguerra, nace el primer sistema operativo, primitivo pero eficiente, simplificaba mucho la labor de las personas abocadas a las computadoras. Siempre han ido evolucionando juntos el hardware y los sistemas operativos; con la introducción de los transistores que permitieron sustancialmente aumentar la confiabilidad de las computadoras, se marcó la pauta para que fuese factible la construcción de máquinas comerciales. Otro movimiento que se originó fue la separación de funciones en diseñadores, fabricantes o constructores y programadores.

Todo esto ha significado progreso. Con la evolución de computadoras y aplicaciones (hardware y software) poco a poco se fueron explorando nuevos horizontes, algunas áreas consideradas solo para ser aplicadas o desarrolladas por la mano del hombre ahora se han visto beneficiadas al tener un tinte de exactitud, detalle y rapidez.

De ser una calculadora dantesca a usar la computadora con fines administrativos, posteriormente hemos visto reflejadas sus bondades en el área médica al hacer uso de ellas como herramienta para el diagnóstico con la automatización de imágenes como el TAC (tomografía axial computarizada), la resonancia nuclear magnética, la angiografía digital computarizada por mencionar algunas, además de pequeñas computadoras de propósito específico como lo son, los marcapasos o la implantación de cóclea; los estudiantes pueden desarrollar proyectos científicos y preparar reportes usando tecnología computacional; los maestros están interesados en la computadora como herramienta interactiva para el aprendizaje; los programas de educación asistida por computadora, programas interactivos de autoaprendizaje pueden enseñar (si el estudiante resuelve correctamente los problemas el programa pasa al siguiente tema y de lo contrario lo hace repasar el área en la que mostró deficiencia); los científicos las utilizan para desarrollar teorías, recolectar y probar datos, para intercambiar electrónicamente información con otros colegas del mundo; en la arquitectura tenemos el diseño asistido por computadora; en las fábricas se usan robots para la manufactura asistida por computadora y podemos seguir mencionado diversas aplicaciones, cine, radio, arqueología, música, hogar, a llegar a ser indispensables en la vida cotidiana de ciertas áreas. Con las comunicaciones alejadas geográficamente y el uso del ahora polifacético Internet con todas sus contribuciones para la educación, comercio, administración o simplemente la cultura general.

Sin embargo algo que ha estado presente desde su creación en 1951 y que ha ido evolucionando conforme la tecnología es el sistema operativo, el administrador de recursos, el encargado de mostrarnos la máquina virtual con la cual interactuamos. Ese sistema operativo que en ocasiones actúa, bajo nuestro punto de vista, impredecible. El sistema operativo que con sus razonamientos heurísticos desafía nuestra intuición. De hecho, nadie, ningún ojo humano ha podido penetrar ese mundo que se esconde en ese gabinete al que solemos llamar, erróneamente, CPU.

Uno de los problemas que encierra esta asignatura es precisamente la dificultad para explicar el funcionamiento interno de cada componente, como interactúan uno con otro y con el medio ambiente externo, sabemos teóricamente que así debe ser, pero no se dispone de algo tangible con que demostrarlo.

Mediante este trabajo se pretende tratar de "aterrizar" el contenido de esta materia, esto es, que el alumno tenga la base conceptual teórica pero que disponga del material necesario para comprender al sistema operativo y prepararlo para otros enfoques y nuevas asignaturas con mayor grado de complejidad y que le serán útiles en el ejercicio de su profesión y todo ello se debe dar en el contexto de un proceso enseñanza-aprendizaje adecuado, hilvanando cuidadosamente tarea, tema, técnica y dinámica sin perder de vista nunca nuestros objetivos.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

Síntesis.-

Este trabajo cubre algunos aspectos de capacitación al maestro orientada a que el educando reciba, de una manera sencilla y práctica, información que hasta ahora ha sido catalogada como teórica. Para ello se han conjugado diferentes corrientes didácticas y teorías de aprendizaje.

Este programa propone un orden diferente del establecido actualmente. Como plataforma define todos los conceptos, dentro de los sistemas operativos, elevando gradualmente el grado de dificultad de cada término e hilvanando poco a poco cada uno de ellos hasta llegar a formular un todo, que es lo que nosotros como usuarios manejamos, es decir, la máquina virtual con la que nosotros creemos trabajar y la que trabaja en realidad.

De una manera subjetiva, hacemos un viaje hacia el interior de una computadora y analizamos cada componente, su funcionamiento, las reglas con las que se rige cada uno de ellos, sus interconexiones internas, su interfase con el medio ambiente interno, externo, y finalmente lo que se aprecia como resultado de cada acción que tomamos en un sistema.

Esta perspectiva es un tanto complicada con el actual programa, es por ello que se plantea la reforma en cuanto a orden de temas, material didáctico y un compendio bibliográfico de varios autores especialistas en el ramo, esto es con el fin de no cerrarnos a una sola opinión manteniendo actualizados los tópicos que se tratan en esta asignatura, requieren una renovación constante aunque los conceptos en sí han prevalecido en el tiempo.

Desde otro ángulo, se puede decir que los contenidos registrados actualmente que definen esta materia se formularon, tal vez, de forma arbitraria con un carácter orientativo y, en ningún caso, deben tomarse al pie de la letra. En algunas ocasiones, incluir en primera instancia de concreción el apartado de contenidos en los currículums puede ser un error, ya que es el apartado más irrelevante del diseño y, por la tradición de nuestro modelo educativo, una mala interpretación de su significado en el conjunto del proyecto educativo puede ocasionar serias divergencias respecto a los fines y objetivos de una reforma. La relación de contenidos nunca debe entenderse como una

secuencia de aprendizaje, equívoco que puede tener lugar si nos situamos en el eje cartesiano del clásico modelo educativo, en el cual, lo único que se formulan son los contenidos y la práctica habitual en el aula se limita a transmitir el conocimiento del docente a través de la exposición de aquellos.

De manera general, los diseños curriculares no definen el proceso de aprendizaje del alumno y el de esta materia no es la excepción. Por lo tanto, a falta de otro documento que lo avale, es nuestro deber realizarlo a partir de una determinada concepción, de algunos elementos básicos y con unos criterios que respeten el espíritu innovador de esta reforma que se pretende.

Como elementos básicos del modelo de diseño curricular que se propone adoptar están: el contenido organizador y la estructura de los contenidos.

El contenido organizador es el primer elemento que hay que definir o seleccionar y tiene la misión de concretar las intenciones educativas, esto es, enuncia, a grandes rasgos, el proceso de aprendizaje que se ha de llevar a cabo.

El contenido organizador puede ser de dos tipos: de procedimiento y conceptual. El primero se utiliza en procesos formativos donde lo fundamental es el aprendizaje de modos o maneras, o sea, de "saber hacer", es decir, cuando lo que se desea es el desarrollo de habilidades o destrezas. Sin

embargo, el contenido organizador de tipo conceptual se utiliza en los procesos que podemos calificar como informativos, en los cuales lo que se persigue es la adquisición del conocimiento, esto es, "del saber".

La estructura de los contenidos es un esquema que desarrolla gráficamente el contenido organizador, quedando recogidos en ella todos sus aspectos. La estructura de los contenidos admite niveles de detalle. Sería necesario elaborar una estructura de carácter general que englobe todos los contenidos que se abordarán durante el proceso de aprendizaje y otras más detalladas de cada una de las unidades en que se divida. Este elemento es clave para establecer una secuencia ordenada de los contenidos que permita el desarrollo progresivo de las capacidades.

Existen tres tipos de estructuras de los contenidos, dos de ellas corresponden con los procesos formativos en los que el contenido organizador tiene carácter de procedimiento. En este caso hablamos de estructuras de contenido de tipo lineal o con tomas de decisión. Cuando el contenido organizador es conceptual, la estructura adquiere forma de árbol invertido en la que aparecen varios niveles de elaboración o de detalle del contenido.

Las estructuras de tipo lineal expresan relaciones de orden y describen el procedimiento que hay que aprender, indicando todas las etapas que lo constituyen y en el orden que deben ser ejecutadas.

Las estructuras con tomas de decisión establecen varios caminos mediante los cuales se puede llevar a cabo el procedimiento. En cada uno de los caminos han de estar contempladas las principales etapas del procedimiento, pero unos ofrecerán mayor dificultad que otros. El grado de dificultad estará dado por el número de etapas que aparecen en cada camino o por la complejidad de cada una de ellas.

Aunque las dos formas de plantear el aprendizaje del procedimiento son válidas, siempre que sea posible, nos inclinamos por la segunda ya que es la mejor manera de adquirir de forma paulatina todas las capacidades integradas en el procedimiento.

Las estructuras en forma de árbol, típica de los procesos de aprendizaje de corte conceptual, comienzan con el concepto más general e inclusive del que se van ramificando otros más elementales en diferentes niveles.

Cuanto más alejados estén del concepto general, más ramificaciones habrá en el nivel de elaboración correspondiente y más específicos serán los conceptos que en él se ubiquen.

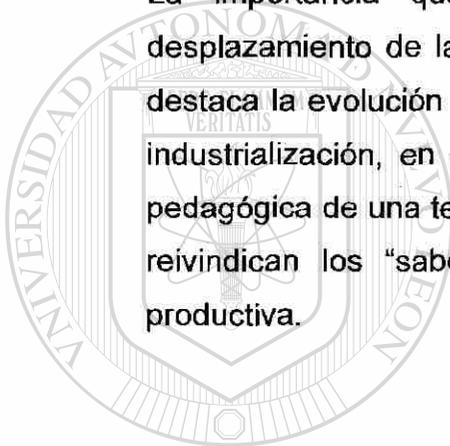
A través de las unidades que constituyen la secuencia deducida de la estructura de los contenidos, los alumnos van desarrollando paulatinamente una serie de capacidades que les permiten acceder a unas etapas formativas más específicas con la formación previa adecuada. En este proceso se desarrollan habilidades o destrezas que perduran y, a la vez, se adquieren los conocimientos que se requieren para llevar a cabo las acciones señaladas en

el apartado de las actividades. Ésta es una manera más eficaz de aprender los contenidos conceptuales.

En este tipo de organización, los "saberes", constituyen los contenidos de soporte del aprendizaje.

Si bien es cierto que los procedimientos aprendidos y las capacidades desarrolladas de esta manera, en estas áreas de especialidad, no tienen una proyección directa en la vida laboral, también lo es que constituyen una cimentación fundamental para el aprendizaje de futuros procesos formativos específicos más ligados a la actividad productiva.

La importancia que realmente tiene lo curricular y el consecuente desplazamiento de la didáctica es producto de varias causas, entre las que destaca la evolución de la escuela, al ritmo de las exigencias del proceso de industrialización, en este caso, la "teoría curricular" constituye la expresión pedagógica de una teoría educativa específica de este proceso y en la que se reivindican los "saberes útiles", los cuales son necesarios para la vida productiva.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

Capítulo 1

Introducción

Esta asignatura es teórica en un 100% dado que se tratan generalidades de los sistemas operativos tales como el manejo de memoria, funcionamiento de hardware, la interacción del sistema operativo como administrador de recursos, temas de los que sabemos como se llevan a cabo pero que difícilmente pueden demostrarse como no sea haciendo alegorías o bien haciendo alusión a algunas “extravagancias”.

Se dispone de la teoría por ello presumimos que sabemos el cómo, pero al no encontrar un medio apropiado con el cual hacer el enlace con la práctica se pierde mucho de lo que en esencia debería ser el proceso enseñanza-aprendizaje.

Se presentaba la situación de que el alumno solía aburrirse o bien tomar en broma algunos detalles, pero siempre se veía en la situación de tomar apuntes y, dado lo extenso del material a cubrir, el maestro debía elegir entre explicar o preocuparse porque los alumnos tuvieran apuntes. Cabe señalar que esta técnica funciona y en efecto, es de provecho para el alumno, pero si buscamos

“algo más”, ése algo más podría ser encaminado a la excelencia educativa. En una clase tecnológica con un material al que no es fácil, en un momento dado, demostrar su existencia y su funcionamiento.

De hecho, no existe una perspectiva que proponga al docente analizar lo que no funciona en clase, lo que aburre o enfada a los alumnos, estudiar conceptual y prácticamente formas de resolverlo, realizar la experimentación y analizar los resultados.

Debemos entender que, el aula es un espacio vivo, es un laboratorio de didáctica; la didáctica se construye en ese trabajo cotidiano – sobre el que después tenemos que reflexionar -, no es sólo un conocimiento práctico como tradicionalmente la han considerado algunos autores. La práctica que reclama

la investigación en el aula requiere ser confrontada en un plano conceptual y reflexivo.

Videos, electrónica y computación están destinados a realizar una segunda gran revolución educativa de la humanidad; la primera la realizó Gutenberg con la imprenta. Una lectura de prácticas escolares en la universidad medieval muestra cómo el libro de texto trastocó de fondo las formas de enseñanza.

Nuestra búsqueda esta fundamentada, tenemos un problema, una población creciente de alumnos candidatos a obtener un título de Ingenieros Administradores de Sistemas a los que no es posible conformar con una teoría y que estos lo tomen como un dogma de fe. La solución esta en nuestras manos y todo depende del empeño que pongamos en el presente y en el futuro en el ejercicio de una digna profesión: la docencia.

1.1. Antecedentes

La carrera de Ingeniero Administrador de Sistemas se fundó en 1975 ante la necesidad de proveer a la industria mano de obra calificada en el ámbito computacional. Esta asignatura, Sistemas Operativos I, ha sufrido varias transformaciones, en cuanto a enfoque, contenido, nombre y bibliografía.

En el primer plan de estudios se le llamó: "Introducción a los Sistemas Operativos", ubicada en el 5º semestre, con frecuencia diaria señalándosele como requisito la materia "Introducción a los Sistemas Computacionales" (ver anexo 4).

En éste tiempo la tecnología "popular", en lo que a sistemas computacionales se refiere, estaba enfocada a sistemas operativos para macrocomputadoras, minicomputadoras y aunque no como se les conoce ahora, las microcomputadoras, por lo que los contenidos temáticos se inclinaban a satisfacer las necesidades de ese entonces.

En el mes de Enero de 1985 se estableció en el programa la siguiente bibliografía:

Autor: Stuart E. Madnick y Jonh J. Donovan

Nombre: Sistemas Operativos

Editorial: Diana

como básica y

Autor: Harry Katzan Jr.

Nombre: Operating Systems

Editorial: Van Nostrand Reinhold Co.

como complementaria.

Al hacerse una revisión de los programas en 1988 se hicieron las consideraciones pertinentes y al ser analizado el material a cubrir, extenso e importante, se dividió formando la base conceptual Sistemas Operativos I ubicándose en el 6º semestre y Sistemas Operativos II en el 7º semestre, la parte aplicativa (ver anexo 4).

En un principio se utilizó el libro de "Introducción a los Sistemas Operativos" del autor Harvey M. Dietel y se siguió el contenido temático como lo marcaba el autor. En la búsqueda de un enfoque, obviamente más actual, en 1995 se toma como texto el libro de "Sistemas Operativos Modernos" cuyo autor es Andrew S. Tanenbaum y nuevamente se sigue el orden del contenido que marca el autor. Este autor es bastante minucioso en sus descripciones, sin embargo, es tanta su minuciosidad que llega uno a perder de vista el punto desde el que se partió, por supuesto que cuenta con el mérito de haber participado personalmente en la construcción, diseño e implementación de al menos tres sistemas operativos por lo que es de valor incalculable su opinión a través de sus publicaciones y libros, como también lo es la contribución de Dietel. Sin embargo, existen otras personas doctas en la materia y se deben analizar cada una de las posturas de los diferentes autores.

Los sistemas operativos, en sí, son parte fundamental de toda computadora, constituyen una parte integral del ambiente de cómputo, por lo tanto, necesitan ser comprendidos en alguna medida por lo que ayer y hoy sigue vigente en la red curricular de los Ingenieros Administradores de Sistemas.

1.2. Objetivo de la Tesis.-

- Facilitar el trabajo del maestro y del alumno.
- Convertir teoría en práctica.
- El alumno, como producto final, tendrá visión amplia acerca del funcionamiento interno de la computadora y la interacción de Hardware-Sistema Operativo-Software-Firmware

1.3. Justificación.-

Este trabajo ha sido orientado a capacitar a maestros para impartir la materia de Sistemas Operativos I, dado que en esta asignatura se cubre la parte conceptual de otras que serán cursadas posteriormente en el curriculum de la carrera de Ingeniero Administrador de Sistemas, se debe estar consciente de la trascendencia que origina el que el alumno comprenda los tópicos que aquí se manejen. Debemos puntualizar estructuras y conceptos básicos fundamentales para el desarrollo de temas con mayor grado de dificultad y por tanto se buscan los mecanismos para lograr el mayor aprovechamiento.

Es interesante que el alumno conozca, comprenda, y maneje los conceptos que el maestro explique, que se dispongan los mecanismos para que este proceso, enseñanza-aprendizaje, se lleve a cabo en una atmósfera en la que le resulten familiares todos los términos, por ejemplo: cuando explicamos el acceso a un disco, no es lo mismo dar una explicación al aire que tener a mano un hard disk abierto (originalmente se empaacan al vacío) en el cual se aprecian cada uno de los platos, cabezas, eje y brazo mecánico implicados en una operación tan simple y tan común.

Tal vez hayamos caído en la forma en que operan algunos "museos prácticos", como el "Centro Cultural Alfa" o el "Papalote, museo del niño", sin embargo, si realmente captamos la idea de universidad, de ciencia, de tecnología, debemos considerar que hay temas que por su complejidad deben ser tratados de la manera más simple.

Creo que en cuanto a conocimiento no deberíamos decir "estos son métodos para niños y estos son métodos para universitarios" sino buscar el método con

el cual el maestro puede proyectar su conocimiento y que el alumno pueda capitalizar lo que se le está brindando.

El aprendizaje es un proceso caracterizado por saltos, avances, retrocesos, parálisis, miedos, pausas y construcciones, podemos decir que el aprendizaje es un proceso dinámico. En una misma escuela no hay dos grupos que tengan la misma dinámica, que respondan igual a las mismas actividades de aprendizaje.

La relación maestro-alumno está mediada por el conocimiento del profesor y su postura frente a este conocimiento, razón por la que los discípulos lo siguen.

Siendo nuestro compromiso enseñar y el del alumno aprender el proceso debe ser la conjunción de las dos, enseñanza-aprendizaje. Aquí la idea es que el alumno disponga de todo aquello que sea tangible o proyectable a su alcance.

El movimiento de una mano no puede ser interpretado únicamente como un cambio de conducta observable; se requiere estudiar el contexto y la situación en la que se originó para entenderlo como una conducta total del ser humano.

La fragmentación de la información hace difícil que el estudiante construya una estructura conceptual que le permita tener elementos de confrontación y articulación entre una estructura de ideas, conceptos y procedimientos que ha aprendido – y por lo tanto construido previamente – y la nueva información con la que tiene que interactuar.

No podemos comprender que el aprendizaje se da de manera aislada o fragmentaria; por lo que consideramos que no se puede sostener que la memorización sea un aprendizaje. Si un objeto de estudio se ha construido a partir de una realidad que necesariamente guarda un grado de integración y estructuración, la o las disciplinas tienen una estructura de integración entre sus conceptos, principios y procedimientos.

Aquí la reflexión que cabe es: ¿ Educar para llenar cabezas o para que cada sujeto desarrolle sus propias ideas?

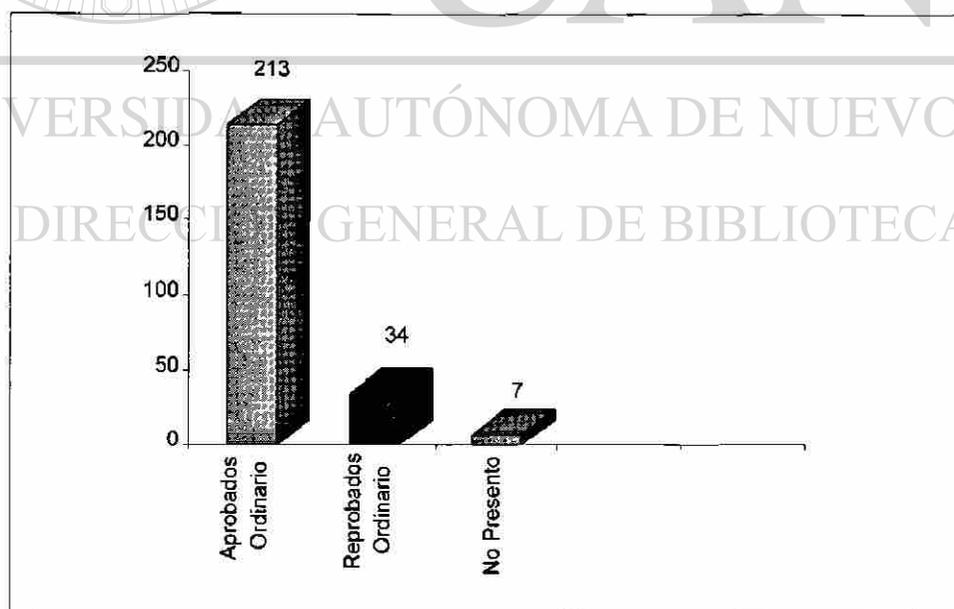
No debemos olvidar que es nuestro desarrollo social el que nos permite pensar como pensamos.

1.4. Planteamiento.-

La problemática inicial es la dificultad implícita del aprovechamiento de los conceptos que se manejan en este tipo de asignaturas y por consecuencia el alto índice reprobatorio. Como se puede observar a través de las tabulaciones y las gráficas que fueron proporcionadas por el Departamento de Servicios Escolares de la Facultad de Ingeniería Mecánica y Eléctrica.

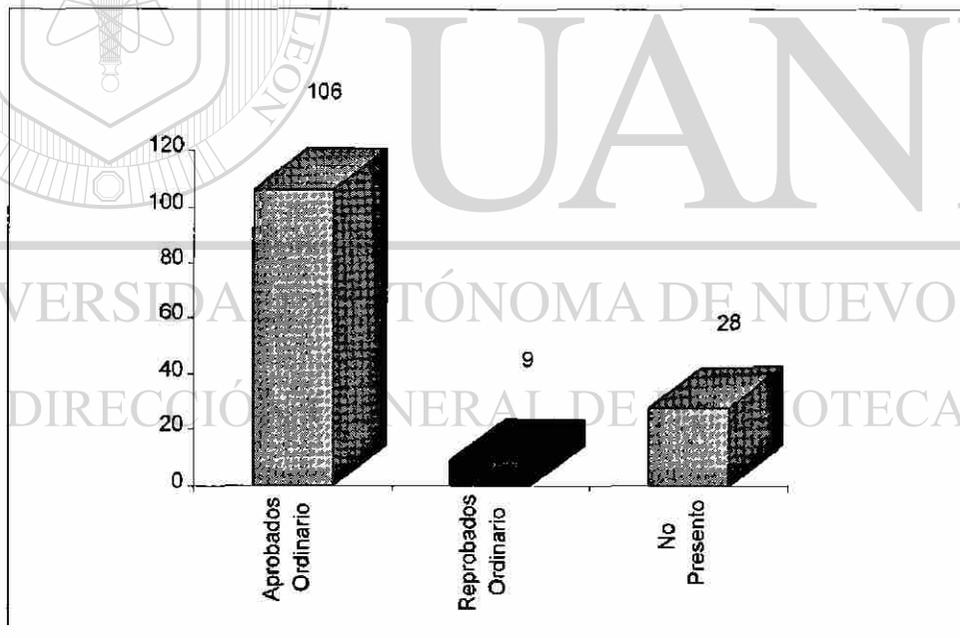
Semestre Febrero Julio 1993

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presente
M1	1	36	33	2	1
M1	2	31	31	0	0
M2	3	40	32	7	1
N2	4	45	42	3	0
N3	5	42	40	1	1
N6	6	24	23	0	1
N6	7	36	12	21	3
Totales		254	213	34	7



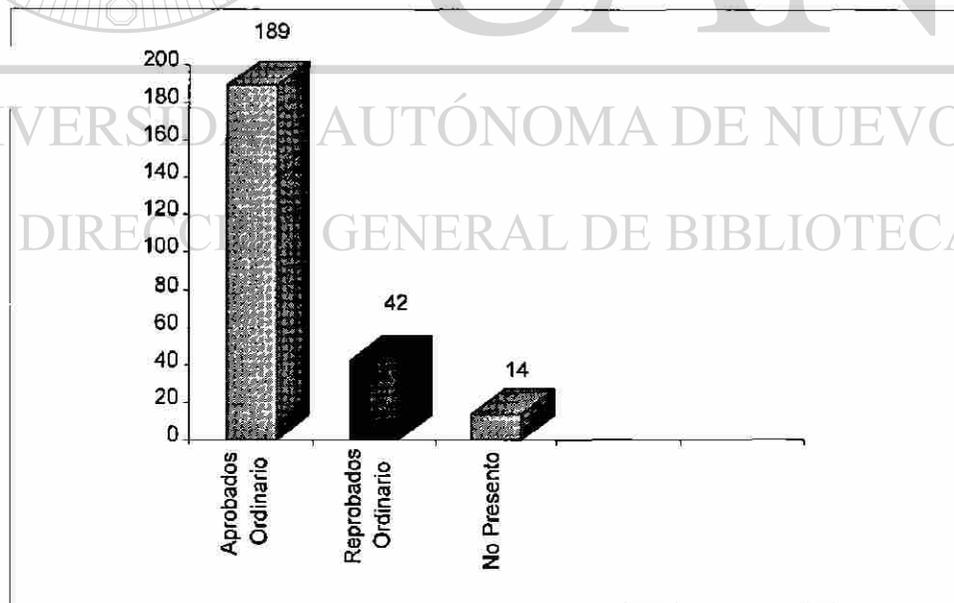
Semestre Agosto-Diciembre 1993

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	15	10	2	3
M2	2	34	25	7	2
N1	3	41	29	0	12
N2	4	41	31	0	10
N6	5	12	11	0	1
Totales		143	106	9	28



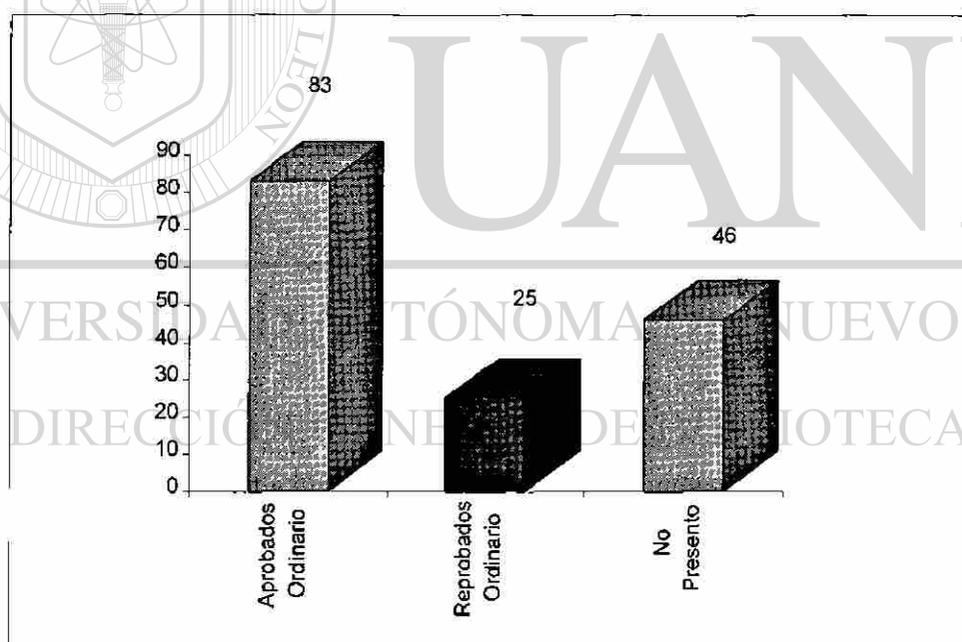
Semestre Febrero Julio 1994

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	44	27	15	2
M2	2	46	38	8	0
V2	3	39	29	10	0
N1	4	37	25	6	6
N2	5	47	40	1	6
N5	6	28	27	1	0
N6	7	4	3	1	0
Totales		245	189	42	14



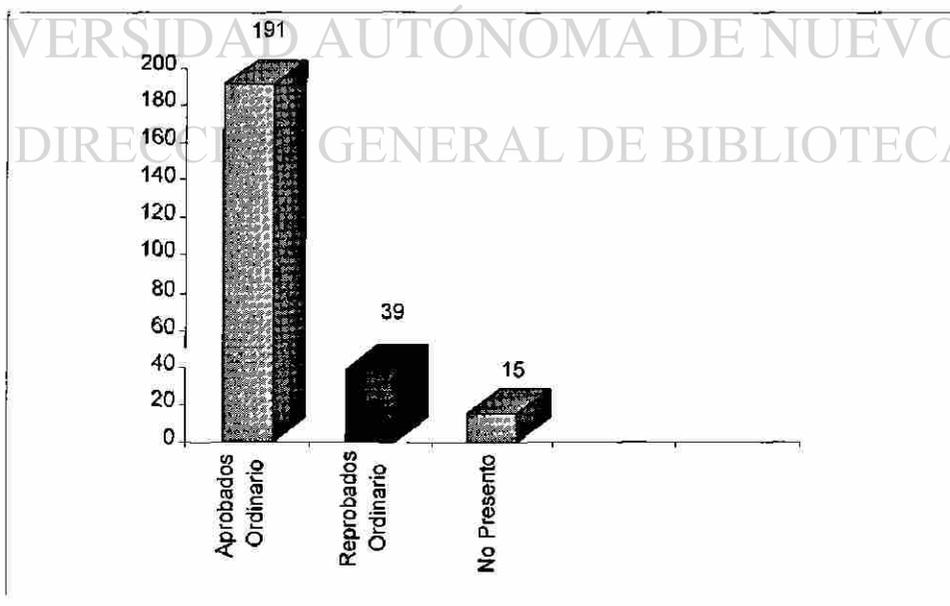
Semestre Agosto-Diciembre 1994

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	16	10	3	3
M2	2	29	20	5	4
V2	3	29	18	8	3
N1	4	39	15	6	18
N2	5	41	20	3	18
Totales		154	83	25	46



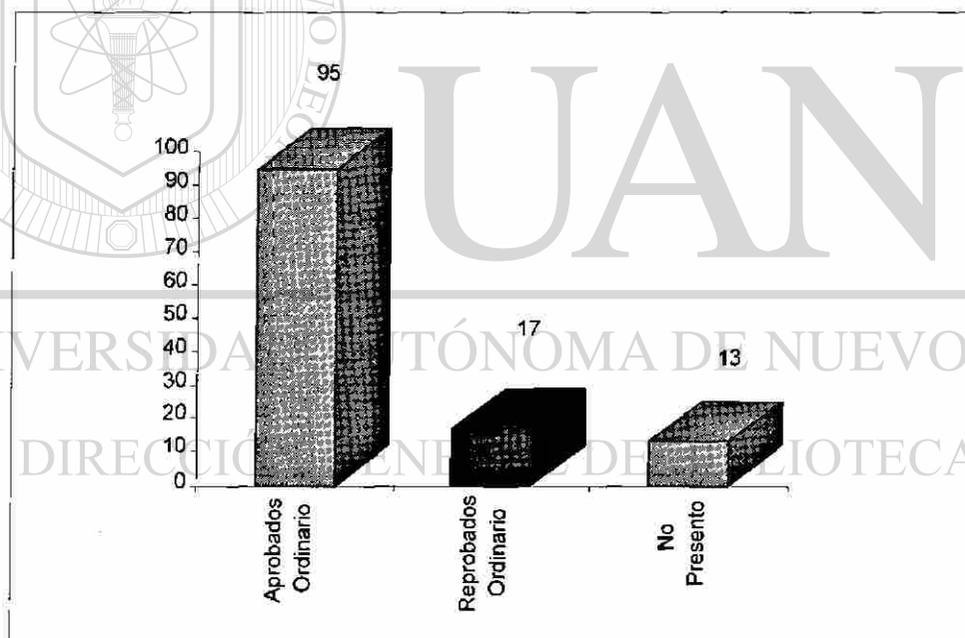
Semestre Febrero Julio 1995

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	29	23	5	1
M2	2	32	30	2	0
V1	3	32	27	4	1
V2	4	20	20	0	0
N1	5	35	18	10	7
N2	6	40	30	9	1
N4	7	32	24	6	2
N6	8	25	19	3	3
Totales		245	191	39	15



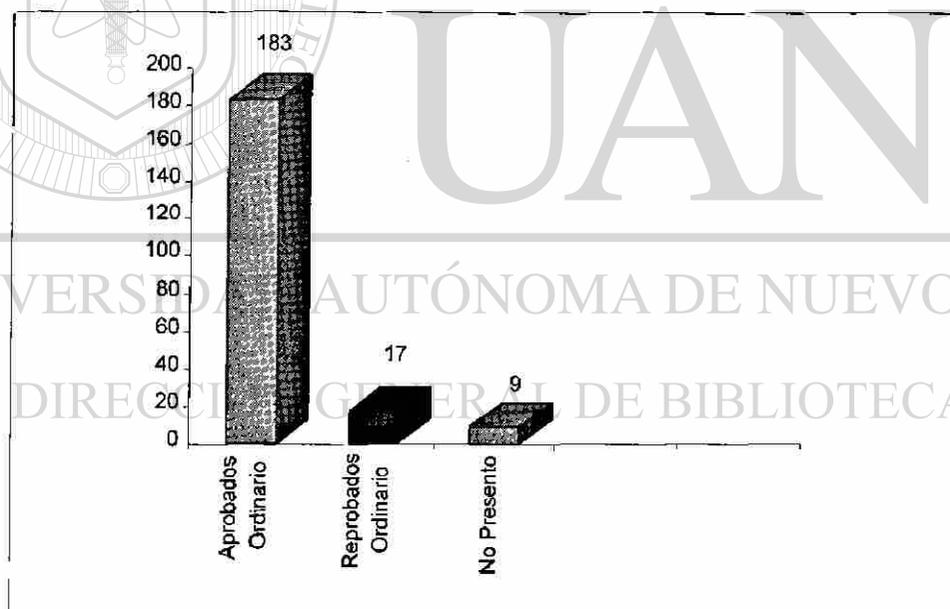
Semestre Agosto-Diciembre 1995

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	24	18	4	2
M2	2	43	34	3	6
N1	3	28	19	7	2
N6	4	31	24	3	3
Totales		126	95	17	13



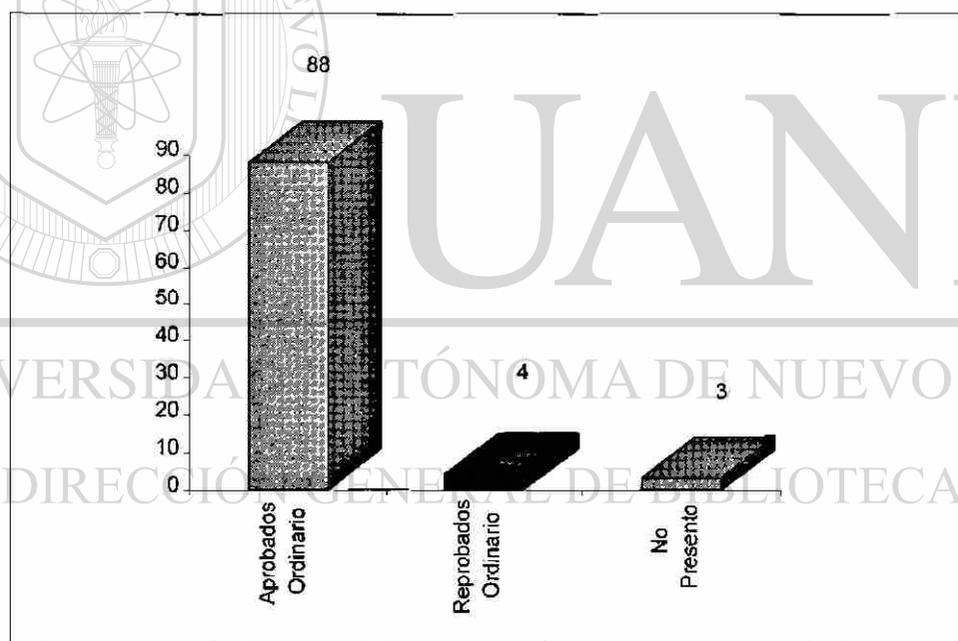
Semestre Febrero Julio 1996

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	31	29	1	1
M2	2	38	37	1	0
V1	3	34	33	0	1
N1	4	33	25	7	1
N3	5	24	17	6	1
N3	6	37	33	2	2
N6	7	12	9	0	3
Totales		209	183	17	9



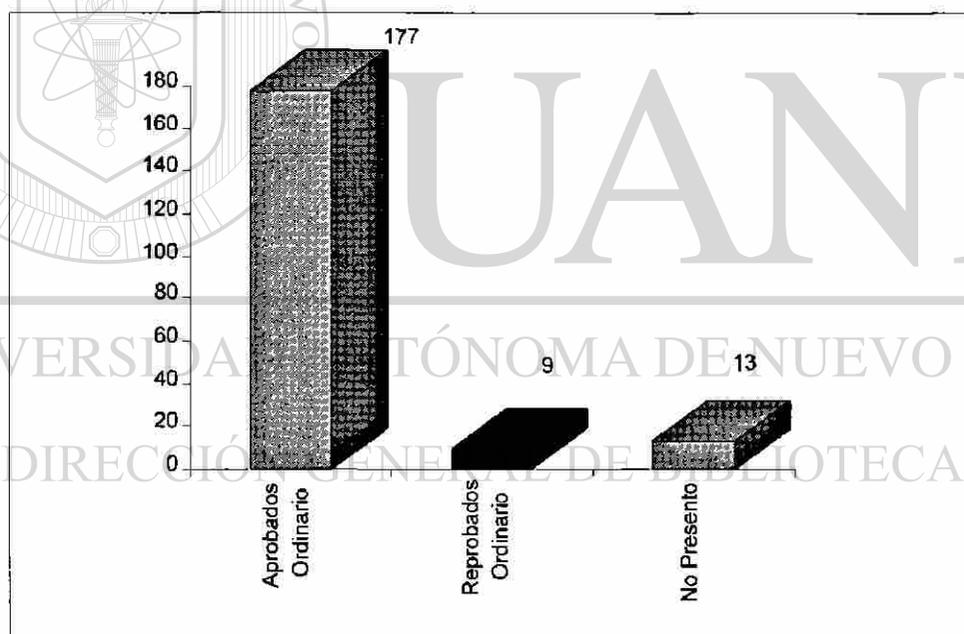
Semestre Agosto-Diciembre 1996

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	16	13	3	0
M2	2	27	26	1	0
N1	3	20	18	0	2
N6	4	32	31	0	1
Totales		95	88	4	3



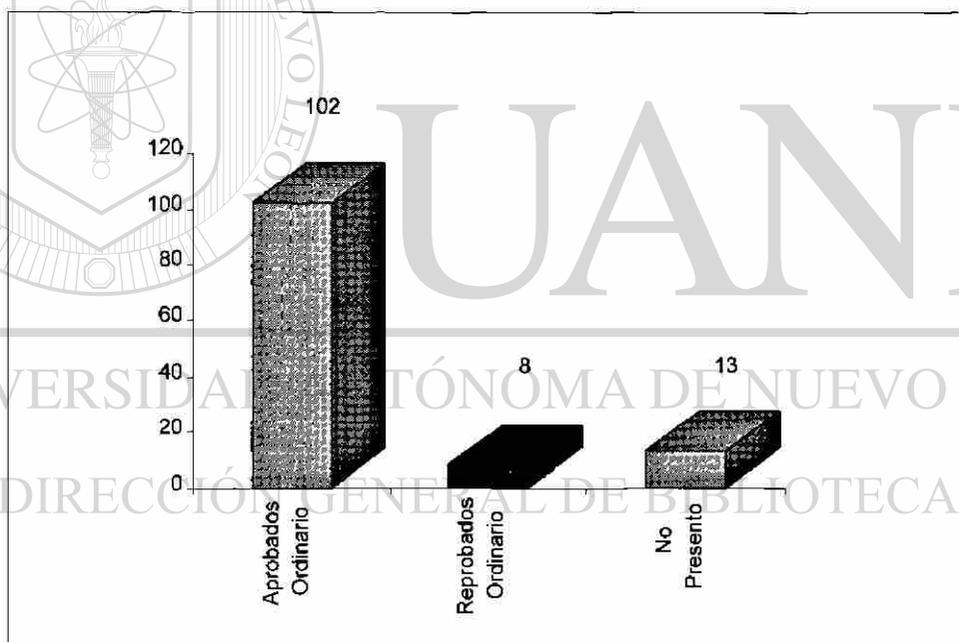
Semestre Febrero Julio 1997

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	30	26	3	1
M2	2	38	34	4	0
V1	3	25	25	0	0
N1	4	22	17	2	3
N2	5	32	26	0	6
N3	6	52	49	0	3
Totales		199	177	9	13



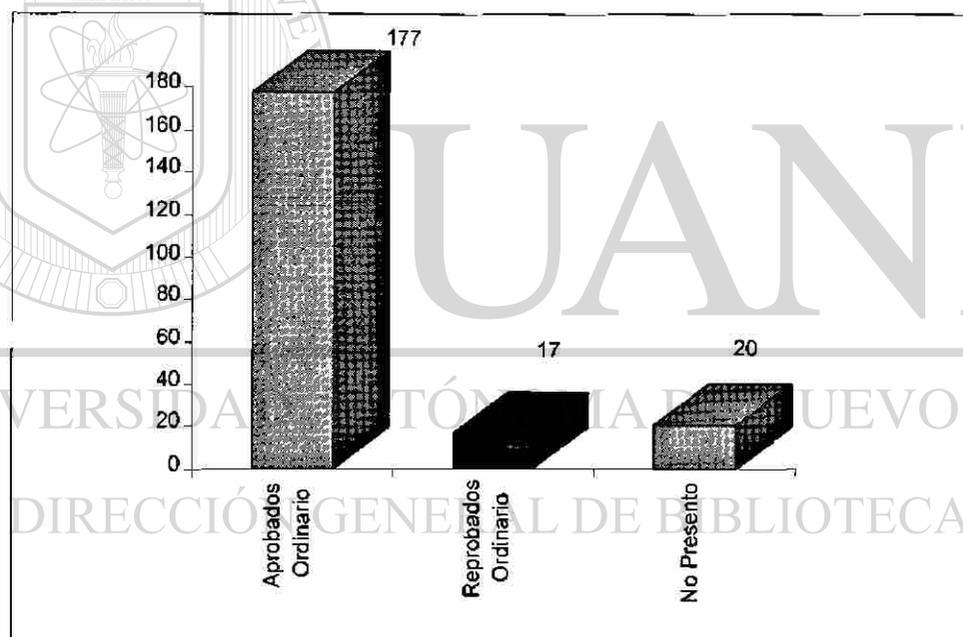
Semestre Agosto-Diciembre 1997

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	20	16	2	2
V1	2	27	25	1	1
N1	3	19	14	4	1
N2	4	23	14	1	8
N3	5	34	33	0	1
Totales		123	102	8	13



Semestre Febrero Julio 1998

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	23	16	3	4
V1	2	45	44	0	1
N1	3	16	14	1	1
N2	4	31	18	2	11
N3	5	59	50	8	1
N6	6	40	35	3	2
Totales		214	177	17	20



1.5. Hipótesis.-

Con la metodología que se plantea implementar para esta asignatura se motivará el interés en maestros y alumnos, además de que puede haber un aprovechamiento del 100% por ambas partes y decrementar el índice reprobatorio. Esto podría implementarse en materias semejantes, obviamente tecnológicas, más para materias de otras ramas de esta u otra carrera tendrían que hacerse los estudios pertinentes.

1.6. Metodología.-

Se tomará como muestra la población de estudiantes de la carrera de Ingeniero Administrador de Sistemas con un intervalo que va desde el semestre febrero – julio de 1993 hasta el más reciente, febrero – julio de 1998, información que será proporcionada por el Departamento de Servicios Escolares de la Facultad de Ingeniería Mecánica y Eléctrica, con esta información se harán las mediciones y estudios pertinentes del caso.

En segunda instancia, se procederá a la revisión del material bibliográfico con el fin de obtener una amplia perspectiva y profundizar en los temas fundamentales de la asignatura.

Posteriormente se analizará cada material didáctico y su posible incorporación y/o aplicación.

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

Capítulo 2

Análisis de Información Bibliográfica

Para abordar los primeros temas que conciernen a la historia de la computación me parece conveniente la lectura del “Tomo II de la Enciclopedia Metódica Larousse” que habla abundantemente del origen de la computación, para este curso se toma desde Charles Babbage hasta nuestros días. Para este tipo de información siempre nos ha parecido más confiable una enciclopedia pues se puede encontrar imparcialidad y la misma historia varía (en los textos) según el autor, ya que se le suele dar más importancia a algunos temas de acuerdo a las preferencias que tenga este.

De las aportaciones que el libro “Introducción a la Computación” de Peter Norton tomamos el capítulo 1, páginas 30-36, en el que nos apoyamos para establecer la panorámica de las computadoras, en esta sección se habla de la clasificación de las computadoras y sus características; en el capítulo 3, páginas 74-84, podemos encontrar un apartado de seguridad en los sistemas operativos; en el capítulo 6 se aborda el almacenamiento de información en una computadora, dando detalles precisos de cada dispositivo llevándonos a comprender mejor el material ya que cuenta hasta con fotografías magnificadas de la información respaldada en los dispositivos de almacenamiento secundario y se cuenta con una visión profunda respecto a este tema.

Del libro “Sistemas Operativos, conceptos y diseño” de Milan Milenkovic podemos apreciar una forma más clara de abordar la asignación contigua y no contigua de memoria en la página 23. Del capítulo 2 páginas 48 y 49 en la que se hacen observaciones de interés sobre los procesos concurrentes; en el mismo pero en la página 65 se definen muy bien los mecanismos de las llamadas al supervisor; la página 71 y de la 85-87 nos sirve como

complemento para la planificación de trabajos y del procesador. Del capítulo 3 páginas 97 y 98 para la sincronización de procesos concurrentes; de la 101-103 nos habla de la sección crítica y de la exclusión mutua; en la página 109 nos define a los semáforos, todo esto de una forma más ideal que otros libros. En cuanto al capítulo 4 el tema de las regiones críticas simples y condicionales en las páginas 146-147 se abordan con más claridad; en las páginas 183-184 detalla poco más exacto que otros autores los recursos reutilizables y consumibles así como en las páginas 195-196 el método combinado que ningún autor lo aborda. Del capítulo 5 en las páginas 214-217 se habla del swapping con más actualidad que otros autores. Del capítulo 6 en las páginas 266-268 el tema de memoria virtual es mejor abordado y el tópico de hiperpaginación y el algoritmo de frecuencia de fallas de página se tratan con mayor claridad en las páginas 287-288. En las páginas 322-326 del capítulo 7 se puede encontrar el tema del controlador y rutina de disco con especial cuidado con relación a otros autores. Los capítulos 8 y 9 de "Seguridad y Protección" y de "Entrada/Salida principios y programación" respectivamente son los que se consideran con amplio valor de consulta.

La razón por la que hacemos puntualización de las páginas y los temas es en función de que no se considere las 825 páginas de un libro para consulta sino sólo los temas que se manejan más adecuadamente.

DIRECCIÓN GENERAL DE BIBLIOTECAS

William Stallings nos aporta en su libro de "Sistemas Operativos" un panorama más rico en ejemplos y algunas figuras que hablan por sí solas por lo que no puede quedar fuera de consideración. Del capítulo 1 tomamos los elementos básicos que están en las páginas 19-20, las figuras 1.16 y 1.20 nos ilustran la jerarquía de la memoria y de la memoria principal y caché respectivamente siendo bastante claras en su aportación; en las páginas 70-71 se habla más objetivamente del DMA. Del capítulo 2 página 93 se habla de los servicios del sistema operativo; en las páginas 118-120 se aborda suficientemente sobre la protección y seguridad de la información. El capítulo

3 nos aporta la figura 3.22 de modos de acceso. El capítulo 4 en las páginas 279-283 aborda abundantemente sobre sincronización, direccionamiento y formato de mensajes. El capítulo 6 nos habla de la retroalimentación en las páginas 468-469. El capítulo 7 nos provee una tabla de dispositivos de entrada/salida en sus páginas 520-521; en las páginas 540-546 aborda con actualidad las políticas de planificación del disco. El capítulo 10, de seguridad, lo contemplamos enteramente como de consulta en sus páginas 711-756. Tal vez lo más interesante de este libro es que presenta un ejemplo del tema que esta hablando con cada sistema operativo lo que justifica su consulta.

El libro de texto oficial es el "Sistemas Operativos Modernos" de Andrew S. Tannenbaum es bastante amplio y minucioso tanto que en ocasiones nos perdemos aun una persona que ya cuenta con la experiencia en la materia por lo que sólo podemos tomar de él lo que corresponde como son: Del capítulo 1 figura 1.1, figura 1.2 y figura 1.3 de las páginas 2, 7 y 8 respectivamente. Capítulo 3 página 93 que aborda el tema de intercambio y la figura 3.47 de la página 159. Del capítulo 5 páginas 233-243 los temas hardware y software de entrada/salida, software para relojes de la 254 y terminales en la página 257.

El libro "Sistemas Operativos de H.M. Dietel en que nos proporcionó la base para los demás ajustes ya que tiene lógica y seguimiento tal vez su principal inconveniente son los errores de traducción que tiene, de éste tomamos los capítulos 2,3, parcialmente los capítulos 4 y 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 17. Tal vez porque la edición anterior "Introducción a los Sistemas Operativos" había dejado satisfechas nuestras aspiraciones pero al prescribir algunos de sus contenidos se buscó otra alternativa. Sin embargo, es este libro el que usamos como plataforma para el despegue ya que como dijimos antes tiene una secuencia lógica.

El libro de "Arquitectura de Computadoras " de Miguel A. Miguel y Teresa Higuera nos amplía nuestros horizontes desde el ángulo de la electrónica en sus capítulos 1 (figura 1.1, página 2) y 5 página 145-194. Lo consideramos un buen libro de consulta.

El libro "Arquitectura de Computadoras" de Morris Mano es un texto bastante amplio para consulta y nos explica a través de sus capítulos la información que debemos transmitir en clase, nos fundamenta todos y cada uno de los conceptos que manejamos en clase desde una perspectiva electrónica, todo el libro es muy interesante. Sin embargo nos parece apropiado el hacer las siguientes anotaciones: del capítulo 1, la figura 1.1 página 3, capítulo 2 el rom páginas 65-67, capítulo 3 el bit de paridad página 93, capítulo 5 el pulso de reloj página 144, capítulo 6 el lenguaje de máquina en las páginas 184-187 y lenguaje ensamblador en las páginas 189-190; capítulo 7 la memoria de control páginas 225-248, capítulo 8 unidad central de proceso páginas 255-256 y tipos de interrupciones páginas 299-303, arquitecturas RISC del capítulo 9 en las páginas 337-338, capítulo 11 dispositivos periféricos en las páginas 407-409, la transferencia serial asíncrona de la 423-424, transmisor-receptor en la 427, modos de transferencia en la 431, DMA en la 445-450 y comunicación serial de la 460-465; organización de la memoria en el capítulo 12 de la página 477-519 y multiprocesadores en el capítulo 13 páginas 525-551.

DIRECCIÓN GENERAL DE BIBLIOTECAS

Buscando otras opciones encontramos la dirección de Internet <http://alebrige.escom.ipn.mx/~gruiz/pavell/som/Apuntes.html> de la que podemos comentar que en general son apuntes actualizados ya que su última modificación corresponde a diciembre de 1997 y ofrece temas para Sistemas Operativos I y II. Se complementa con ejemplos muy interesantes aunque algo abstractos y no muy propios ya que podrían en un momento dado, dificultar el proceso, esto no le resta el valor que como aportación tiene.

Capítulo 3

Diseño del Programa

La construcción de programas se encuentra en el eje de la articulación didáctica-curriculum. De lo curricular se destaca la búsqueda de un contenido que se va a enseñar, lo didáctico apunta a redefinir una situación global en la que el contenido enseña.

El currículum surge históricamente en la perspectiva de atender a la “uniformidad” de los procesos institucionales, mientras que la didáctica considera que el docente, en su papel de intelectual y como profesional, le corresponde elegir y graduar los contenidos de enseñanza, así como idear las propuestas metodológicas pertinentes. Por lo tanto, aseveramos que el maestro es un intelectual y le concierne una función histórica estrechamente vinculada a la selección y a la organización de contenidos.

La formación pedagógica de los docentes de nivel superior debe proporcionar los elementos teórico-técnicos que les permitan interpretar didácticamente un programa escolar a partir de una teoría y de una concepción del aprendizaje que los lleven a propiciar en sus estudiantes aprendizajes acordes con el plan de estudios de la institución donde realizan su labor.

Ante la carencia de una formación didáctica sólida de los profesores que trabajan en la educación media superior y superior, originada por una escisión entre el conocimiento científico y el conocimiento didáctico, en las instituciones educativas se ha llegado a aceptar, tácita o explícitamente, que basta con saber para enseñar. “La peor expresión sería afirmar que si uno sabe bien un tema, le es posible enseñarlo; esta expresión es un rechazo cínico a la dimensión teórica de la educación”

Con las aportaciones de Hilda Taba la discusión sobre la propuesta curricular adquiere mayor solidez conceptual, y en ella las etapas para la formulación de

un plan y un programa de estudios tienen por objeto lograr la articulación entre la teoría y la técnica, esto tiene su fundamentación en Piaget y Brunner.

Para realizar el juicio ordenado que permita tomar decisiones en relación con los programas escolares, Taba enumera siete pasos:

1. Diagnóstico de necesidades
2. Formulación de objetivos
3. Selección de contenido
4. Organización de contenido
5. Selección de actividades de aprendizaje
6. Organización de las actividades de aprendizaje
7. Determinación de lo que se va a evaluar y las maneras de hacerlo

En cuanto los intentos por realizar una programación que se desprenda de planteamientos radicalmente diferentes, aun cuando existen dificultades para generalizar, podemos mencionar la elaboración de programas centrados en:

1. Las competencias o habilidades, tal como han recomendado en múltiples instrumentos los organismos internacionales.
2. El análisis y la función de los contenidos, lo cual supone un meticuloso trabajo de discriminación entre tipos de contenidos: los datos que requieren fundamentalmente el ejercicio de la memoria; los conceptos, que se refieren a procesos individuales de construcción de ideas complejas, no fácilmente traducibles a definiciones, y se vinculan a un proceso cualitativo de cada sujeto, y procedimientos, referidos a las formas de construcción del conocimiento en una disciplina particular, esto es, al método o la lógica del pensamiento que esa disciplina sigue.
3. La función del mismo programa, en cuyo caso se diferencian los elementos que integran un programa de estudios a partir del papel que tienen en la dinámica institucional y en el trabajo docente. De aquí se pueden distinguir tres tipos de programas:
 - Del plan de estudios o del sistema educativo.
 - De la institución o academias de maestros

➤ **Del docente.**

El aula es un espacio fundamentalmente docente-estudiantil. Y en este sentido, la propuesta que desarrollamos en relación con los programas tiende a cumplir una doble finalidad:

1. La consideración de funciones genera la posibilidad de conciliar los intereses institucionales con los docentes en los diversos tipos de programas.
2. El programa de estudios, visto desde la perspectiva de las funciones que cumple, es un elemento articulador de las problemáticas que emanan tanto de la concepción curricular como de la didáctica.

3.1. Programa del plan de estudios o del sistema educativo (programa sintético)

Es un punto de concreción de los proyectos educativos amplio que materializan una política de formación para un nivel educativo, y un punto de partida para los desarrollos que cada academia de maestros o escuela realice con objeto de adecuar los criterios globales que se desprenden del programa a las condiciones institucionales particulares.

El programa escolar no es un elemento aislado sino que tiene una profunda inserción curricular, esto significa que todo programa escolar constituye una parte concreta de un plan de estudios. Esta idea de que hay relaciones horizontales y verticales entre los contenidos de un plan de estudios apunta de manera indirecta a la necesidad de aceptar que en un plan de estudios los contenidos suponen relaciones antecedente-consecuente entre los diversos temas que se tratan cada semestre. A su vez obliga a considerar el problema de la profundidad con la que estos contenidos pueden ser desarrollados en un curso, o bien las diversas temáticas de las asignaturas que hay que abordar en un semestre académico, o los elementos que contribuyen a su mutuo enriquecimiento y a su integración; aunque se puede dar el caso de que en una elaboración deficiente del plan no exista el suficiente encadenamiento de temas entre una asignatura y otra, o bien que las materias previstas para un semestre

tengan elementos que plantean una contradicción entre ellas, lo que se puede convertir en un obstáculo para el aprendizaje.

Este concepto reconoce la necesidad de que los docentes tengan elementos para interpretar el plan de estudios y para estudiar de qué manera determinado programa forma parte de una táctica concreta que permite, por medio de los aprendizajes, el logro de las metas curriculares.

El plan de estudios se encuentra íntimamente relacionado con los propósitos de ese plan o sistema, el tipo de necesidades sociales e individuales que se consideraron en su elaboración, las áreas de formación en que está organizado, las nociones básicas de cada una de dichas áreas; con el fin de obtener un mapa curricular que permita visualizar la forma en que se apoyan e integran los diferentes contenidos de las asignaturas del plan de estudios, para así evitar la repetición de contenidos y procurar la integración de los aprendizajes. El mapa curricular está constituido por la descripción sintética de contenidos de cada una de las asignaturas que forman el plan de estudios.

Los programas del plan de estudios permiten esclarecer las nociones básicas por desarrollar en cada área de formación y en cada asignatura que forme parte de ella; estos contenidos básicos constituyen un programa sintético.

3.2. Programa de la institución o de la academia de maestros (programa analítico)

Fundamentos de un programa de institución o de academia

El programa escolar es una propuesta de aprendizaje, una propuesta referente a los aprendizajes mínimos que se promoverán en un curso, que forma parte de un plan de estudios y orienta las estrategias de trabajo de autoridades, maestros y alumnos.

La elaboración de este tipo de programa se fundamenta en el conocimiento del plan de estudios y en el análisis sistemático tanto de las condiciones institucionales como del conjunto de experiencias docentes frente a los temas que se intenta desarrollar. El valor de este programa depende de la manera en

que se realicen diversos análisis que articulen el programa al plan de estudios del que forma parte, a las particularidades de un proyecto institucional y a las condiciones de sus maestros y alumnos.

Es difícil determinar cuáles son los elementos que conforman este programa, una vez que se han establecido sus fundamentos y determinado sus propósitos más importantes, los principales contenidos por unidades, la bibliografía y, cuando se han esclarecido los criterios generales que orientan su sentido metodológico – cómo se espera que participen docentes y alumnos, de qué presupuestos se parte para pensar las estrategias de aprendizaje, y cómo se establecen los criterios de acreditación y evaluación educativa – se ha avanzado mucho.

El programa de institución o de academia de maestros es un instrumento que orienta globalmente el trabajo en una institución escolar. Se construye a partir de una interpretación del programa del plan de estudios o del sistema educativo; cuando el plan llega a la institución educativa, los docentes, en academia o en forma individual, se enfrentan a la tarea de adecuarlo a su realidad y desarrollar los elementos básicos que en él se establecen.

La experiencia del conjunto de docentes que participan en la elaboración del programa institucional o de academia es un punto nodal para el desarrollo de esta tarea. Se trata de la posibilidad de reflexionar haciendo un desarrollo conceptual con respecto a los grupos de alumnos con los que anteriormente han trabajado, de la forma como desarrollaron el contenido, de las principales estrategias metodológicas empleadas y de los resultados grupales e individuales que se obtuvieron. La construcción del programa institucional es un espacio privilegiado para promover la reflexión, permite potenciar las diversas posibilidades institucionales frente a esta tarea. No obstante, es en la experiencia, ligada a su propia práctica, donde cada docente tiene condiciones para construir nuevas significaciones y generar procesos creativos, pues la tensión que surge cuando se está “cara a cara” con un grupo de alumnos genera cierta ansiedad que permite la recreación de la información.

Tanto el “mundo interno” del docente como el del estudiante se encuentran completamente signados por lo singular, esto dificulta el conocimiento preciso de cada uno de estos actores del proceso didáctico. Sin embargo, ateniéndonos a una teoría social de la subjetividad, hay elementos para considerar que los jóvenes de la generación de los noventa estructuran socialmente un conjunto de valores, comportamientos y procesos de identidad muy singulares. Como un efecto de la “universalización” de ciertos patrones de comportamiento a partir de la globalización de las series televisivas, los comerciales, los videojuegos, y el creciente acceso a las redes de información como Internet, los jóvenes han ido integrando una escala de valores marcada por el inmediatez, lo práctico, el “sinsentido de la vida y del esfuerzo”; una cultura del “placer” y del “aquí y ahora”. Estos elementos, muy característicos de nuestra época, tienden a crear un sostenido divorcio entre las propuestas educativas y la realidad histórico-social de los jóvenes. El sueño de la democracia, de la igualdad de oportunidades y de la equidad se encuentra completamente desvanecido frente a ellos. Un conocimiento de los procesos culturales generales en los que están inmersos los estudiantes de un plantel educativo ayudará a prever una programación más adecuada a sus necesidades y condiciones.

Es conveniente tener información sobre las condiciones de estudio de los alumnos, sus hábitos de trabajo, los dominios básicos que se exigen para un curso escolar: capacidad de lectura, capacidad de síntesis, etc. Este conjunto de elementos seguramente contribuirá a desarrollar programas pensados en función de determinados grupos escolares y no sólo desde la perspectiva universal del sistema educativo. La información que se obtenga sobre los estudiantes es un elemento valioso para elaborar el programa institucional o de la academia.

El conocimiento de las condiciones y posibilidades institucionales es un elemento importante para construir el programa institucional o el plan de estudios. El trabajo grupal requiere determinado tipo de formación y disposición en los maestros, exige un número específico de alumnos y algunas condiciones en el aula. El uso de material de apoyo (videos, retroproyectors,

computadoras, etc.) implica que la institución cuente con una infraestructura específica.

Conocer las posibilidades que institucionalmente tiene un plantel educativo nos ayudará a prever de manera realista el tipo de programas que se puede aplicar y orientará a los docentes sobre las respuestas creativas que tiene que dar en su trabajo escolar, y las condiciones particulares que tienen que afrontar en el mismo.

Cada grupo escolar vive una situación particular que determina sus condiciones de aprendizaje, y éstas, lejos de ser exclusivamente individuales y de carácter metodológico, están conformadas por una serie de situaciones sociales, históricas y culturales que es necesario conocer para comprender al grupo y así elaborar una propuesta didáctica pertinente.

El programa institucional o de academia debe partir del reconocimiento de las inercias institucionales, de las potencialidades que se encuentran en cada escuela y, a la vez, necesita prever qué procesos de cambio se pueden potenciar desde la misma institución.

Es necesario ampliar los esquemas de análisis con que se realiza esta tarea, y comprender cómo se relacionan los elementos individuales y los sociales, los elementos metodológicos y los históricos, para así poder organizar una primera configuración explicativa del grupo, que permita elaborar las estrategias de intervención docente. Es aquí donde consideramos que determinar la situación y el campo de un grupo se convierte en un elemento indispensable para la realización de esa tarea.

Una tarea formal entra en conflicto permanente con las exigencias de la realidad, lo que a fin de cuentas desembocará en un proceso de crear soluciones sobre la marcha al que llamaremos "resolución situacional". Por ello el docente debe tener una idea clara de los presupuestos globales y finalidades que existen tanto en el plan de estudios como en el programa institucional o de academia y, a la vez, a partir de su propia posición conceptual (filosófica didáctica), tomar las mejores decisiones en espacios situacionales que no necesariamente permiten que el maestro resuelva con calma la evolución de un

grupo escolar. Cada grupo necesita que el docente “decida”, confirme o modifique sobre la marcha una estrategia propuesta. Esta decisión situacional resuelve la tensión entre lo formal y lo vivido, y en ella se realiza el trabajo en el aula. Esta decisión se toma con mejores bases cuando se dispone de un programa institucional o de academia, y cuando se tiene el bagaje conceptual y la sensibilidad necesarios para trabajar como docente en un curso escolar.

Desarrollo del programa institucional o de academia

El programa institucional o de academia tiene como fin orientar a la comunidad académica (maestros y estudiantes) acerca del desglose de los principales temas a trabajar, de su ordenamiento en unidades de aprendizaje, de las principales propuestas metodológicas que se desprenden del mismo y de la forma en que, atendiendo a los reglamentos de evaluación y al desarrollo académico institucional, se sugiere realizar la evaluación.

Presentación general

La presentación escrita de un programa institucional o de un plan de estudios consiste en la especificación de las principales características del curso, de los contenidos básicos que se desarrollarán, de las relaciones que guarda esta materia con las que la preceden y las que la siguen, desde un punto de vista de los problemas concretos que ayuda a resolver. Esta presentación permite concebir una panorámica general del curso y representa un primer intento por estructurar el objeto de estudio, de manera que se perciban las relaciones que guardan entre sí los principales elementos que la conforman. Su redacción tiene por objeto desarrollar una explicación que incluye el significado del curso, sus propósitos explícitos y su vinculación con el plan de estudios del que forma parte, en un lenguaje accesible para el alumno.

Determinación de los propósitos o intencionalidades, y establecimiento de los criterios de acreditación del curso

Una vez determinados los propósitos de la totalidad del curso y los contenidos básicos, se pueden precisar los grandes propósitos o intenciones de un curso en particular (el sentido general al que apunta), así como los desempeños y compromisos que se exigirán a los estudiantes y docentes. Es conveniente pensar en algunos resultados o productos del aprendizaje que permitan integrar la información, a lo largo de un curso, en relación con una problemática (teórica o práctica); la elaboración de los productos de aprendizaje constituye una síntesis de los contenidos desarrollados de acuerdo con el papel que tienen dentro de una disciplina en particular y en un plan de estudios específico. Estos productos de aprendizaje orientan con toda claridad los elementos para la acreditación del curso; estos elementos se desarrollan en dos niveles:

- Uno que atiende a las exigencias institucionales (requisitos de asistencia, participación).
- Otro que se centra en los contenidos, el concepto de aprendizaje y las formas en que se pueden obtener datos de este proceso cualitativo, interno y complejo.

Con este planteamiento deseamos apartarnos totalmente de la redacción de objetivos como expresiones de desempeño, llámense éstos terminales, generales o de unidad, y en su lugar proponer que en los programas de estudio se expresen con claridad los grandes propósitos o intenciones del curso (para maestros y alumnos) así como las precisiones globales sobre los mecanismos y contenidos de la acreditación.

Si se concibe el conjunto de actividades de acreditación como evidencias de un proceso de aprendizaje, necesitamos elaborar en estos resultados una propuesta que busque tanto la integración del contenido del curso, como la relación que existe entre esa información y una problemática (teórica o práctica) con la cual necesariamente debe entrar en juego. También las actividades de acreditación se pueden considerar un mecanismo para ayudar a los estudiantes

a acercarse a una información (exámenes diarios), lo cual complica el proceso de acreditación al confundirlo con los desarrollos metodológicos de un curso. Pretendemos diferenciar ciertas prácticas de acreditación que exigen al estudiante sólo una síntesis de cierta información, pues esta síntesis refleja solamente un ejercicio memorístico. Es necesario permitir la interrelación de toda la información, por ello hablamos de una dimensión metodológica, así como del trabajo con ciertos problemas concretos, los cuales en ocasiones pueden ser de índole práctica y aun empírica, sin negar el indispensable valor formativo que tiene la vinculación con problemas teóricos, como elemento indispensable para la formación del pensamiento autónomo. Este problema guarda una relación estrecha con el nivel de acceso a la formación que tiene cada estudiante, el nivel educativo del que forma parte cada programa y el sentido que el plan de estudios le asigna a las diferentes temáticas que se abordan en un curso.

Organización del contenido en unidades

Esta es una tarea fundamental en la elaboración del programa institucional o de academia de maestros. Es necesario reconocer la carencia de una metodología y una técnica adecuada para el análisis de los contenidos. Esta situación ha propiciado que las unidades de los programas reproduzcan los capítulos de un libro, o bien que los temas que integran los programas se inicien con un tema cronológico sin profundizar en los elementos históricos que puedan dar sentido al conocimiento de una disciplina. Estas unidades definidas como introductorias, generalmente no se relacionan con los propósitos globales de un curso, ni con los conocimientos y habilidades que forman parte de la acreditación.

El análisis de la organización de los contenidos nos remite también al problema metodológico. Contenido y método forman parte de una unidad indisoluble y es necesario abordarlos en forma conjunta. Así como los contenidos matemáticos, físicos, históricos o literarios llevan de manera implícita formas y experiencias de enseñanza vinculados a su disciplina, lo que hace que las propuestas metodológicas en este punto difieran entre sí.

Es conveniente que la organización del contenido refleje la estructura interna de una disciplina; por ello, consideramos que son insuficientes las técnicas de análisis de contenido que particularmente se han difundido para realizar este trabajo basándose en la enseñanza programada.

Del empleo de la lógica formal hemos pasado al esquema estímulo-respuesta, o a una simplificación numérica del sentido común; paradójicamente ninguno de estos planteamientos pretende “descubrir” o “formalizar” las estructuras conceptuales de una disciplina.

Visualizamos dos supuestos que deben tomarse en cuenta con respecto a la problemática de los contenidos; el primero está relacionado con la necesidad de que los contenidos se presenten a los estudiantes integrados de tal manera que permitan la percepción de la unidad y la totalidad que guardan los fenómenos entre sí.

Los contenidos conceptuales y de proceso se caracterizan por un elemento cualitativo que provoca en los sujetos el desarrollo de procesos que no concluyen. La nueva información que se presenta en un curso escolar tiende a incrementar cualitativamente este proceso de conceptualización, pero conservando la singularidad de cada sujeto. El concepto que tiene cada alumno es algo más que la exclusiva verbalización que pueda dar de una noción en un momento dado, ese “algo más” se encuentra determinado por el carácter cualitativo de la construcción del concepto; su aprendizaje, lejos de ser memorístico, reclama una serie de procesos de comprensión, de construcción de ideas y de apropiación que van singularizando la expresión conceptual en cada integrante de un grupo escolar.

El problema de una estructura subyacente en los contenidos corresponde a un análisis que necesariamente debió figurar en la definición de contenidos realizada para elaborar el programa del plan de estudios. Esta actividad permite visualizar cuáles son los elementos de los contenidos de una asignatura que se relacionan con los que se abordan en otras materias de ese plan.

Si existe claridad sobre los contenidos como sobre su estructura interna es posible buscar un adecuado mecanismo de agrupación de los mismos. A esta agrupación le podemos llamar:

- Unidades temáticas
- Núcleos problemáticos
- Capítulos de un curso escolar

Es conveniente que estos grupos de contenidos respondan a un criterio organizador; por lo que es importante que no reflejen una fragmentación o que no sean excesivos en su planteamiento.

Cuando el contenido del curso se encuentra organizado en unidades temáticas, a cada una de ellas se le puede asignar un nombre que refleje el contenido que la integra. Una vez hecho esto, es factible redactar una presentación escrita de cada unidad o bloque temático, lo que contribuye a aclarar a los maestros y alumnos el papel, la estructura y el tipo de aprendizaje que se buscará promover, así como su relación con la totalidad del programa.

El programa de la institución o de academia de maestros presenta una propuesta de aprendizajes mínimos que se van a promover, los que se desprenden directamente de un plan de estudios determinado.

Reconocemos que a cada docente – a partir de su formación, su conocimiento de la materia y del plan de estudios y ante un grupo particular de alumnos – le corresponde interpretar y adecuar el programa institucional o de la academia de maestros, a una situación particular. De hecho, son los profesores quienes imprimen vitalidad al plan de estudios y hacen posible su realización. La elaboración de una propuesta mínima de bibliografía tiene un valor importante en la construcción del programa institucional o de la academia de maestros. En esta selección es necesario considerar el nivel escolar al que se dirige el programa. En la selección de bibliografía para la enseñanza superior, conviene tener en cuenta el rigor y la actualidad con los que se aborda determinado tema, la diversidad de enfoques y el enriquecimiento de la perspectiva del estudiante, la posibilidad de confrontación que pueden ofrecer los autores. La elección de bibliografía es un factor que necesita manejarse con flexibilidad en

los programas escolares, puesto que cada año aparecen nuevas publicaciones que es necesario que los estudiantes conozcan.

En resumen, la elaboración del programa institucional o de academia de maestros es una propuesta de aprendizajes mínimos que la institución presenta a sus docentes y a sus alumnos y constituye la información básica con la que se trabajará. Este programa puede constar de cuatro elementos:

1. Una presentación general que explica el significado del programa y las articulaciones que establece con el plan de estudios
2. La presentación de una propuesta de acreditación
3. La estructuración del contenido en forma de unidades, bloques de información, problemas y,
4. Bibliografía mínima

3.3. El programa del docente (programa guía)

El docente es el responsable profesional del trabajo en el aula; a él le corresponde elaborar una propuesta de programa en el que fundamentalmente se definan las estrategias de enseñanza que se realizarán en el curso escolar.

En este programa, el docente combina el plan de estudios, el institucional o de academia de maestros, y sus experiencias profesionales; su formación y manejo de los contenidos, el análisis que realiza del efecto de las diversas actividades de aprendizaje en grupos escolares similares, la perspectiva que tiene sobre los elementos que singularizan al grupo escolar con el que trabajará este programa.

La elaboración de este tipo de programa surge al reconocer que es imposible efectuar una propuesta metodológica general para todas las situaciones en las que un programa se desarrolla, ya que esto supone la existencia de grupos y estudiantes en los que no existe ninguna variación social ni cultural; así mismo, sería una negación de la especificidad institucional y de las características de cada docente.

Este programa parte de los elementos básicos establecidos en el programa institucional o de la academia de maestros, y constituye un puente privilegiado para promover una articulación entre el currículum y la didáctica.

La didáctica es la ciencia del docente, desarrollada en atención a las características que adopta la práctica escolar en cada aula. Por ello, un elemento central del programa del docente es la elaboración de una propuesta metodológica, la elección de una estructura en la que se enlacen las diversas actividades de aprendizaje, pensadas éstas en el sentido de las experiencias educativas que promueven en los estudiantes. Así, la propuesta metodológica surge de la misma experiencia y formación didáctica, y permite una articulación específica con los contenidos que se seleccionan. El método debe ser considerado como la posibilidad de articular ciertos contenidos, como un punto de unión de lo epistemológico (un campo disciplinar particular) y lo didáctico.

En el programa del docente es importante la especificación del mecanismo de acreditación, tanto de los requisitos formales que se deben cumplir para tener acceso a la misma, como del conjunto de resultados y actividades que serán tomados en cuenta para integrar la calificación final.

La acreditación escolar es una exigencia institucional para certificar ciertos conocimientos. La aparición de la acreditación transformó el sentido histórico del examen como el momento de mostrar en público la madurez adquirida, o como un momento significativo en la metodología de la enseñanza. ®

Corresponde al docente, como responsable del trabajo didáctico y metodológico, la tarea de definir los elementos que se tomarán en cuenta en la acreditación escolar. Las precisiones sobre la acreditación, al establecer una serie de actividades que se considerarán para calificar el curso, ayudan al estudiante a clarificar el curso, ayudan al estudiante a clarificar el sentido de su actuación escolar. Estas actividades (ejercicios, trabajos, reportes, exámenes) necesitan ser pensadas como una forma de articular el método, las estrategias realizadas a lo largo de un curso con el contenido del curso.

Una aproximación a lo metodológico

Este es quizá el punto de partida de mayor riqueza del programa del docente, puesto que en él fundamentalmente se comunica la propuesta metodológica general que orientará el curso. El método tiene la perspectiva global de las estrategias de enseñanza que sintetiza tanto las concepciones generales que tiene cada docente, como su punto de vista psicopedagógico; esto supone la adopción de una visión cognoscitiva, humanista, genética y psicoanalítica que busca la apropiación del conocimiento, o bien, el descubrimiento y la construcción del mismo, lo que genéricamente denominamos didáctica. En la propuesta metodológica, el docente expresa su creatividad y a través de ella manifiesta la vitalidad que tiene frente a la educación. Es en el diseño y, sobre todo, en el desarrollo de diversas estrategias didácticas en lo que el docente necesita obtener las mayores satisfacciones, pues finalmente en ellas se traduce el reto de lograr que el trabajo de los contenidos sea agradable, aunque no por ello menos riguroso.

El momento de asimilación está conformado por prácticas educativas en las que se presenta al alumno una nueva información, bien sea por exposición del docente, por exposición de los mismos alumnos, o por medio de textos, material audiovisual o conferencias; mientras que el momento de acomodación estaría representado por prácticas educativas que fomentan la discusión un contenido en relación con otros contenidos, o en relación con ciertos problemas.

Esta discusión puede hacerse con pequeños grupos o con todo el grupo, y permite, en una última etapa, el planteamiento de nuevas preguntas, el señalamiento de algunas contradicciones en el contenido, la precisión sobre los alcances y limitaciones del tema estudiado, así como la formulación de otras hipótesis sobre el objeto de estudio.

El método se puede abordar en tres niveles de conceptualización:

1. Como problema epistemológico, en el que tiene íntima relación con los procesos de construcción de disciplina.
2. Como vinculación con las teorías del aprendizaje y una posibilidad de concreción de los principios que se derivan de cada una de ellas.

3. Como un ordenamiento de las etapas que se establecen como parte de las estrategias de enseñanza en el aula.

Un criterio útil para pensar en la estructura global de las actividades de aprendizaje que se desprende de estos planteamientos en los que se subraya la conveniencia de propiciar procesos de pensamiento de análisis y síntesis, es el que plantea Hilda Taba cuando organiza las actividades de:

1. Introducción
2. Desarrollo
3. Generalización
4. Aplicación

Estos momentos no pueden considerarse independientes del contenido ni del proceso de aprendizaje de los estudiantes; las síntesis iniciales, los análisis que de éstas se desprenden y las nuevas síntesis que se configuran a partir de los momentos anteriores, están vinculados a la necesidad de identificar una articulación entre el contenido de un curso y el aprendizaje de los alumnos (considerando su edad, condiciones socioeconómicas y culturales, y los apoyos institucionales existentes); éstos son elementos básicos que deberán tenerse presentes en esta definición.

La riqueza metodológica que a cada docente le corresponde explorar y experimentar, atendiendo a criterios de eficacia, fundamentación conceptual, pero también de sencillez y gratificación en la misma acción educativa. La pasión, la sensibilidad, la creatividad y la capacidad de innovación constituyen un gran reto para el docente; la organización de propuestas metodológicas permite enfrentar con los elementos concretos dicho reto, pues cada estructura de actividades bien concebida y bien lograda es un motivo de satisfacción para el docente.

Evaluación y determinación de criterios de acreditación del curso

Una distinción más pertinente se puede establecer entre las nociones de evaluación y de acreditación; en esta perspectiva, la evaluación podría referirse

a la comprensión de las situaciones que acompañan el proceso de aprendizaje, a los elementos que afectaron positiva o negativamente ese proceso, así como al estudio de aquellos aprendizajes que, no estando previstos curricularmente, ocurrieron en el proceso grupal, en un intento por comprender el proceso de la educación. La acreditación, por su parte, se refiere a la verificación de ciertos resultados del aprendizaje previstos curricularmente, y que han sido traducidos a determinadas exigencias en los lineamientos de acreditación establecidos en el programa de la institución o de la academia de maestros, reinterpretados y precisados en el programa del docente, atendiendo a un mínimo manejo de cierta información por parte de los estudiantes. Los problemas de la acreditación son más restringidos que los de la evaluación, porque se insertan en una perspectiva de evidencias que dependen de la dinámica institucional y social, y no de la que se deriva del proceso de aprendizaje.

Recordemos que las precisiones sobre la acreditación se empiezan a establecer desde que se formula el programa institucional o de la academia de maestros. En ese momento se consideran tanto los requisitos de acreditación que establece cada institución educativa, como la especificación de lineamientos de acreditación que de alguna forma sintetizan una serie de acuerdos internos en una planta académica.

Cuando los maestros planifican la acreditación de un curso – al principio del mismo – es conveniente que se aclaren los tipos de aprendizaje que desean promover en él; ejemplo, si se desea promover procesos analíticos o críticos, necesitan propiciarlos en las actividades de aprendizaje, y el proceso de acreditación sólo será una posibilidad de expresión de lo que las actividades de aprendizaje han promovido, no pueden ir más allá.

El interés de propiciar aprendizajes analíticos y críticos no se promueve a partir de un examen difícil, y que si se pretende que los alumnos no sólo memoricen datos y fechas, es necesario modificar las prácticas de enseñanza que llevan a los docentes a recitar su clase ante un auditorio y que los alumnos anoten o copien lo que dice el profesor.

La planificación de la acreditación abarca la determinación de criterios generales de acreditación del curso, en los que se consideren los productos de aprendizaje que puedan expresar la mayor integración del aprendizaje posible; asimismo, estos productos serán el medio en el cual los estudiantes recuperen la información fundamental de un tema o disciplina. Esto representa una interesante combinación de elementos procesales con memorización de la información. No parece que este problema se pueda resolver mediante pruebas construidas basándose en preguntas establecidas en un "banco" de reactivos y seleccionadas usando técnicas algorítmicas por computadora, dado que para comprender el manejo de los contenidos, es necesario detectar la capacidad de establecer las relaciones, de hacer síntesis y de realizar juicios críticos que permitan el desarrollo de los procesos de pensamiento.

Planear las directrices para la acreditación implica establecer los criterios de lo que se exigirá durante y al término del curso, sus grandes etapas y sus formas de desarrollo. Es muy conveniente dar a conocer estos criterios junto con el programa desde el primer día de clases, ya que es un elemento importante del llamado "contrato escolar" (o la sección de acuerdos del encuadre).

Las tareas que se exigen como resultado del aprendizaje (tareas, ensayos, prácticas, reportes, investigaciones, exámenes, etc.), no tienen que realizarse forzosamente en el ámbito del aula, y es conveniente exigir algunos procesos que vayan más allá de la mera repetición de la información o de la ejercitación mecánica de la misma. Tampoco es adecuado que se limiten a una sola actividad (trabajo, examen) realizada sólo al final del curso. De hecho, en las situaciones de resolución de exámenes se demuestra la poca importancia que se concede a la biblioteca y al manejo de fuentes de información en el aprendizaje de los alumnos.

Especificar, a grandes rasgos, en qué consiste este tipo de tareas permite reconsiderar la propuesta metodológica que el docente adoptará en un curso escolar, así como la selección y organización de contenidos adoptada.

Elaborar un reporte individual que refleje las discusiones que se hicieron en pequeños grupos y se insiste en que, cuando no haya acuerdo en la discusión

grupal, el desacuerdo se manifieste en estos informes, con las fundamentaciones del caso.

Los estudiantes deberán estar enterados del plan de acreditación del curso desde el inicio del mismo; esto constituye un elemento que puede favorecer su motivación y el compromiso para su desarrollo. El desarrollo de las etapas definidas en este plan no tiene que realizarse forzosamente al finalizar el curso. Promover el manejo de estructuras de contenido es algo que va más allá de los problemas de la acreditación, en general el desarrollo de las estructuras conceptuales no es el resultado de la resolución de un examen, sino de las actividades realizadas en el aprendizaje en una perspectiva integral. El examen no es el instrumento más adecuado para verificar el proceso de aprendizaje del estudiante, ni permite analizar la manera como elabora y reelabora un contenido. Es conveniente hacer la distinción entre acreditación y calificación, si bien las dos cumplen una función institucional y social, en la calificación se manejan escalas y se asignan números al aprendizaje de los estudiantes. La institución y la sociedad adjudican a estas calificaciones un valor que no tienen en sí mismas. Y justamente en la asignación de calificaciones se agudizan los aspectos de justicia y objetividad de la evaluación.

La calificación es el último aspecto por resolver cuando se ha decidido sobre un contenido, las estrategias de aprendizaje, y una concepción de evaluación y sus lineamientos de acreditación. Es necesario decidir antes sobre la acreditación del estudiante y buscar alternativas en el trabajo grupal, para que los estudiantes se responsabilicen del desarrollo de la evaluación general del curso. La asignación de notas suele ser responsabilidad del docente, quien en casos excepcionales puede compartirla con los estudiantes, por ejemplo, cuando se ha realizado un excelente trabajo grupal. Cuando los participantes de un curso han podido realizar una serie de experiencias grupales – a partir de ciertos fundamentos de una noción de grupo que permiten internalizar un esquema de valores, en el que el grupo es una fuente y una experiencia de aprendizaje –, éstos se responsabilizan con acierto y utilizando la autocritica de

la designación de sus calificaciones, claro, se trata de situaciones excepcionales.

El programa del docente es su carta de presentación frente al grupo, es el resultado del trabajo profesional y del ejercicio de la dimensión intelectual de la docencia. El docente es quien asume la responsabilidad de crear determinadas condiciones para favorecer el aprendizaje de los estudiantes. Por medio del programa docente, cada maestro trata de desarrollar una serie de posibilidades profesionales; de esta manera, la actividad docente se convierte en un reto en el que es necesario experimentar, innovar y desarrollar ideas creativas. Existen múltiples posibilidades de innovación en el aula, una innovación puede estar definida por la necesidad de emplear nuevas tecnologías en el salón de clases, lo que indudablemente requiere el apoyo social e institucional.

El docente tiene retos, enfrentarlos echando mano de su formación conceptual en teorías didácticas, psicopedagógicas o sociológicas es indispensable para fundamentar su labor. Dar a la docencia un carácter experimental es un elemento básico para luchar contra la "rutina profesional" y contra la "enajenación" en la profesión docente. La actividad del maestro debe surgir de una vitalidad interna, y es esa virtud interna la que lo puede ayudar a ser un buen maestro a lo largo de su vida. Si en esta profesión es difícil identificar los resultados "objetivos" de nuestro trabajo, porque en realidad "nunca sabemos qué se lleva el alumno de nosotros", sí podemos sentir la gratificación del artesano y del profesional al realizar permanentes innovaciones pedagógicas en el aula.

Se puede afirmar que el programa escolar tiene tres niveles que se vinculan a las necesidades de la institución educativa (currículum) y al desarrollo de los sujetos de la educación (didáctica). Programas que tienen una diferentes estructura entre sí, pero que a la vez guardan una estrecha relación, y cada uno de ellos permite la interpretación que exigen los otros; así, el programa del plan de estudios, el de la institución o de la academia de maestros y el del docente son resultado de una unidad interpretativa que reconoce la heterogeneidad de los procesos escolares y la profesionalización de la docencia.

Definición de los criterios para la acreditación y su papel en los programas

Los objetivos son “los enunciados que indican los tipos de cambios que se buscan en el estudiante; por ello, la forma más útil de enunciar objetivos consiste en expresarlos en términos que identifiquen el tipo de conducta que se pretende generar en el estudiante”. Pueden ser de precisión individual o de grupo, tiempo, lugar, materiales, etcétera.

Se plantean objetivos de aprendizaje como comportamientos de los estudiantes, con la finalidad de clarificar el proceso para su acreditación y asignación de calificación en un curso.

El docente tiene que recordar permanentemente qué es lo que lo mantiene en la docencia, éste es un punto importante para mantener su “vitalidad interna”, para poder comunicarse con sus estudiantes con la pasión que requiere su profesión.

Una alternativa en la elaboración de programas sería establecer algunos lineamientos de acreditación, dejando abierto el tema de los objetivos para que los docentes y alumnos tuvieran claro cuáles son sus propósitos frente al desarrollo de un curso escolar.

Acreditación escolar y evaluación, la primera se expresa en el procedimiento que culmina en la calificación que se asigna a un estudiante al término de un curso escolar y que permite decidir si ha aprobado o no, mientras que la segunda hace referencia a la comprensión del proceso que permitió determinados aprendizajes.

Las directrices para la acreditación pueden elaborarse a partir de tres criterios generales:

- Que tengan en cuenta los productos o resultados del aprendizaje
- Que reflejen el nivel de integración del objeto de estudio
- Que tengan distinto grado de especificidad de acuerdo con el tipo de programa al que respondan.

Al estudiar y analizar los productos de aprendizaje se puede inferir, por ejemplo, lo que un sujeto conoce acerca de determinado tema, sin embargo, nunca se puede garantizar que el sujeto “conozca” de un tema en determinado nivel, ni mucho menos analizar el “proceso cualitativo” de aprendizaje que tiene un sujeto. Este proceso se describe, no se califica.

En los programas han de establecerse determinados productos de aprendizaje como prueba de lo acontecido en un curso.

Los programas deben responder a un claro análisis de los contenidos que lo estructuran.

Lo metodológico: Tema central del debate didáctico y la responsabilidad docente

Método en general es todo proceder ordenado y sujeto a ciertos principios o normas, para llegar de una manera segura a un fin u objetivo que de antemano se ha determinado.

El aspecto metodológico también constituye una articulación entre formas de aprendizaje y teorías del aprendizaje, desde las cuales se efectúa una explicación de los tipos de procesos que acompañan el aprendizaje; cuando este aspecto se considera un elemento definitorio del problema, se procura buscar el “modelo” desde el que cualquier contenido pueda ser enseñado. El método se traduce en actividades concretas que realizan el docente y sus alumnos, y en él se requiere efectuar un debate sobre aquello que realmente hacen maestros y alumnos, lo que marca el retorno a lo cotidiano, como elemento desde el cual se puede efectuar la problematización didáctica.

Resulta difícil que la mejora del aprendizaje reclama mejorar los sistemas de enseñanza, y que esta tarea se encuentra estrechamente vinculada a lo que los docentes pueden realizar en el salón de clases. Sería importante regresar a la mirada tradicional de la didáctica, que resaltaba el valor del método y el papel profesional que el docente desempeña en su definición; no para desligar el método del contenido, de las aportaciones de otras disciplinas (psicología, sociología, antropología), ni de las institucionales y personales de maestros y

alumnos, sino para reconstruir y potenciar sus posibilidades en el marco de las exigencias educacionales de fin de siglo.

La metodología de la enseñanza se construye en la síntesis de la elaboración conceptual y la experiencia educativa, marcada por la creatividad y la sensibilidad del docente.

Las actividades deben responder a una perspectiva estructurada, de suerte que unas se apoyen en otras y permitan el desarrollo gradual y pausado de procesos cognoscitivos en los estudiantes. Por ello se propone que se organicen actividades de aprendizaje que posibiliten la asimilación de la información, frente a otras que permitan su organización. No se trata sólo de dominar o promover ejecuciones, sino, fundamentalmente, de atender a procesos de construcción de la información y a exigencias psicológicas de apoyar estos procesos en una experiencia, esto es, en una vivencia reflexionada por parte del sujeto.

Repetir y memorizar no son elementos básicos. Una vivencia puede adquirir un nuevo significado como experiencia incluso mucho tiempo después.

El docente, al organizar sus estrategias de enseñanza, necesitaría pensar respecto del tipo de experiencia al que desea acercar a sus estudiantes.

Como ya lo habíamos contemplado, Hilda Taba propone que los temas de una unidad se aborden a partir de una secuencia de aprendizaje que responde a cuatro tipos de actividades y estos son:

1. Actividades de introducción.- Tienen por objeto crear un clima intelectual para que el estudiante pueda retraer la experiencia y la información que tiene en relación con la temática que se va a abordar. Se trata de actividades sencillas que se basan en una función de recuerdo o que permiten pensar en un problema que le da sentido a la información. Estas actividades se basan en la inventiva docente para idear problemas virtuales que representen un reto para el conocimiento de los estudiantes y que muestren en algunos casos una dimensión "aplicable" de dicho conocimiento. Se pueden encontrar casos prácticos en donde el conocimiento adquiere sentido.

2. Actividades de desarrollo.- Contribuyen a que el estudiante entre en contacto con una información; para su ejecución se puede acudir a las múltiples formas de acceso a la información: clase expositiva o conferencia magisterial; clase interrogatorio, en la que se va incorporando nueva información, exposiciones hechas por grupos de estudiantes o por invitados; lectura comentada o trabajo de biblioteca; exhibición de temas que incorporen nuevas tecnologías como vídeo, los programas tutoriales, o los simuladores de cómputo. Una característica central de este tipo de actividad es que cada sujeto entra en contacto con una nueva información.

3. Actividades de generalización.- Exige que los estudiantes coordinen sus ideas, las reformulen en sus propios términos y efectúen comparaciones y contrastes; en ocasiones, un debate entre los propios alumnos permite confrontar las ideas propias o construir argumentos para sostener una posición; estas actividades dan pie a un momento de información previa y la nueva información que se ha obtenido, y hacen posible el momento que los autores constructivistas denominan construcción de conceptos y procedimientos.

4. Actividades de culminación.- Tienen por objeto la aplicación de la información en la resolución de problemas (reales o inventados), de suerte que se creen las condiciones para organizar la información y construir síntesis conceptuales. Se le pide al estudiante que resuelva un ejercicio.

Habría que cuidar que esta resolución de ejercicios no implicara sólo la mecanización de estrategias o de aplicación de técnicas; el mecanismo resolver supone algo más: la posibilidad de enfrentar nuevas situaciones, de emplear la información en otro tipo de casos, en algún momento puede ser útil saber cómo tener acceso a nuevas informaciones.

Este modelo de actividades posee una sólida consistencia conceptual, aunque reconocemos que el empleo de esta posibilidad de estructuración de las actividades de aprendizaje depende de la formación y la sensibilidad pedagógica de cada docente, así como de las condiciones particulares de la

asignatura que imparte, la edad de los estudiantes y el número de alumnos que conforman el grupo escolar.

Es responsabilidad de cada docente releer estas propuestas, con el fin de tener un espacio personal de reflexión en el que elabora una propuesta metodológica de trabajo para cada uno de sus grupos escolares. Lo que se debe tener presente es que cada grupo escolar reclama estrategias particulares que a cada docente corresponde generar.

Categorías que permiten ayudar a definir el papel del docente frente a la metodología de la enseñanza, desde una perspectiva didáctica:

- La ansiedad creadora.- En ocasiones, en el desarrollo de una clase, el docente percibe que las actividades que se están realizando no contribuyen al logro de las expectativas formuladas; en el conflicto circunstancial que experimenta, sobre todo cuando tiene claridad sobre el tipo de proceso que desea propiciar en sus alumnos, el maestro se encuentra en condiciones de suspender el tipo de actividades realizadas e idear, sobre la marcha de esa sesión de clase, otro tipo de estrategia.
- La pasión por enseñar.- Reclama que todo docente examine profesionalmente los resultados y los obstáculos de su labor, que analice cuáles son los aspectos que tiene que modificar en su método de trabajo, con la finalidad de mejorar la enseñanza.
- La dimensión intelectual del trabajo docente.- En la profesión docente es necesario determinar hasta dónde pueden existir prescripciones curriculares formales y hasta dónde corresponde al docente presentar su propia interpretación del contenido, de la cultura y de las estrategias de aprendizaje. Corresponde al docente elaborar determinadas síntesis de información. El docente es un intelectual crítico-transformativo.

Esta situación demanda una redefinición de la formación del docente, que ponga énfasis en su formación conceptual en el saber educativo y, específicamente, en el didáctico.

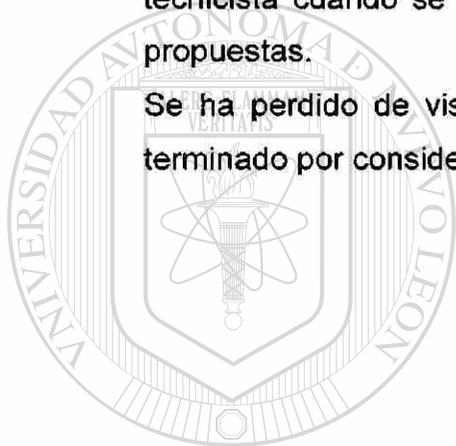
Corresponde a cada docente como profesional de la educación y de la enseñanza, analizar hasta dónde puede intervenir con los instrumentos técnico-

profesionales que tiene a su disposición – y los que pueda generar -, en la creación de mejores condiciones para el aprendizaje.

Existen muchas formas de colaborar en el aprendizaje en grupo: la realización consistente y amplia de lecturas, la propuesta de enfoques de discusión o de interpretación de un material, las interrogantes que plantea cada integrante de un grupo escolar son, entre otros, factores de enriquecimiento y potenciación del grupo; sin embargo, las actuales políticas educativas tienden a desconocerlos.

La didáctica pretende orientar los procesos de enseñanza, pero se vuelve tecnicista cuando se niega a analizar los fundamentos conceptuales de tales propuestas.

Se ha perdido de vista la dimensión intelectual del trabajo docente y se ha terminado por considerar que el profesor es un mero ejecutor de programas.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

3.4. Programa para el Profesor

Aspectos Generales.

1. Datos Generales:

Nombre de la Materia:	Sistemas Operativos I
Tipo de Actividad:	Teórica
Duración:	65 sesiones de 50 minutos
Créditos:	5
Horas Clase por Semana:	Distribuidas en 5 días durante un semestre
Carrera:	Ingeniero Administrador de Sistemas
Semestre:	6º semestre

2. Ubicación de la Materia:

Antecedentes:	Estructura de Datos y Organización Computacional
Paralelas:	Informática I
Subsecuentes:	Sistemas Operativos II y Teleproceso

Tipo de Alumnos: Estudiantes de Licenciatura

Número de Alumnos: Variable

Horario: El asignado

Salón: El asignado

Tipo de Mobiliario: Sillas, escritorio

Recursos Físicos y

Materiales: Hardware de desecho, Cartulinas, Pizarrón, Gis

3. Objetivos Generales de Aprendizaje:

Informativos: El alumno deberá la historia de las computadoras, conocer los componentes internos de una computadora, así como su ambiente interno y externo, comprender los conceptos de

sistemas operativos, su relación y manejar la interrelación de cada uno de sus componentes.

Formativos: **Humanos:** El alumno fortalecerá valores tales como son la responsabilidad, ética, honestidad.

Profesional: El alumno desarrollará un sentido de la ética en lo que respecta al área.

Social e Intelectual: El alumno aplicará sus habilidades, destrezas y capacidades combinado con el espíritu de colaboración al efectuar investigación por equipo.

4. Contenidos Temáticos

- **Panorámica de los Sistemas Operativos**
- **Conceptos de Procesos**
- **Bloqueo Mutuo**
- **Almacenamiento Real**
- **Sistemas de Archivos**

- **Entrada/Salida**

- **Casos de Estudio**

5. Metodología de Trabajo

- El maestro utilizará la técnica expositiva durante el curso además de las dinámicas expuestas en el anexo 2.
- Se utilizaran los apuntes del anexo 1.
- Se formarán grupos de trabajo para llevar a cabo una investigación (casos de estudio, uno por equipo) que al final del curso será expuesta por los integrantes del equipo proporcionando memoria del trabajo para el maestro y una para cada equipo. El tema se les proporciona el primer día de clase.
- Se les encomendará un trabajo final individual (la asignación se hace el primer día de clase).

- Se aplicarán dos exámenes parciales en las fechas que marca la institución.
- Respecto de las evaluaciones se les aplican exámenes con preguntas cerradas, relacionar columnas, lectura comprensiva con una serie de conceptos inmersos en el tema de tal forma que el alumno relacione preguntas y conceptos, al dar una leída al examen se le da un repaso efectivo a la información que se tiene, logrando un efectivo aprovechamiento. Lo anterior sobre la base de que tenemos que tener un testimonio escrito del aprovechamiento del alumno, sin embargo en una simple discusión o en una sesión de preguntas frente a grupo, puede verse reflejado si el mecanismo esta trabajando o no.

6. Criterios y Mecanismos para:

A. Calificación: Parciales: 70%

Trabajo de Equipo: 10%

Trabajo Final: 20%

B. Acreditación: • Será requisito para derecho de 1º Examen Parcial un avance del 40% del trabajo final.

• Trabajo Final completo como requisito para tener derecho de 2º Examen Parcial

• 80% de Asistencia

C. Evaluación: Sesiones de preguntas y respuestas moderadas por el maestro posterior a la recapitulación o enlace de los temas.

Más información en el anexo 5

7. Bibliografía Básica y Complementaria

➤ Harvey M. Dietel

Sistemas Operativos

Editorial Addison Wesley, Iberoamericana S.A.

3ª Edición, 1993

- **Morris Mano**
Arquitectura de Computadoras
Prentice-Hall
3ª Edición, 1994
- **Milan Melenkovic**
Sistemas Operativos
Editorial Mc Graw Hill/Interamericana de España S.A.
2ª Edición, 1994
- **Miguel Angel de Miguel Cabello**
Teresa Higuera Toledano
Arquitectura de Computadoras
Editorial Alfa-Omega
1ª Edición, 1997
- **Peter Norton**
Introducción a la Computación
Editorial Mc Graw Hill
1ª Edición, 1995
- **Ariadne B. Sánchez Ruiz**

Apuntes de Sistemas Operativos

- FIME**
1ª Edición, 1997; 2ª Edición 1998
- **William Stallings**
Sistemas Operativos
Editorial Limusa/Megabyte
1ª Edición, 1995
- **Andrew S. Tanenbaum**
Sistemas Operativos Modernos
Prentice Hall Hispanoamericana
1ª Edición 1992

3.5. Programa para el Alumno

Aspectos Generales.

1. Datos Generales:

Nombre de la Materia:	Sistemas Operativos I
Duración:	65 sesiones de 50 minutos
Créditos:	5
Horas Clase por Semana:	Distribuidas en 5 días durante un semestre
Carrera:	Ingeniero Administrador de Sistemas
Semestre:	6° semestre

2. Ubicación de la Materia:

Antecedentes:	Estructura de Datos y Organización Computacional
Paralelas:	Informática I
Subsecuentes:	Sistemas Operativos II y Teleproceso
Horario:	El asignado

Salón: El asignado

3. Objetivos Generales de Aprendizaje:

Informativos: El alumno deberá conocer los componentes internos de una computadora, la historia de las computadoras, así como su ambiente interno y externo, comprender los conceptos de sistemas operativos, su relación y manejar la interrelación de cada uno de sus componentes.

Formativos: El alumno fortalecerá valores tales como son la responsabilidad, honestidad, ética. A su vez, desarrollará un sentido de la ética en lo que respecta al área. Además, aplicará sus habilidades, destrezas y capacidades

combinado con el espíritu de colaboración al efectuar investigación por equipo.

4. Contenidos Temáticos

- **Panorámica de los Sistemas Operativos**
- **Conceptos de Procesos**
- **Bloqueo Mutuo**
- **Almacenamiento Real**
- **Sistemas de Archivos**
- **Entrada/Salida**
- **Casos de Estudio**

5. Metodología de Trabajo

- El maestro utilizará diversas técnicas durante el curso según lo juzgue pertinente.
- Se utilizarán los apuntes de sistemas operativos I.
- Se formarán grupos de trabajo para llevar a cabo una investigación (casos de estudio, uno por equipo) que al final del curso será expuesta por los integrantes del equipo proporcionando memoria del trabajo para el maestro y a cada equipo. El tema se les proporciona el primer día de clase.
- Se les encomendará un trabajo final individual (la asignación se hace el primer día de clase).
- Se aplicarán dos exámenes parciales en las fechas que marca la institución.

6. Criterios y Mecanismos para:

A. Calificación:	Parciales:	70%
	Trabajo de Equipo:	10%
	Trabajo Final:	20%

- B. Acreditación:
- Será requisito para derecho de 1º Examen Parcial un avance del 40% del trabajo final.
 - Trabajo Final completo como requisito para tener derecho de 2º Examen Parcial
 - 80% de Asistencia
- C. Evaluación: Sesiones de preguntas y respuestas moderadas por el maestro posterior a la recapitulación o enlace de los temas.

7. Bibliografía Básica y Complementaria

- Harvey M. Dietel
Sistemas Operativos
Editorial Addison Wesley, Iberoamericana S.A.
3ª Edición, 1993
- Ariadne B. Sánchez Ruiz
Apuntes de Sistemas Operativos
FIME
1ª Edición, 1997; 2ª Edición 1998
- Andrew S. Tanenbaum
Sistemas Operativos Modernos
Prentice Hall Hispanoamericana
1ª Edición 1992

DIRECCIÓN GENERAL DE BIBLIOTECAS

3.6. Planeación Didáctica

1. Panorámica de los Sistemas Operativos

Objetivos:

Informativos: El alumno conocerá las bases, evolución y tipos de sistemas operativos con el fin de comprender las funciones, estructuras y componentes.

Formativos: El alumno analizará pasado y presente de los sistemas operativos y su trascendencia. Lo anterior, mediante la exposición de algunos souvenirs del pasado y aportaciones del presente.

Actividades de Aprendizaje:

A partir de la información generada por el diagnóstico se trabajan las ideas poniendo en el pizarrón algunas de las palabras que se relacionen o que haya que hacer más énfasis, deduciendo un concepto global del grupo y comparándolo con el que se manejará en clase. El maestro expone sus temas para cubrir los objetivos informativos, poniendo de manifiesto el pasado de la computación, su evolución hasta nuestros días y las expectativas para el mañana.

En algunos temas aparte de la exposición verbal se hace uso de hardware para que los alumnos lo conozcan, lo palpen y se explica de una manera tangible el funcionamiento del mismo.

Materiales:

Pizarrón, gis, chips, motherboards, sim's de memoria, drives, tarjetas de vídeo, puertos paralelos y seriales, ratón, procesador (el chip correspondiente al CPU) y hard disk.

Observaciones:

Los materiales electrónicos que se usan como material didáctico ya no

funcionan por lo que su destino en lugar de ser el olvido se les da nueva aplicación y no ocupa inversión. También se implementa el empleo de las dinámicas apropiadas (ver anexo 3).

Tiempo Estimado: 8 sesiones

2. Conceptos de Procesos

Objetivos:

Informativos: El alumno comprenderá la definición de proceso, sus transiciones de estado, la comunicación entre ellos, problemáticas en las que el sistema operativo se ve involucrado.

Formativos: El alumno analizará las principales técnicas para la administración del CPU y de los procesos en su secuencia de ejecución. Todo esto mediante, animaciones, dinámicas, mesas de discusión.

Actividades de Aprendizaje:

Nuevamente, mediante cuestionamiento, buscamos entre los conocimientos pasados del alumno una definición que nos dé información acerca del tema, trabajando las ideas en el pizarrón se anota lo significativo de cada respuesta y se hacen relaciones de conceptos hasta llegar a complementarlas con las del libro. Ya habiendo establecido el concepto se pueden hacer ejemplificaciones con sucesos de la vida cotidiana para hacerlo más claro. El maestro expone sus temas para cubrir los objetivos informativos, poniendo cuidado de las diferencias entre cada uno de los conceptos, definiciones y nociones que en esta unidad deben de manejarse, aquí se enfoca al sistema operativo como administrador de recursos.

Materiales:

Pizarrón, gis, cartulinas con dibujos ilustrativos, marcadores.

Observaciones:

La administración parcial o total de diferentes dinámicas puede ser un marco perfecto para la mejor comprensión de este tema (ver anexo 2).

Tiempo Estimado: 8 sesiones

3. Bloqueo Mutuo**Objetivos:**

Informativos: El alumno comprenderá la definición de bloqueo, sus áreas de investigación, las condiciones que lo propician y aspectos que involucra.

Formativos: El alumno analizará los bloqueos a los procesos y aplazamiento indefinido, esto mediante comparaciones, discusiones y sesiones de preguntas.

Actividades de Aprendizaje:

Nos trasladamos al centro de la ciudad, en un crucero conocido por todos, alumnos y maestro para ubicarnos en lo que sería un bloqueo mutuo (este traslado es imaginario) planteamos todas las variables en el pizarrón, un mapa o "croquis", cuando ya queda establecido lo que pasa en el centro de la ciudad con el tráfico, nos ubicamos en el interior de la computadora (esto también es imaginario) y hacemos las analogías correspondientes. El maestro expone sus temas para cubrir los objetivos informativos, clarificando las condiciones necesarias para que ocurra un bloqueo, las áreas de investigación como son la prevención, evasión, detección y "recuperación" de un bloqueo y los problemas semejantes como lo sería un aplazamiento indefinido. Lo anterior sin perder de vista ejemplos con experiencias vividas por todos.

Materiales:

Pizarrón, gis

Observaciones:

Hay que propiciar un ambiente participativo y de mucha imaginación para ello se puede emplear algunas dinámicas o técnicas de aprendizaje expuestas en el anexo 2.

Tiempo Estimado: 10 sesiones

- * **Obedeciendo las disposiciones de la institución se aplica primer examen parcial.**

4. Almacenamiento Real**Objetivos:**

Informativos: El alumno comprenderá las técnicas para el manejo de memoria real, memoria virtual, conceptos de paginación, segmentación y memoria asociativa.

Formativos: El alumno analizará las técnicas más comunes en las asignaciones y relocalizaciones de programas y datos en memoria principal y virtual. Lo anterior se realiza con la ayuda de animaciones, paradojas, discusiones, etc.

Actividades de Aprendizaje:

En esta unidad hacemos una retrospectiva de la memoria, proyectándola al presente y al futuro. Vemos nuevamente los sim's de memoria y las motherboard. Tratamos de hacer analogías con situaciones cotidianas. El maestro expone sus temas para cubrir los objetivos informativos, se define la organización y la administración del almacenamiento real y virtual, técnicas para un mejor aprovechamiento de este recurso y como se lleva a cabo la traducción de direcciones y el intercambio de memoria.

Materiales:

Pizarrón, gis, sim's de memoria, motherboard, ver anexo 3

Observaciones:

Esta unidad es la más difícil de exponer, ya que no se dispone de algo tangible para demostrar lo que se trata, se requiere de mucha imaginación. Sin embargo, se pueden usar algunas técnicas de aprendizaje (anexo 2) que pueden ser útiles, además de algunas animaciones que pueden correr en la computadora descrita en el anexo 3.

Tiempo Estimado:10 sesiones

5. Sistemas de Archivos**Objetivos:**

Informativos: El alumno comprenderá el uso de los archivos, sus propiedades; la utilización de directorios, su organización, sus propiedades, las operaciones que se pueden llevar a cabo con ellos; la implantación de un sistema de archivo, la seguridad de la información y mecanismos de protección que debe de tener un sistema de archivos.

Formativos: El alumno analizará la forma de almacenar grandes cantidades de información, que no se pierda hasta que el proceso que la generó termine su ejecución y que dos o más procesos tengan acceso concurrente a la información. También serán analizadas las técnicas para salvaguardar la integridad de la información y del equipo.

Actividades de Aprendizaje:

En esta unidad el alumno ya trae bases de otras materias por lo que hay más participación y más inquietudes, se pueden generar discusiones de experiencias que ellos han tenido y se pueden conjugar exposición y participación. Se pueden “desmenuzar” cada uno de los temas, entrar en discusión, establecer definiciones, analizar estrategias.

Materiales:

Pizarrón, gis

Observaciones:

En algunos temas de esta unidad se requieren conocimientos de historia, principalmente de cuestiones bélicas, estrategias y de seguridad. Se usan dinámicas de aprendizaje (ver anexo 2) además de reforzamiento a su sentido de responsabilidad, ética y moralidad.

Tiempo Estimado: 13 sesiones**6. Entrada/Salida****Objetivos:**

Informativos: El alumno comprenderá los principios del hardware y software de E/S

Formativos: El alumno analizará la forma de controlar los dispositivos de entrada/ salida de la computadora.

Actividades de Aprendizaje:

En esta unidad se profundizan algunos temas que el alumno ya conoce, se hace uso del hardware en "desuso" que hemos estado usando a través del semestre, solo que nos abocamos a los drives, compact disk, hard disk abierto y diskettes desmantelados para hacer más tangibles los conceptos que el maestro expone para cubrir los objetivos informativos. Se ponen a consideración del alumno algunos tópicos de planificación, esquemas de interconexión y otros para que el mismo logre discernir qué es lo conveniente, cuándo y porqué.

Materiales:

Pizarrón, gis, hard disk abierto (están cerrados al vacío), drives de 3 1/2" y 5 1/4", diskettes

Observaciones:

Hay que propiciar un ambiente participativo y de mucha imaginación para ello se pueden utilizar las dinámicas del anexo 2.

Tiempo Estimado: 7 sesiones

7. Casos de Estudio**Objetivos:**

Informativos: El alumno conocerá y comprenderá las características principales de cada sistema operativo, objeto de su investigación.

Formativos: El alumno analizará las características de los diferentes sistemas operativos que hay en el mercado de mayor aplicación actualmente.

Actividades de Aprendizaje:

Estos casos de estudio fueron investigados por equipos integrados por alumnos que expondrán su material, darán memoria a los demás equipos y contestarán dudas que se les pregunten, en este caso el maestro es moderador e intervendrá para hacer aclaraciones o para abundar más en algún tema que pueda causar confusión.

Materiales:

Pizarrón, gis, cartulinas, hojas de rotafolio, acetatos, microcomputadora (algunos alumnos hacen su presentación en Power Point y traen su propia micro), marcadores.

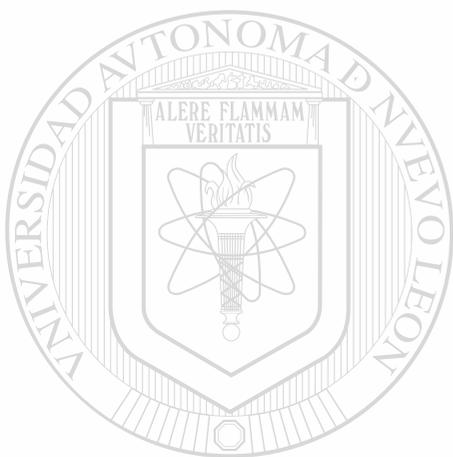
Observaciones:

Este trabajo tiene requerimientos mínimos, pero se hace la aclaración que la profundidad del tema será de acuerdo a su inquietud y sus ganas de

prepararse para el otro semestre.

Tiempo Estimado: 10 sesiones

- * **Obedeciendo las disposiciones de la institución se aplica el segundo examen parcial.**



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

Capítulo 4

Recursos Materiales

Para que se lleve a cabo este proyecto se requiere del siguiente material didáctico.-

- Pizarrón
 - Gis
 - Papel cartoncillo o marquilla
 - Marcadores
 - Televisión a color (preferentemente mayor a 20")
 - Videocassetera VHS
 - Videocasete de Película TRON
 - Hardware dañado y/o desconfigurado
 - Microcomputadora, con monitor a color (preferentemente mayor a 14")
 - Software adecuado (pueden ser simuladores, animaciones o bien algún software orientado al desarrollo del concepto de realidad virtual)
 - Office'97 o superior
-
- Rotafolio
 - Acetatos
 - Telebeam
 - Retroproyector de acetatos
 - Hojas para rotafolio
 - Crayones
 - Hojas tamaño carta
 - Diskettes
 - Versiones de software incompatibles con el hardware y/o software.

Capítulo 5

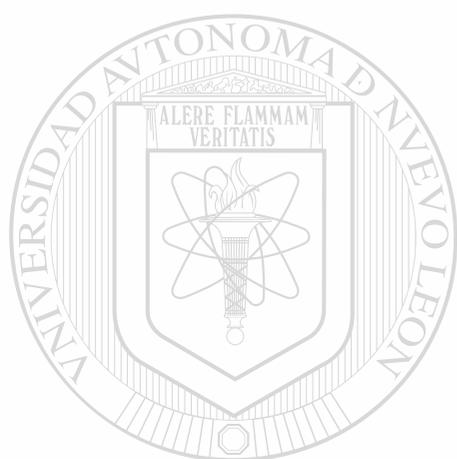
Conclusiones

Al proporcionar al alumno los apuntes de la materia, se le dio una herramienta más de apoyo a un bajo costo, además de motivársele para que se provea a sí mismo de material de estudio para mantenerse actualizado. De este modo el maestro puede dedicar más tiempo para ser explicativo de sus conocimientos al respecto del tema que se este desarrollando. También pueden aplicarse algunas de las técnicas de dinámicas de grupo que vienen en el anexo 2, esto es, se dispone del tiempo para hacer de más provecho la hora/clase. Todo acto de aprendizaje tiene algunas finalidades, explícitas o implícitas, que le dan sentido y significado al aprendizaje, por ejemplo, propiciar el pensamiento crítico o analítico en los estudiantes y este tipo de asignatura es ideal para este propósito.

En mi desempeño como maestro de la carrera de Ingeniero Administrador de Sistemas (esta asignatura pertenece al 6o. semestre), realmente han sido pocos los casos en los que se presenta que el alumno conozca el interior de una computadora y muchos menos los que conceptualizan el cómo se llevan a cabo las funciones internas de una computadora, por lo que al contar con el hardware que me fue donado pudo haber un acercamiento "hombre-máquina" y al mostrárseles por medio de simuladores algunas interacciones de hardware-sistema operativo-software y manejo de memoria, se pudo ser más explícito y se dispuso de tiempo para aclaraciones y discusiones alusivas al tema a tratar. Por lo tanto se logró el objetivo de convertir teoría en práctica, se facilitó el trabajo para el maestro, para el alumno y por demás quedó cubierto el objetivo de que el alumno tuviera amplia visión del funcionamiento de una computadora y sus por menores.

Me he percatado que mi trabajo como docente ha convertido el aula en un ámbito de experimentación, en un laboratorio donde ha sido posible poner a prueba distintas propuestas didácticas y, sobre todo, en un espacio para

percibir las reacciones de mis alumnos y las mías frente a las diversas alternativas que he ido construyendo.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

Capítulo 6

Recomendaciones

Un elemento básico para que un proceso didáctico tenga éxito consiste en tener en cuenta la vida institucional, las dinámicas, las historias y los procesos grupales, de conducción y de comportamiento humano, en las diversas instituciones.

El gran olvido de las propuestas pedagógicas sigue siendo el desarrollo de procesos de pensamiento, de formas de construcción de la información, lo que finalmente lleva a la elaboración de síntesis personales sobre procesos complejos. Desde este punto de vista, el hombre, como sujeto de la educación, queda atrapado en una pseudocultura que lo enajena, le impide desarrollarse culturalmente en relación con su medio y, fundamentalmente, que le niega su capacidad de reflexionar acerca de él, para actuar en pro de su mejoramiento.

Por lo anteriormente expuesto, en este proyecto se decidió tomar de los elementos que conjugan la propuesta de la escuela activa o nueva:

- La lectura de diversas fuentes como pueden ser: libros de diferentes autores, revistas orientadas, manuales, manuales en línea, CBT (Computer Based Training), VBT (Video Based Training), IBT (Internet Based Training)
- La sensibilidad para percibir los resultados de su trabajo en el aula
- El atrevimiento al elaborar estrategias metodológicas diferentes, tanto en la forma como seleccionaron y organizaron los contenidos, como en las propuestas metodológicas que construyeron para ello.

Así, la experiencia que los docentes obtienen en su trabajo en el aula, al ser objeto de reflexión –o, mejor aún, de un documento –, ofrece elementos fundamentales para la elaboración del programa institucional o de la academia.

Se requiere que el contenido, tanto en los programas de estudio como en el trabajo del aula, se presente al estudiante con una mínima estructura interna.

El aprendizaje se explica, fundamentalmente, como un proceso de construcción de ideas y conceptos. El conocimiento no se crea ni se reproduce, sino que se construye, o se conquista. “Los detalles, a menos que se coloquen dentro de un

patrón estructural, se olvidan rápidamente”. No es nuestro deseo que esto suceda por lo que es deber del docente estar actualizado en cuanto a tarea, tema, técnica y dinámica y considerar los objetivos que pretendemos, la razón de ser un docente, de tener un horizonte integral y amplio para que todo esto se dé en forma propicia. Estamos llenos de buenos deseos, de grandes proyectos, pero estamos al frente de seres humanos y al proyectar solemos olvidarlo.

El problema metodológico se considera un problema docente y los planteamientos que la didáctica puede elaborar sólo son indicativos en la tarea que puede realizar un profesor.

El método constituye un eje, el sentido de la labor profesional del maestro. Esta propuesta debe ser tal que “facilite”, “posibilite un clima adecuado” y “sea una invitación y no un obstáculo” para el aprendizaje. El método tiene sentido si contribuye a que el estudiante se comprometa con el aprendizaje y desarrolle destrezas que le permitan avanzar en el mismo. A partir de la construcción de una propuesta metodológica, el maestro intenta crear condiciones para que todos los alumnos aprendan. Nadie debe quedar fuera de la posibilidad de aprender, porque el docente tiene que adecuar las condiciones metodológicas a los procesos de los estudiantes.

En otro orden de ideas se puede recomendar que el salón cuente con el equipo y la infraestructura adecuada para el retroproyector, la microcomputadora, etc. Y por último sería muy conveniente tanto para el maestro como para el alumno adecuar los horarios ya que las clases teóricas suelen verse un tanto “pesadas” a ciertas horas del día, esto por más esfuerzos y proyectos que se hagan.

Bibliografía

- ✓ Allende Carlos María de
Evaluación Educativa Calidad en la Educación
ANUIES, 1990
- ✓ Ángel Díaz Barriga
Didáctica y Curriculum
Edición corregida y aumentada
Editorial Paidós Educador
1ª Edición 1997
- ✓ J. Brunner
Aprendizaje por experiencia directa y por experiencia mediatizada
Perspectivas, vol. III, No. 1, Madrid, UNESCO (1973)
- ✓ Harvey M. Dietel
Sistemas Operativos
Editorial Addison Wesley, Iberoamericana S.A.
3ª Edición, 1993
- ✓ Morris Mano
Arquitectura de Computadoras
Prentice-Hall
3ª Edición, 1994
- ✓ Milan Melenkovic
Sistemas Operativos
Editorial Mc Graw Hill/Interamericana de España S.A.
2ª Edición, 1994
- ✓ Miguel Angel de Miguel Cabello
Teresa Higuera Toledano
Arquitectura de Computadoras
Editorial Alfa-Omega
1ª Edición, 1997

- ✓ Peter Norton
Introducción a la Computación
Editorial Mc Graw Hill
1ª Edición, 1995
- ✓ Jean Piaget
Educación e instrucción
Buenos Aires, Proteo (1970)
- ✓ Ariadne B. Sánchez Ruiz
Apuntes de Sistemas Operativos
FIME
1ª Edición, 1997; 2ª Edición 1998
- ✓ William Stallings
Sistemas Operativos
Editorial Limusa/Megabyte
1ª Edición, 1995
- ✓ Andrew S. Tanenbaum
Sistemas Operativos Modernos
Prentice Hall Hispanoamericana

-
- ✓ Hilda Taba
Elaboración del currículo
2ª Edición Buenos Aires
Editorial Troquel (1976)
 - ✓ Tomo II de la Enciclopedia Metódica Larousse página 371
Edición 1998

Las siguientes direcciones de Internet:

- ✓ <http://alebrige.escom.ipn.mx/~gruiz/pavel/som/Apuntes.html>
- ✓ <http://www.mcgrawhill.es/McGrawHill/Curricular/bachillerato/elnc1jus.htm>

Listado de Tablas

Semestre Febrero Julio 1993

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	36	33	2	1
M1	2	31	31	0	0
M2	3	40	32	7	1
N2	4	45	42	3	0
N3	5	42	40	1	1
N6	6	24	23	0	1
N6	7	36	12	21	3
Totales		254	213	34	7

Semestre Agosto-Diciembre 1993

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	15	10	2	3
M2	2	34	25	7	2
N1	3	41	29	0	12
N2	4	41	31	0	10
N6	5	12	11	0	1
Totales		143	106	9	28

Semestre Febrero Julio 1994

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	44	27	15	2
M2	2	46	38	8	0
V2	3	39	29	10	0
N1	4	37	25	6	6
N2	5	47	40	1	6
N5	6	28	27	1	0
N6	7	4	3	1	0
Totales		245	189	42	14

Semestre Agosto-Diciembre 1994

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	16	10	3	3
M2	2	29	20	5	4
V2	3	29	18	8	3
N1	4	39	15	6	18
N2	5	41	20	3	18
Totales		154	83	25	46

Semestre Febrero Julio 1995

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	29	23	5	1
M2	2	32	30	2	0
V1	3	32	27	4	1
V2	4	20	20	0	0
N1	5	35	18	10	7
N2	6	40	30	9	1
N4	7	32	24	6	2
N6	8	25	19	3	3
Totales		245	191	39	15

Semestre Agosto-Diciembre 1995

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	24	18	4	2
M2	2	43	34	3	6
N1	3	28	19	7	2
N6	4	31	24	3	3
Totales		126	95	17	13

Semestre Febrero Julio 1996

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	31	29	1	1
M2	2	38	37	1	0
V1	3	34	33	0	1
N1	4	33	25	7	1
N3	5	24	17	6	1
N3	6	37	33	2	2
N6	7	12	9	0	3
Totales		209	183	17	9

Semestre Agosto-Diciembre 1996

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	16	13	3	0
M2	2	27	26	1	0
N1	3	20	18	0	2
N6	4	32	31	0	1
Totales		95	88	4	3

Semestre Febrero Julio 1997

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	30	26	3	1
M2	2	38	34	4	0
V1	3	25	25	0	0
N1	4	22	17	2	3
N2	5	32	26	0	6
N3	6	52	49	0	3
Totales		199	177	9	13

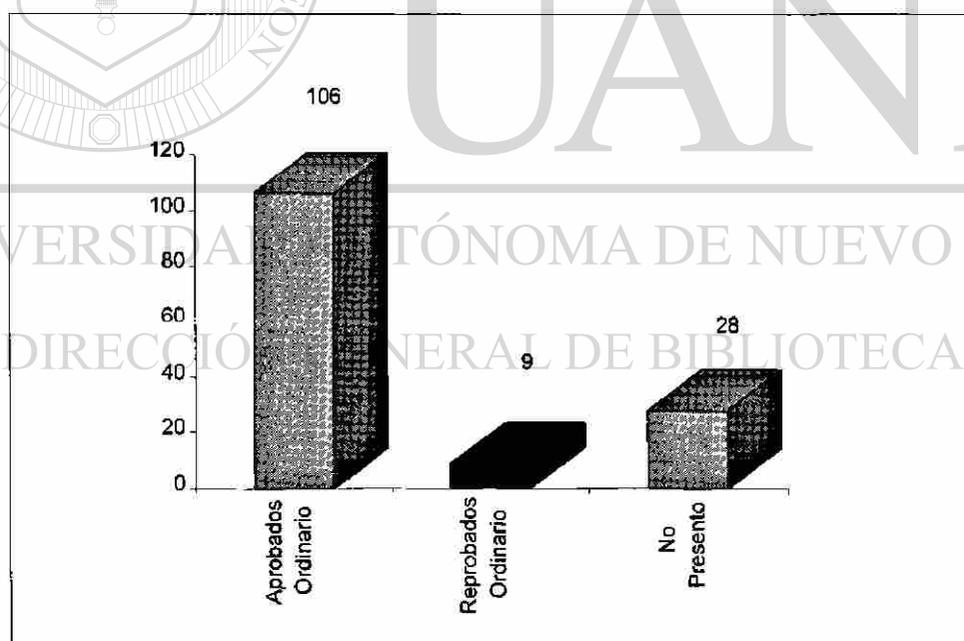
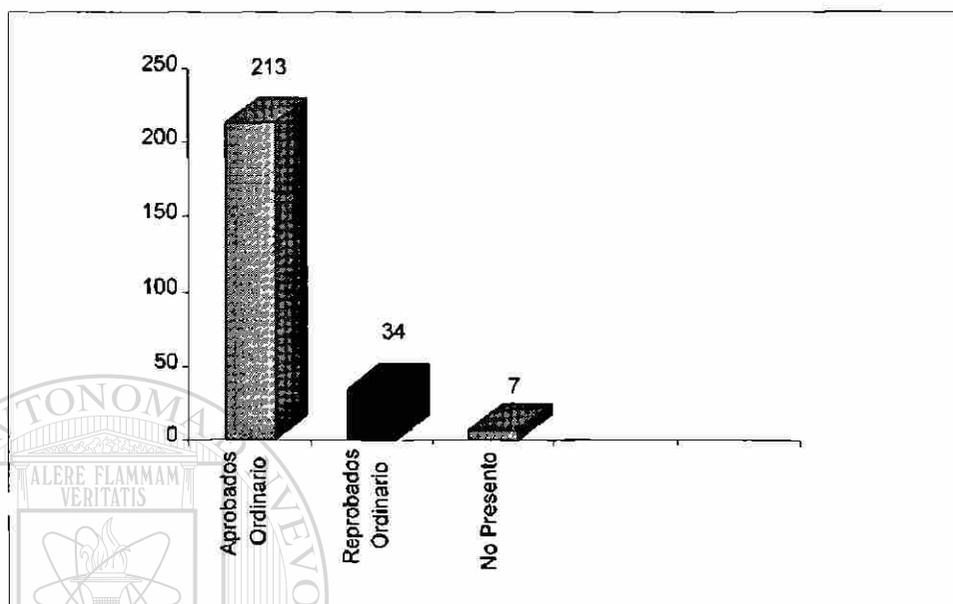
Semestre Agosto-Diciembre 1997

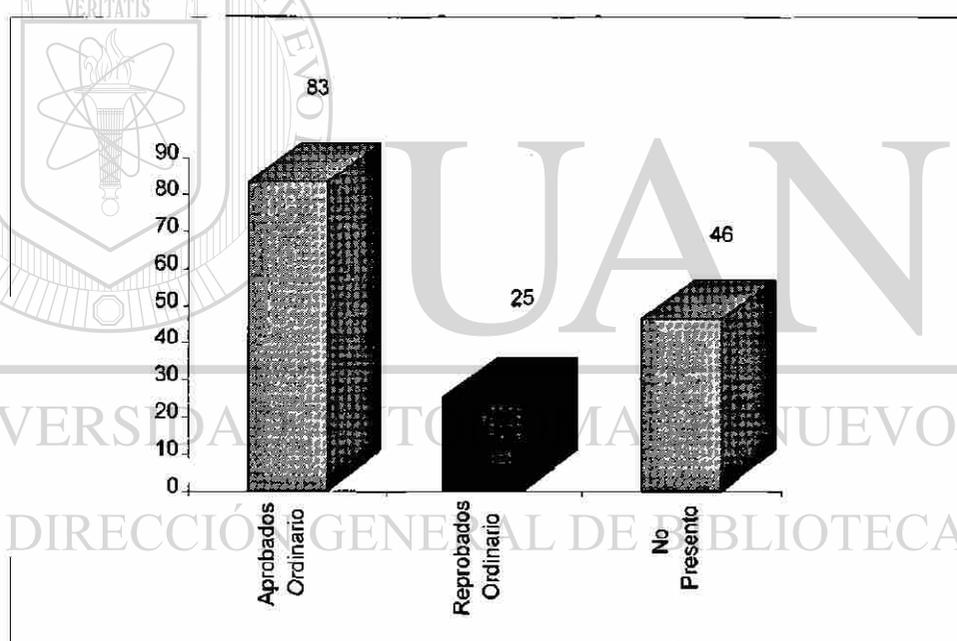
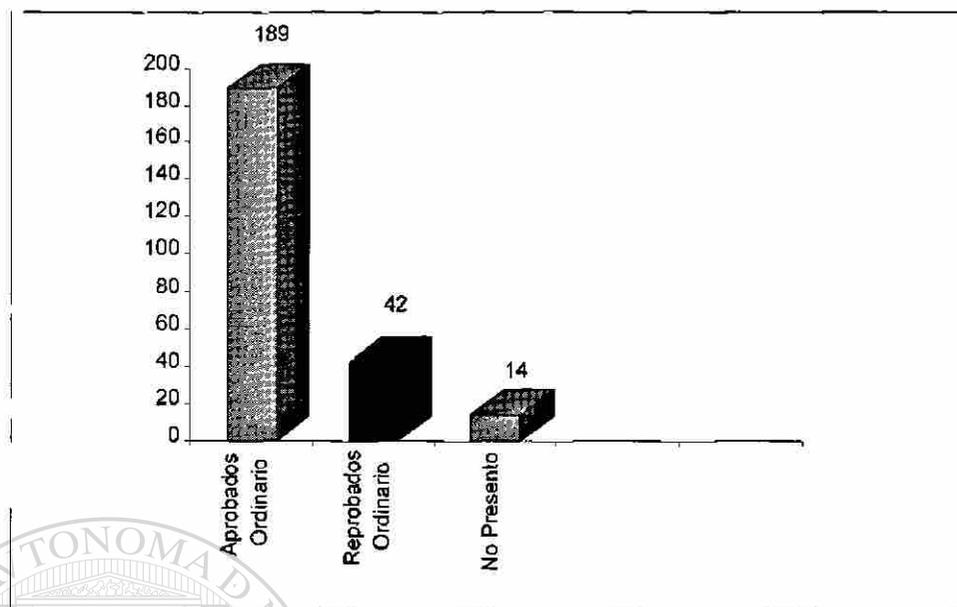
Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	20	16	2	2
V1	2	27	25	1	1
N1	3	19	14	4	1
N2	4	23	14	1	8
N3	5	34	33	0	1
Totales		123	102	8	13

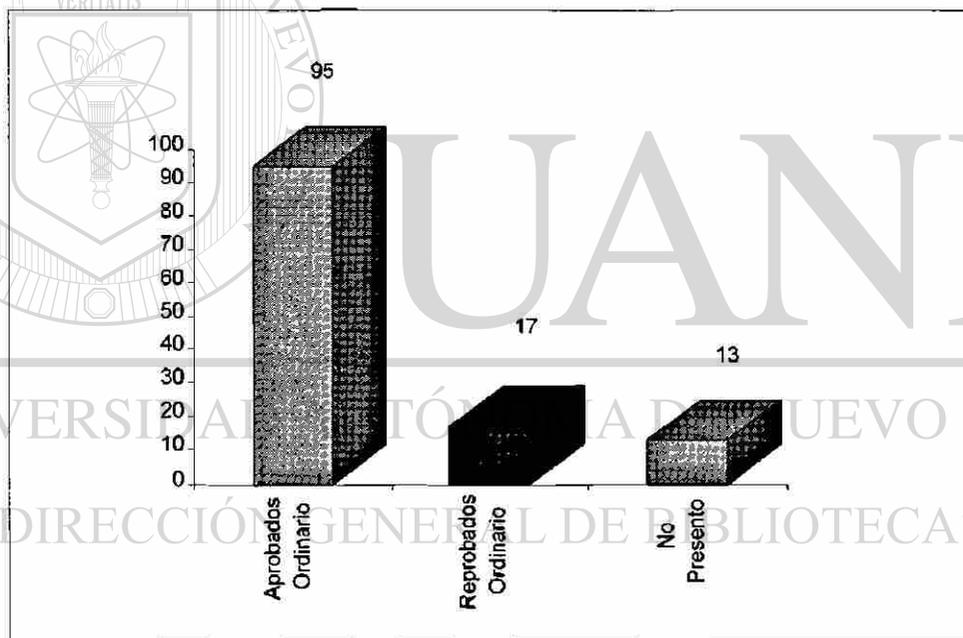
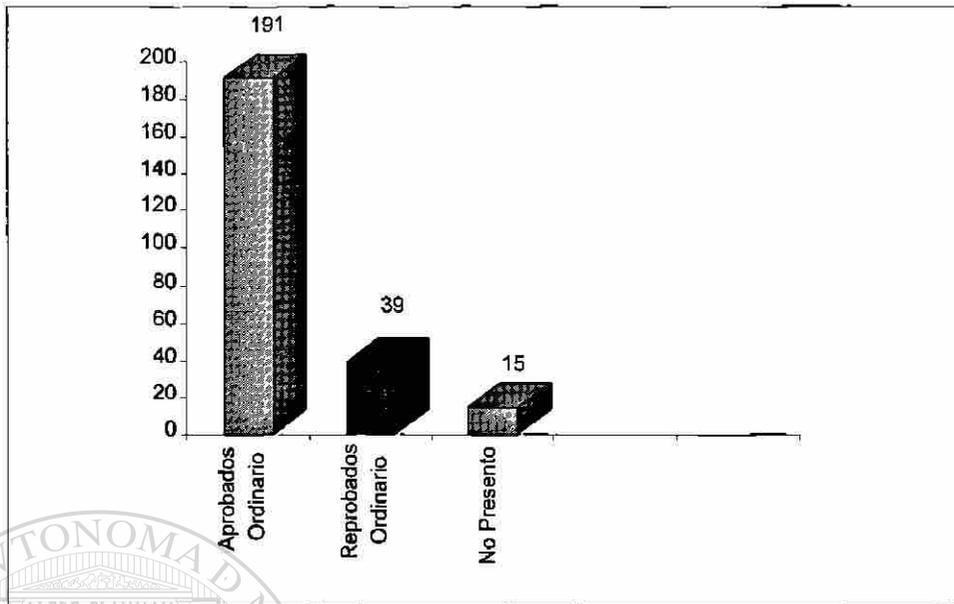
Semestre Febrero Julio 1998

Hora	Grupo	Alumnos	Aprobados Ordinario	Reprobados Ordinario	No Presento
M1	1	23	16	3	4
V1	2	45	44	0	1
N1	3	16	14	1	1
N2	4	31	18	2	11
N3	5	59	50	8	1
N6	6	40	35	3	2
Totales		214	177	17	20

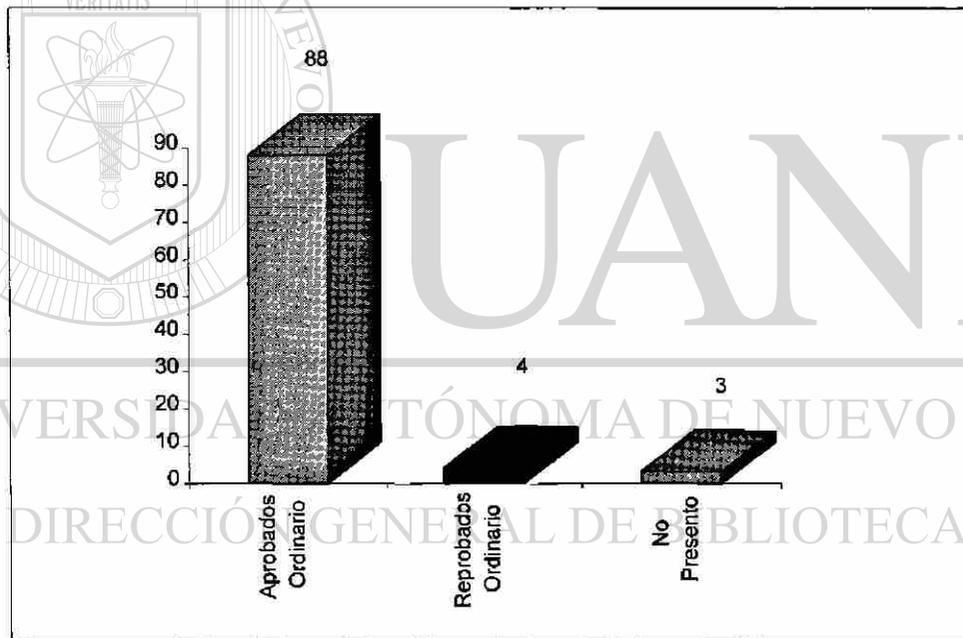
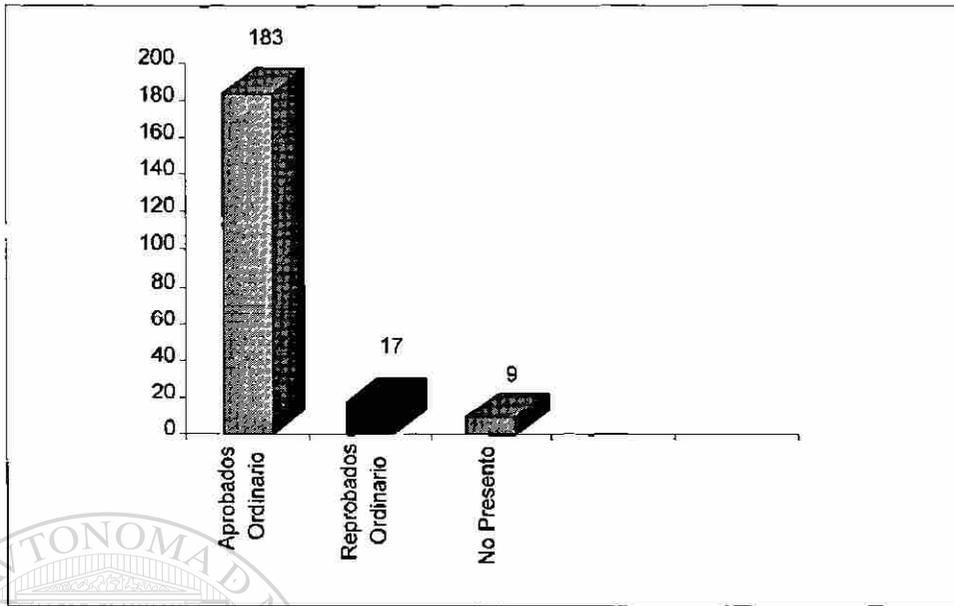
Listado de Gráficas

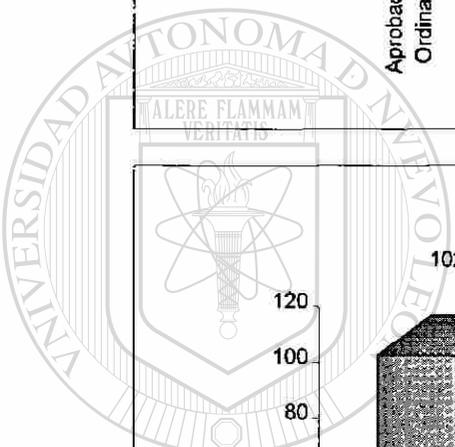
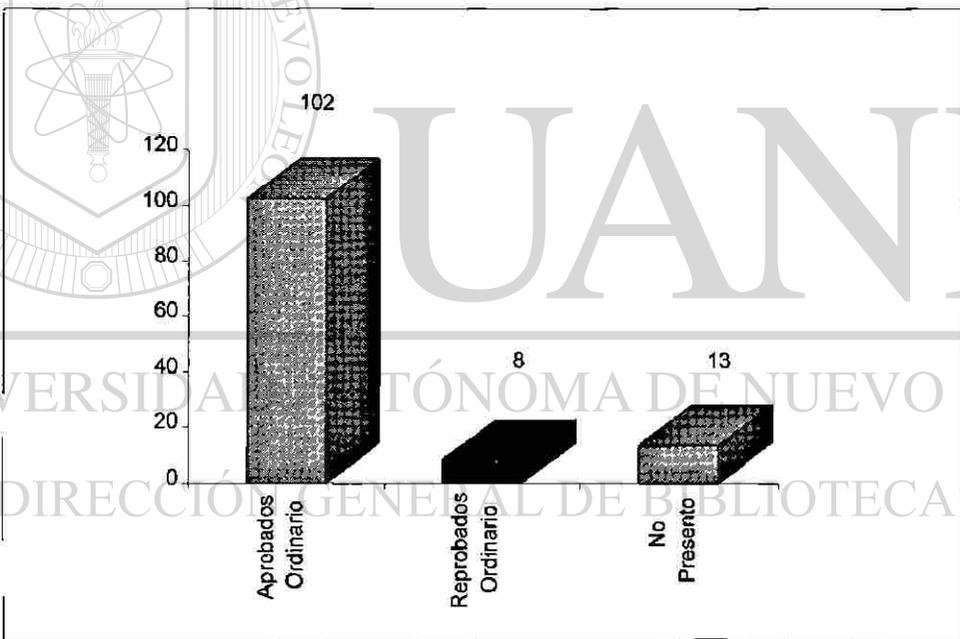
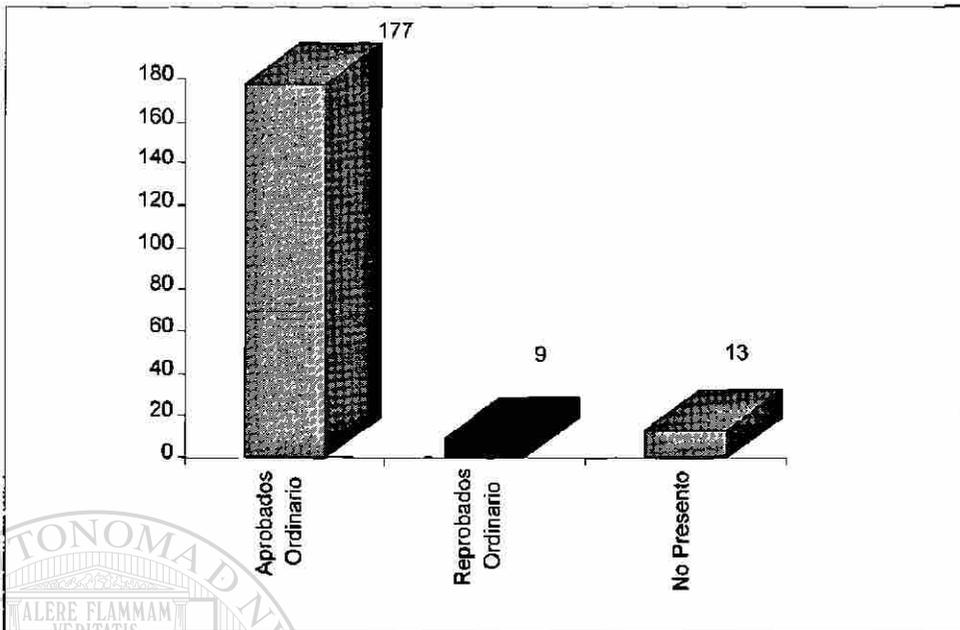






UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
VERITATIS
UANL
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
DIRECCIÓN GENERAL DE BIBLIOTECAS



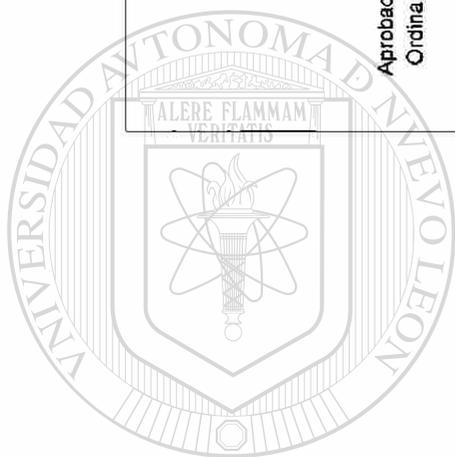
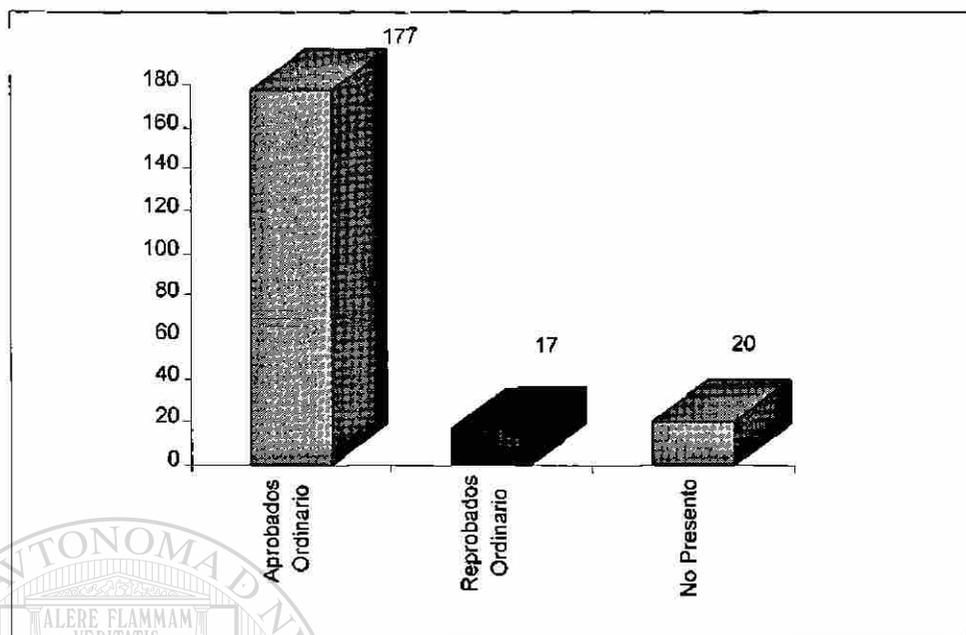


U A N L

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS





UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS



Glosario de Términos

Angiografía axial computarizada.- Computadora de propósito especial para examinar vasos sanguíneos y linfáticos, el sistema vascular en general con fines de diagnóstico.

Chip.-

Circuito integrado.

Cóclea.-

Computadora de propósito especial que se implanta en la cavidad del oído interno con el fin de favorecer en el paciente el sentido del oído.

Cognoscitiva.-

Relativo al conocimiento.

Computer based training.-

Entrenamiento basado en computadora.

CPU.-

Unidad Central de Procesamiento

Epistemológico.-

Estudio filosófico de la ciencia que abarca® la metodología, el problema de la verdad científica y el de las relaciones entre ciencia y tecnología.

Escisión.-

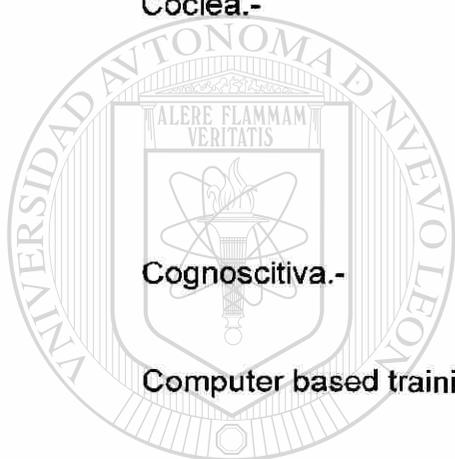
División, separación.

Firmware.-

Software que ha sido copiado en circuitos integrados, usualmente en memorias de solo lectura (ROM).

Hard disk.-

Disco duro, dispositivo de almacenamiento.

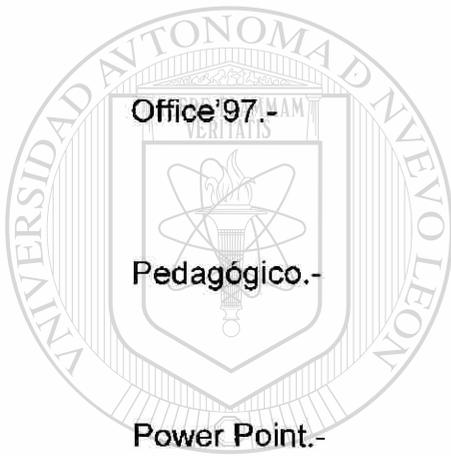


UANL

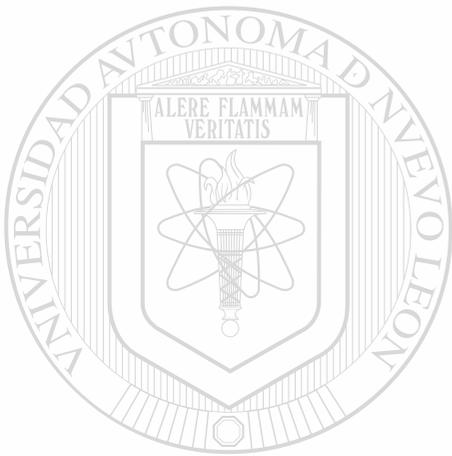
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

Hardware.-	Equipo físico, electrónico, magnético y mecánico.
Internet.-	Red internacional de comunicación en la que se encuentra todo tipo de información.
Internet based training.-	Entrenamiento basado en Internet.
Motherboard.-	Tarjeta principal a la que van conectados todos los dispositivos electrónicos que componen una computadora. Conjunto de paqueterías orientados al trabajo de oficina.
Office '97.-	Ciencia de la educación, método de enseñanza.
Pedagógico.-	Paquete orientado al diseño y elaboración de presentaciones.
Power Point.-	
Resonancia nuclear magnética.-	Computadora de propósito especial que realiza la transmisión de las ondas sonoras por un cuerpo con fines de diagnóstico.
Sim's de memoria.-	Circuitos integrados cuya función es almacenamiento de información temporal.
Software.-	Término genérico para cualquier programa de computadora; instrucciones que hacen que el hardware trabaje.



- Tomografía axial computarizada.- Computadora de propósito especial que realiza el procedimiento que permite hacer radiografías de secciones muy finas del cuerpo o de la cabeza con fines de diagnóstico.
- Video based training.- Entrenamiento basado en video.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

Anexo 1

Apuntes

INTRODUCCION

El objetivo de esta materia es lograr que el alumno tenga conocimientos del funcionamiento, características principales y la evolución de los sistemas operativos. Es fundamental para el Administrador de Sistemas debido a que son precisamente los sistemas operativos los que proporcionan el medio sobre el cual se desarrollan y se da mantenimiento a los sistemas computacionales.

Sistemas Operativos I, analiza la clasificación y características de los sistemas operativos, la administración de procesos, la administración de memoria, los sistemas de archivo, entrada/salida y bloqueos.

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

Unidad I

Panorámica de los Sistemas Operativos

Objetivo de esta unidad.- Durante el desarrollo de esta unidad, se conocerán las bases, la evolución y los diferentes tipos de sistemas operativos, para así comprender las funciones, estructuras y componentes. El alumno deberá comprender las funciones, estructuras y componentes

¿Qué son los Sistemas Operativos?

Los sistemas operativos son ante todo administradores de recursos. El principal recurso que administran es el hardware (dispositivos periféricos, recursos del sistema, etc.)

- los procesadores
- los medios de almacenamiento
- los dispositivos de entrada y salida
- los dispositivos de comunicación y datos

Su principal función es:

- proporcionar la interfaz del usuario
- permitir que los usuarios compartan el hardware entre sí
- evitar que los usuarios interfieran entre sí
- facilidad en la entrada/salida
- recuperarse de errores, etc.



Fig. 1.1 Consideración de acuerdo a Andrew S. Tanenbaum

El sistema operativo lleva la cuenta del estado de cada recurso y decide quien obtiene un recurso, durante cuanto tiempo y cuando; teniendo por objetivo primario incrementar la productividad de un recurso de proceso tal como el hardware del computador, o los usuarios del sistema de computo.

De igual modo el Sistema Operativo proporciona servicios de manera típica en las siguientes áreas:

- ◆ **Creación del programa.-** El sistema operativo proporciona diversas facilidades y servicios, como editores y depuradores, para ayudar al programador en la creación de programas. En general, estos servicios están en la forma de utilería que en realidad no son parte del sistema operativo pero son accesibles mediante él.
- ◆ **Ejecución del programa.-** Necesitan realizarse varias tareas para ejecutar un programa. Las instrucciones y datos deben cargarse en la memoria principal, los dispositivos de E/S y archivos deben inicializarse, otros recursos deben prepararse. El sistema operativo, entonces, maneja todas estas tareas por el usuario.
- ◆ **Acceso a dispositivos de E/S.-** Cada dispositivo de E/S requiere su propio conjunto particular de instrucciones o señales de control, para la operación. El sistema operativo se encarga de los detalles, de manera que el programador pueda pensar en términos de lecturas y escrituras simples.
- ◆ **Acceso controlado a archivos.-** En el caso de archivos, el control debe incluir una comprensión de no sólo la naturaleza del dispositivo de E/S (unidad de disco, unidad de cinta) sino también del formato de archivo en el medio de almacenamiento. De nuevo, el sistema operativo se encarga de los detalles. Además, en el caso de un sistema con múltiples usuarios simultáneos, el sistema operativo puede proporcionar mecanismos de protección para controlar el acceso a los archivos.

- ◆ **Acceso al sistema.-** En el caso de un sistema público o compartido, el sistema operativo controla el acceso a todo el sistema y a los recursos específicos del sistema. La función de acceso debe proporcionar protección de recursos y datos desde usuarios no autorizados y debe resolver conflictos en el caso de disputas por recursos.
- ◆ **Detección de errores y respuesta a ellos.-** Pueden ocurrir varios errores mientras corre un sistema computacional. Esto incluye errores de hardware, internos, externos (errores de memoria, falla, mal funcionamiento de algún dispositivo) y varios errores de software (sobreflujo aritmético, intento de acceso a posición de memoria prohibida e incapacidad del sistema operativo para conceder la solicitud de una aplicación). El sistema operativo se ve precisado a emitir la respuesta que elimina la condición de error, con el menor impacto sobre las aplicaciones que se corren. La respuesta puede variar desde finalizar el programa que causó el error o volver a intentar la operación, hasta solo comunicar el error a la aplicación.
- ◆ **Contabilidad.-** Un buen sistema operativo recopila estadísticas del uso de los diferentes recursos y monitorea los parámetros de ejecución, como tiempo de respuesta. En cualquier sistema, esta información es útil para anticipar la necesidad de mejoras futuras y para ajustar el sistema con el fin de mejorar el rendimiento. En un sistema multiusuario, la información puede usarse para propósitos de facturación. ®

Esto es, provee un ambiente en el cual el usuario puede ejecutar programas de manera conveniente, protegiéndolo de los detalles y complejidades del hardware, es decir, el sistema operativo presenta una interfaz o "máquina virtual" que es más fácil de entender y de programar que una máquina normal.

Personal del Ambiente de Computación.

De manera general, tenemos a:

- ◆ **Usuarios.-** Son clientes del ambiente computacional que utilizan la computadora para organizar un trabajo.

- ◆ **Programadores.-** Se encargan del mantenimiento del sistema adaptando las necesidades de instalación.
- ◆ **Operadores.-** Se encargan de la vigilancia del sistema operativo respondiendo a peticiones para intervenir montando y desmontando cintas o discos, asegurándose de que las impresoras estén cargadas con los formatos adecuados, es decir, desempeñan las funciones no automáticas.
- ◆ **Administradores de Sistemas.-** Establecen las políticas e interfase con el sistema operativo para asegurar que esta política sea adoptada apropiadamente.

Historia

La primera computadora digital fue diseñada por el matemático inglés Charles Babbage (1792-1871) quien gastó parte de su vida y su fortuna en el intento por construir su "máquina analítica". Nunca logró que funcionara adecuadamente ya que era un diseño puramente mecánico y la tecnología de su época no podía producir las ruedas, engranes, levas y demás partes mecánicas con la precisión que él necesitaba. Resulta obvio que esta máquina no tenía sistema operativo.

Evolución del Hardware

Bulbos → Transistores → Circuitos Integrados → Circuitos Integrados a muy gran escala

Durante la 2ª Guerra Mundial se vio la necesidad de suministrar alimentos, pertrechos, armamento, etc. de una forma óptima por lo que había que hacer cálculos muy grandes para lo que se necesitó la precisión y velocidad que podría ser proporcionada por una máquina y esto da origen a:

Generación 0 (Década de los 40's).

- No hay sistema operativo
- Se utiliza lenguaje máquina
- Se introducen programas y datos bit por bit mediante switches mecánicos

- Se introduce la utilización de cintas y tarjetas perforadas
- Se comienza el desarrollo de los Lenguajes Ensambladores para acelerar el proceso de programación.

Generación 1 (Década de los 50's)

- General Motors Research Laboratories implantó el primer sistema operativo en una IBM 701 (a principios de los 50's)
- Ejecución de una sola tarea a la vez
- Se inicio el uso de procesamiento por lotes de secuencia única.
- Se logro cierta fluidez entre los trabajos

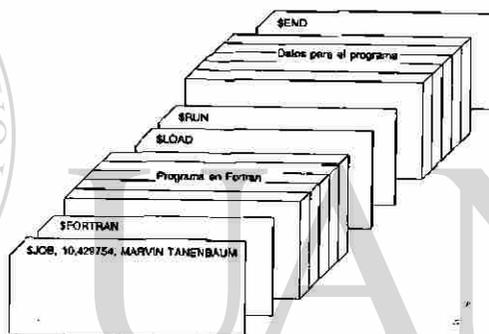


Figura 1.2. Programa en tarjetas perforadas.

Generación 2 (1ª mitad de los 60's)

- Aparece la multiprogramación.
 - Inicia el multiprocesamiento.
 - Aparece el tiempo compartido (Time-Sharing*).
 - Se generan los primeros sistemas de tiempo real.
 - Independencia de dispositivos
- * El concepto de tiempo compartido es una variante de la multiprogramación en la que un CPU atiende simultáneamente los requerimientos de varios usuarios conectados en línea a través de terminales. Ya que los usuarios demoran entre la emisión de un comando y otro, un solo CPU puede atender a cientos de ellos (sabemos que solo uno a la vez). Cuando no hay ningún comando que ejecutar proveniente de un usuario interactivo, el CPU puede cambiar a algún trabajo por lote.

Generación 3 (2ª mitad de los 60's y 1ª mitad de los 70's)

- Mismas problemáticas de la generación dos
- Aparece el sistema 360 de la IBM, también conocido como OS/36
- Computadoras grandes y costosas
- Lentitud y alto porcentaje de error y falla
- Aparece la ingeniería de software.- Debido a que los cambios de personal a menudo hacían desechar sistemas completos
- Así surge un método disciplinado y estandarizado en el que fuera posible construir software confiable, comprensible y fácil de mantener.

Generación 4 (2ª mitad de los 70's hasta nuestros días)

- Aparecen las redes computacionales.
- Aparece el computador personal.
- El hardware disminuye en su costo.
- Se realiza la comunicación alejada geográficamente.
- Se presentan los problemas de seguridad en la información de cifrado y descifrado.

-
- Aparecen los sistemas controlados por menús.
 - Aparecen los sistemas de bases de datos.
 - Se reafirma la familia de computadoras IBM 360 en el mercado.
 - Separación de costos del Software y Hardware
 - Antes se vendía solamente el hardware y el software se daba "gratis" a los clientes.
 - Como el software era regalado no se hacían responsables del mismo.
 - Los vendedores proporcionaban listas de errores a los clientes de ciertas funciones para que no las usaran.
 - **Se presentaron los siguientes cambios:**
 - Se eliminaron las cláusulas donde los vendedores no se responsabilizaban del software.

- Se crean industrias independientes del software (Microsoft, Asthon-Tate, McAffe, etc.)
- Se generaliza la separación de los costos.
- Los usuarios tuvieron mayor posibilidad de seleccionar el software que desearan.
- Se inicio el uso de licencias para software.
- Aparecen las compatibles con IBM.

Tendencias Futuras

1. El hardware bajará aun más de precio.
2. Las velocidades de procesamiento y capacidades de almacenamiento seguirán aumentando.
3. El tamaño físico del equipo seguirá disminuyendo.
4. El multiprocesamiento se hará mas común.
5. Muchas funciones del Sistema Operativo realizadas por software, migrarán al microcódigo.
6. Se fomentará el concepto de familia de computadoras
7. Los sistemas paralelos (computadoras con varios CPU) requieren sistemas operativos capaces de asignar trabajos a los distintos procesadores.

“Fueron los microprocesadores los que cambiaron la forma de en que se consideraba la computación. El microprocesador ha hecho posible crear ciclos de procesador baratos”.

Computación Distribuida

En la actualidad los ciclos de procesador se distribuyen generalmente por toda la organización en las computadoras personales y en las estaciones de trabajo, muchas de las cuales están parados la mayor parte del tiempo. La computación se ha convertido en un fenómeno distribuido mas que centralizado. Esta tendencia es obvia. Para utilizar los ciclos de procesador de un computador central desde una terminal remota, esa terminal necesita estar conectada a la

computadora mediante una línea de comunicación. Las líneas de comunicación más comunes son las telefónicas, que son útiles para lograr transmisiones confiables solo a velocidades relativamente bajas, de hasta unos cuantos miles de bits/segundo. Mas allá de tales velocidades, el ruido de las líneas impide obtener una transmisión rápida y segura. Por muy rápido que procese el computador central la información requerida por los usuarios de las terminales, la información que se intercambia entre el computador central y las terminales remotas no pueden transmitirse en forma confiable a altas velocidades a través de las líneas telefónicas convencionales. La velocidad de las comunicaciones ha sido un serio cuello de botella en el diseño de sistemas de cómputo y comunicación integrados, y es poco probable que tal situación mejore en un futuro próximo.

Solo quedan dos opciones para distribuir la capacidad de computo en los lugares donde se requiera:

1. Tratar de incrementar la velocidad de transmisión para aprovechar mejor la capacidad de un computador central
2. Distribuir computadores dedicados en los lugares de trabajo

Aumentar las velocidades de transmisión es un problema de proporciones enormes; sería necesario modificar por completo la red telefónica mundial. Sin embargo, las redes de computadoras y sistemas distribuidos abren nuevas posibilidades e imponen nuevas obligaciones a los sistemas operativos.

Sistemas tolerantes a fallas.- Son sistemas que pueden funcionar a pesar de la existencia de varios problemas.

Sistemas abiertos.- Son sistemas de computación, de comunicaciones o de ambas cosas, cuyas especificaciones están ampliamente disponibles, aceptadas y estandarizadas.

Los sistemas abiertos tienen varios componentes:

1. Normas de Comunicación Abierta

2. Normas de Sistemas Operativos Abiertos
3. Normas de Interfaz de Usuarios Abiertas
4. Normas de Aplicaciones de Usuarios Abiertas

Hardware, Software y Firmware

Introducción.

Una computadora debe estar formada por:

- Procesador
- Memoria
- Componentes de E/S

Al menos uno o más de estos elementos de cada tipo. Estos elementos básicos interconectados realizarán la función principal de cada computadora: **Ejecutar programas**. Tomaremos cuatro elementos estructurales principales:

Procesador.- Controla la operación de la computadora y ejecuta sus funciones de procesamiento de datos. Cuando existe un solo procesador, con frecuencia se hace referencia a él como unidad central de procesamiento (CPU, Central Processing Unit).

Memoria Principal.- Almacena datos y programas. Es típicamente volátil; se le conoce como memoria real o memoria primaria.

Módulos de E/S.- Mueven datos entre la computadora y su ambiente externo, esto incluye:

- ◆ Dispositivos de memoria secundaria
- ◆ Equipo de comunicaciones
- ◆ Terminales

Interconexión del sistema.- Una parte de la estructura y mecanismos que proporcionan comunicación entre procesadores, memoria principal y módulos de E/S.

HARDWARE

Hardware.- Consiste en los dispositivos del sistema de cómputo

- su procesador,
- sus dispositivos de almacenamiento,
- sus dispositivos de entrada/salida,
- sus conexiones de comunicación.

◆ **Memoria entrelazada.-** Se usa para acelerar el acceso al almacenamiento primario. Normalmente mientras se obtiene acceso a algunas de las localidades de un banco de almacenamiento primario no puede haber otras referencias en proceso. El entrelazado de memoria coloca las localidades de memoria contigua en diferentes bancos de almacenamiento de manera que pueden existir en proceso muchas referencias al mismo tiempo.

◆ **Registro de reubicación.-** Permite reubicar programas de manera dinámica. La dirección base del programa en memoria principal se coloca en el registro de reubicación. El contenido del registro de reubicación se suma a cada dirección generada por un programa en ejecución. El usuario puede programar como si su programa comenzara en la dirección cero. En el momento en que se ejecuta el programa, el registro de reubicación participa en todas las referencias a direcciones, esto permite que el programa resida en localidades diferentes de las que se pretendía que ocuparan.

◆ **Escrutinio (Polling).** - Es una técnica con la que una unidad puede verificar el estado de otra unidad de funcionamiento independiente. La primera unidad verifica si la segunda unidad se encuentra en cierto estado, si no es así, entonces la primera unidad continúa con la tarea que estaba realizando. Es una técnica costosa.

◆ **Interrupciones.-** Por medio de interrupciones una unidad puede obtener de inmediato la atención de otra para informarle un cambio de estado. La interrupción hace que se almacene el estado de la unidad interrumpida

antes de procesar la interrupción. Después se procesa la interrupción y se restablece el estado de la unidad interrumpida.

- ◆ **Buffer.-** Es un área de memoria principal para retener datos durante transferencias de E/S. Mientras se efectúa una transferencia de entrada/salida su velocidad depende de muchos factores relacionados con el hardware de E/S.
- ◆ **Buffer único.-** El canal deposita datos en el buffer, el procesador los procesa y el canal sigue depositando nuevos datos, mientras el canal esta depositando datos no puede hacerse ningún procesamiento, mientras se procesa un dato no pueden depositarse mas datos.
- ◆ **Buffer doble.-** Este hace posible empalmar operaciones de entrada/salida con operaciones de procesamiento, mientras el canal esta depositando datos en el buffer el procesador puede procesar los datos del otro buffer. Cuando el procesador acaba de procesar los datos de un buffer puede procesar los datos del otro buffer mientras el canal deposita los otros datos en el primero, por esta razón, el uso de estos buffer dobles se les denomina buffers alternantes.

-
- ◆ **Dispositivos periféricos.-** Permiten el almacenamiento de cantidades masivas de información fuera de la memoria principal. Los dispositivos de cinta por su naturaleza son dispositivos secuenciales que se escriben o leen sobre una tira larga de cinta magnética. Tal vez los dispositivos periféricos más importantes en lo que se refiere a los sistemas operativos sean los discos magnéticos. Los discos son dispositivos de acceso directo, permiten hacer referencia a unidades de datos individuales sin necesidad de realizar una búsqueda ordenada entre todos los datos contenidos en el disco.
 - ◆ **Cronómetros y Relojes.-** También llamados temporizadores. Un cronómetro de intervalos es útil en un sistema multiusuario para evitar que un usuario monopolice el procesador. Después de un intervalo designado, el cronómetro genera una interrupción para llamar la atención del

procesador, este puede asignarse entonces a otro usuario. Con un reloj horario el computador puede controlar la hora de reloj pared con incrementos de millonésima de segundos o menores.

- ◆ **Operación en línea y fuera de línea.-** Algunos periféricos están equipados para la operación en línea, en la cual están conectados al procesador, o para la operación fuera de línea en la cual son manejadas por unidades de control que no están conectadas al sistema central. Las unidades de control fuera de línea son útiles porque hacen posible manejar dispositivos periféricos sin que esto represente una carga directa para el procesador. Ejemplo: Las operaciones para copiar de cinta a impresora se realizan a menudo con unidades fuera de línea.

- ◆ **Canales de E/S.-** Un canal es un computador de propósito especial dedicado al control de E/S independientemente del procesador central del sistema. Un canal puede obtener acceso directo al almacenamiento primario para guardar o recuperar información. La importancia real de los canales es que aumenta considerablemente la cantidad de actividad concurrente del hardware del computador y libera al procesador de gran parte del control de E/S.

-
- ◆ **Canal selector o simple.-** Son utilizados para la transferencia de datos a alta velocidad entre los dispositivos y la memoria principal. Los canales selectores tienen un solo subcanal y solamente pueden atender a un dispositivo a la vez.

- ◆ **Canales multiplexores.-** Los canales multiplexores tienen muchos subcanales; pueden intercalar muchos flujos de datos a la vez.

- ◆ **Canal multiplexor de bytes.-** Alterna las transmisiones de dispositivos lentos como terminales, impresoras y líneas de comunicación de baja velocidad.

- ◆ **Canal multiplexor de bloques.-** Alterna la transmisión de varios dispositivos de alta velocidad como impresoras láser y unidades de disco.

- ◆ **Robo de ciclos.-** Un punto de conflicto entre los canales y el procesador es el acceso a memoria principal. Como no puede realizarse más que un acceso (a un determinado banco de memoria principal) a la vez, y como es posible que los canales y el procesador deseen obtener acceso simultáneo a memoria principal, los canales suelen tener prioridad y a esto se denomina **robo de ciclos**. Cuando se inicia una operación de E/S, los caracteres se transfieren hacia la memoria principal mediante el robo de ciclos; el canal usurpa temporalmente la ruta del procesador hacia el almacenamiento mientras se transfiere un carácter; el procesador continúa después con su operación.
- ◆ **Estado problema.-** Para los programas de usuarios, el subconjunto de instrucciones que se puede ejecutar en el estado problema impide, por ejemplo, la ejecución directa de instrucciones de entrada/salida; un programa usuario al que se le permitiera realizar entrada y salida en forma arbitraria podría vaciar en pantalla la lista maestra de contraseñas del sistema, imprimir información de cualquier otro usuario o destruir el sistema operativo.
- ◆ **Estado supervisor.-** El sistema operativo se suele ejecutar con una categoría de usuario de máxima confianza en un estado supervisor, en el cual se tiene acceso a todo el conjunto de instrucciones de la máquina.
- ◆ **Multiprocesamiento.-** Varios procesadores comparten un almacenamiento primario común y un solo sistema operativo. En el multiprocesamiento es necesario secuencializar el acceso a una localidad de almacenamiento compartido, de manera que dos procesadores no intenten modificar al mismo tiempo, con la posibilidad de alterar inadecuadamente su contenido.
- ◆ **Acceso directo a memoria.-** Conocido también simplemente como DMA. Una forma de obtener un buen rendimiento en un sistema de cómputo es reducir al mínimo el número de interrupciones. El DMA no requiere más que una sola interrupción por cada bloque de caracteres transferidos en una operación de E/S. Esto es bastante más rápido que el método en el cual el procesador es interrumpido por cada carácter que se transfiere. Cuando un

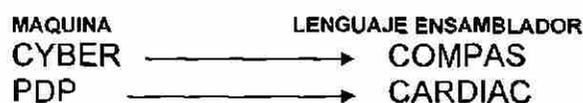
dispositivo esta listo para transmitir un carácter de un bloque, “interrumpe” al procesador. Pero con el DMA no es necesario guardar el estado del procesador; este sufre un retraso, más que una interrupción. Bajo el control de hardware especial, el carácter se transfiere a la memoria principal. Una vez terminada la transferencia, el procesador prosigue su operación. El DMA es una característica de rendimiento particularmente útil en sistemas que manejan gran volumen de transferencias de E/S. El hardware encargado del robo de ciclos y de operar los dispositivos de E/S en el modo DMA se conoce como canal de DMA.

- ◆ **Canalización.-** Es una técnica de hardware utilizada en computadoras de alto rendimiento para aprovechar ciertos tipos de paralelismo en el procesamiento de instrucciones.

SOFTWARE

- ◆ **Software.-** Se compone de los programas con instrucciones en lenguaje máquina y los datos que son interpretados por el hardware. Algunos tipos comunes de software son los compiladores, ensambladores, cargadores, editores de enlace, programas de aplicación, sistemas de manejo de base de datos, sistemas de comunicación y sistemas operativos.

- ◆ **Lenguaje de máquina.-** Es el lenguaje de programación que el computador puede entender directamente.
- ◆ **Lenguajes ensambladores .** - La programación en lenguaje máquina consume mucho tiempo y es propensa al error, los lenguajes ensambladores se desarrollaron para aumentar la velocidad de proceso, de programación y reducir los errores de codificación; se utilizan abreviaturas significativas para reemplazar grandes hileras de 1's y 0's, pero los programas en lenguaje ensamblador deben ser traducidos por la máquina. Los programas ensambladores son dependientes de la máquina.



- ◆ **Lenguajes de alto nivel.**-Permitieron a los usuarios, escribir programas de un modo independiente de la máquina. Con ellos se aumenta en gran medida la velocidad de programación, se hicieron transportables entre diferentes sistemas y permite escribir programas de aplicaciones sin necesidad de ser un experto en la estructura interna de la máquina.
- ◆ **Lenguajes orientados hacia el procedimiento.**- Los lenguajes de alto nivel orientados al procedimiento, son los de propósito general y puede utilizarse para resolver gran variedad de problemas, ejemplo: Pascal, Basic, Cobol, Fortran, etc.
- ◆ **Lenguajes orientados hacia el problema.**- Los lenguajes orientados hacia el problema se usan para un solo fin, ejemplos: SPSS y GPSS.
- ◆ **Programación orientada a objetos.**- Las aplicaciones se han vuelto grandes y complejas y ya no es posible pensar en instrucciones individuales en lenguaje de máquina, ni siquiera en subrutinas pequeñas, como unidades de desarrollo de programas y de cálculo, dando importancia a la programación orientada a objetos. Los objetos son entidades abstractas que encapsulan todos los procedimientos y datos que tienen alguna relación. Tales entidades pueden manejarse como un paquete que se puede manipular de diferentes formas. Los objetos pueden representar hardware, como el CPU, la memoria y los dispositivos; entidades de software, como programas o archivos y varias otras identidades. La programación orientada a objetos se ha popularizado bastante en los proyectos de desarrollo de sistemas operativos.
- ◆ **Spool.**-Se encarga de supervisar la transmisión de salida del almacenamiento principal al canal, del canal al disco duro, y del disco duro a la impresora. Este proceso de SPOOL rompe la asociación entre un programa en ejecución y la operación de dispositivos lentos como una impresora.
- ◆ **Sistemas de control de E/S (IOCS).**- Es un programa supervisor desarrollado para manejar las complejidades de E/S. Evita al usuario el

trabajo de controlar los detalles de la E/S. Estos paquetes son componentes importante en los sistemas operativos actuales.

- ◆ **Compiladores rápidos y sucios.-** Producen rápidamente programa objeto, pero el código generado puede ser bastante ineficiente, en cuanto al consumo de memoria y su velocidad de ejecución.
- ◆ **Compiladores optimizadores.-** Se ejecutan más lentamente, pero producen código de calidad comparable o superior a la del código escrito por programadores expertos en lenguaje ensamblador.
- ◆ **Intérpretes.-** No producen programas objeto, sino que ejecutan directamente los programas fuente. Se ejecutan lentamente si comparamos con un código compilado ya que debe traducir cada instrucción siempre que esta se ejecuta.
 - ◆ **Cargador.-** Es un programa que coloca las instrucciones de un programa y sus datos en las localidades de memoria principal.
 - ◆ **Cargador absoluto.-** Coloca las instrucciones y datos en localidades precisas indicadas por el programa en lenguaje máquina.
 - ◆ **Cargador con reubicación.-** Puede cargar un programa en diversos lugares de memoria dependiendo de la disponibilidad del espacio en el momento de la carga.
 - ◆ **Cargador de enlace.-** Combina todos los programas necesarios y los carga directamente en memoria principal.
 - ◆ **Editor de enlace.-** También combina programas pero además crea una imagen de carga que se conserva en almacenamiento secundario para referencia futura. Son útiles sobre todo en ambientes de producción; cuando se va a ejecutar un programa, la imagen de carga producida por el editor de enlace puede cargarse de inmediato sin la sobrecarga de recombinar segmentos de programa.

FIRMWARE

El concepto de microprogramación se atribuye por lo general al Prof. Maurice Wilkes, su artículo publicado en 1951 presentó los conceptos que forman la

base de las técnicas actuales de microprogramación. En los años 60's se generalizó la microprogramación.

- ◆ **Firmware.**-También conocida como memoria fija. Consiste en los programas en microcódigo almacenados en una memoria de control de alta velocidad. Los programas objeto de uso común colocados en memoria de solo lectura, ROM y PROM también se les conoce como Firmware.
- ◆ **Microprogramación.**- Introduce una capa de programación debajo del lenguaje de máquina de la computadora, como tal, hace posible la definición de instrucciones en lenguaje máquina. Es parte integral de las arquitecturas de computador modernas y es importante para los aspectos de rendimiento y seguridad de los sistemas operativos.
 - ◆ **Microprogramación dinámica.**- Permite cargar fácilmente nuevos microprogramas en la memoria de control desde donde se ejecutan los microprogramas. De este modo, los conjuntos de instrucciones de máquina se pueden variar dinámica y frecuentemente.
 - ◆ **Microprogramas.**- Se ejecutan en una memoria de control de alta velocidad y están formados por microinstrucciones individuales de naturaleza mucho más elemental y de función menos densa que las instrucciones convencionales de lenguaje máquina.
 - ◆ **Microcódigo.**- Una importante decisión de diseño es cuales funciones deben realizarse en microcódigo. El microcódigo representa una oportunidad real para mejorar el rendimiento de la ejecución de un sistema. Mediante la codificación cuidadosa de secuencias de instrucciones de ejecución frecuente en firmware, en lugar de software, los diseñadores han logrado extraordinarias mejoras en el rendimiento.
 - ◆ **Microdiagnóstico.**- Los microprogramas tienen acceso a mas hardware que los programas en lenguaje de máquina, por lo que es posible efectuar una detección y una corrección de errores más amplia y con mayor grado de precisión. Algunos sistemas alternan el microdiagnóstico con las instrucciones en lenguaje de máquina. Esto permite evitar errores potenciales y lograr una operación más confiable, por lo que la

microprogramación puede ser efectiva en el diseño de sistemas tolerantes a fallas.

- ◆ **Emulación.-** Es una técnica para lograr que una computadora se comporte como si fuera otra. El conjunto de instrucciones en lenguaje de máquina de la computadora que se va a emular se microprograma en la máquina anfitriona. Así, los programas en lenguaje maquina de la máquina emulada pueden ejecutarse directamente en la anfitriona. Los fabricantes de computadoras utilizan muy a menudo la emulación cuando presentan nuevos sistemas. Los usuarios habituados a los computadores anteriores pueden ejecutar directamente sus antiguos programas en los sistemas nuevos sin alteración alguna, esto modela el proceso de conversión.

Algunas funciones del Sistema Operativo que suelen realizarse en Microcódigo son:

- Manejo de Interrupciones
 - Mantenimiento de varias estructuras de datos
 - Primitivas de sincronización que controlan el acceso a datos compartidos y a otros recursos
-
- “Conmutación contextual”; esto es, la conmutación rápida de un procesador entre varios usuarios en un sistema multiusuario.
 - Secuencias de llamada a procedimientos y retorno.

Realizar en microcódigo las funciones del sistema operativo puede mejorar el desempeño, reducir los costos de desarrollo de programas y mejorar la seguridad de un sistema.

Unidad II

Conceptos de Procesos

Objetivo de esta unidad.- Durante el desarrollo de esta unidad, se analizarán las principales técnicas para la administración del CPU y de los procesos en su secuencia de ejecución. El alumno deberá comprender qué es un proceso, cuáles son las transiciones de estado de un proceso, cómo se logra la comunicación entre estos conjuntamente con los problemas que enfrenta el Sistema Operativo para lograr la comunicación; cómo decide el sistema operativo que procesos debe efectuar cuando dos o más procesos son ejecutables lógicamente.

Definiciones de “proceso”

El término “proceso” ha sido utilizado a veces como sinónimo de tarea, ha tenido muchas definiciones:

- Es una actividad asíncrona.
 - Un programa en ejecución.
 - El “espíritu animado” de un procedimiento.
 - El “centro de control” de un procedimiento en ejecución.
 - Lo que se manifiesta por la existencia de un “bloque de control de proceso” (PCB) en el sistema operativo.
-
- La entidad a la que se asignan los procesadores
 - La unidad despachable.

Un programa es una entidad inanimada, solo cuando un procesador le infunde vida, se convierte en la entidad activa que se denomina proceso.

Estados de un Proceso

Se dice que un proceso esta en estado de ejecución si tiene asignado el CPU.

Se dice que un proceso esta en estado listo si pudiera utilizar un CPU en caso de haber uno disponible. Un proceso esta en estado bloqueado si esta esperando algún evento antes de proseguir su ejecución.

Transiciones de Estado de los Procesos

Cuando se admite una tarea en el sistema, se crea el proceso correspondiente y se inserta normalmente al final de la lista de procesos listos. El proceso se

desplaza poco a poco hacia el frente de la lista de procesos listos, a medida que los procesos que se encuentran antes que él completan su turno de uso del CPU.

Cuando el proceso llega al principio de la lista, se le asigna el CPU cuando este queda disponible y entonces se dice que hay una transición de estado del estado listo al estado de ejecución. A la asignación del procesador al primer proceso de la lista de procesos listos se le llama despacho.

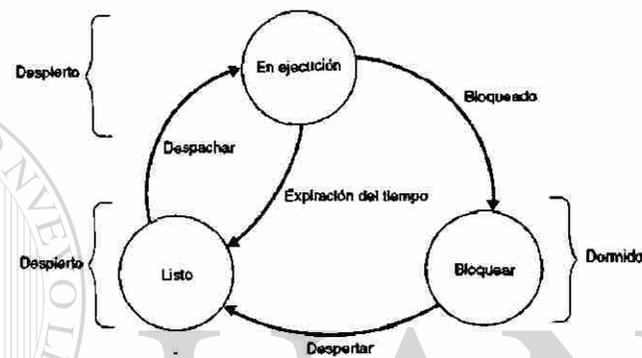


Fig. 2.1. Diagrama de tres estados básicos de los procesos

Despachar(x_proceso):Listo → Ejecución

Para evitar que un proceso monopolice el sistema, en forma accidental o intencional, el sistema operativo utiliza un reloj de interrupción por hardware (también llamado cronómetro de intervalos) para que las tareas de ese usuario se ejecuten durante un intervalo específico de tiempo o cuanto (quantum). Si el proceso no libera voluntariamente el CPU antes de que expire el cuanto, el reloj genera una interrupción, haciendo que el sistema operativo retome el control. El sistema operativo envía al proceso que se estaba ejecutando a la lista de procesos listo y procede a ejecutar el primero de la lista de procesos listos.

Tiempo Expirado(x_proceso):Ejecución → Listo

Si el proceso que se esta ejecutando inicia una operación de entrada/salida antes de que expire su cuanto, libera voluntariamente al CPU, es decir el proceso se bloquea a sí mismo esperando que se complete la operación de entrada/salida.

Bloquear(x_proceso):Ejecución \longrightarrow Bloqueado

Cuando se completa una operación de entrada/salida o algún otro evento que espere un proceso. El proceso realiza la transición del estado bloqueado al estado listo.

Despertar(x_proceso):Bloqueado \longrightarrow Listo

Como se observará la única transición de estado iniciada por el proceso de usuario es el bloqueo; las otras tres transiciones son iniciadas por entidades externas al proceso.

Bloque de control del proceso (PCB)

La forma en que se manifiesta un proceso en un sistema operativo es mediante el PCB (Bloque de Control de Proceso). EL PCB es una estructura de datos que contiene información importante acerca de un proceso incluyendo:

- El estado actual del proceso.
 - Un identificador único del proceso.
 - Un apuntador hacia el padre del proceso (hacia el proceso que lo creó).
 - Apuntadores a los hijos del proceso en el caso de tener procesos creados por él.
-
- La prioridad del proceso.
 - Apuntadores hacia las zonas de memoria del proceso.
 - Apuntadores a los recursos asignados del proceso.
 - Un área para salvaguarda de los registros.
 - El procesador en que se esta ejecutando el proceso (en el caso de múltiples procesadores).

El PCB es un almacén central de información que permite al sistema operativo localizar toda la información importante acerca de un proceso. PCB es la entidad que define un proceso al sistema operativo.

Cuando el sistema operativo conmuta el CPU entre varios procesos activos utiliza las áreas de salvaguarda de los PCB para guardar la información que necesita para reiniciar un proceso cuando obtenga el CPU.

Operaciones sobre procesos

Los sistemas que administran procesos deben ser capaces de realizar ciertas operaciones sobre procesos:

- Crear un proceso que implica:
 1. Darle un nombre al proceso.
 2. Dar de alta en la lista de procesos conocidos del sistema o de la tabla de procesos (insertar)
 3. Determinar la prioridad inicial del proceso.
 4. Crear el PCB de ese proceso.
 5. Asignar recursos al proceso.

Un proceso puede crear un nuevo proceso y al proceso creador se le va a llamar proceso padre y al proceso creado se le va a llamar proceso hijo, lo cual nos va a originar la estructura jerárquica de procesos.

- Destruir un proceso que implica:
 1. Eliminarlo de la lista de procesos, no solamente del sistema.
 2. Los recursos que estaba utilizando se devuelven al sistema.
 3. El PCB se borra.

La destrucción de un proceso es más difícil cuando este ha creado a otros procesos.

- Suspender un proceso

Un proceso suspendido no puede proseguir hasta que lo reanuda otro proceso. La suspensión es una operación importante y ha sido puesta en práctica en diferentes formas en diversos sistemas. La suspensión dura por lo regular breves momentos o breves períodos de tiempo. El sistema efectúa suspensiones para eliminar temporalmente ciertos procesos y así reducir la carga del sistema durante situaciones de carga máxima. Cuando hay suspensiones largas se deben liberar los recursos del proceso.

- Reanudar un proceso

Reanudar o activar un proceso implica reiniciarlo a partir del punto en el que se suspendió.

- Bloquear un proceso
- Despertar un proceso
- Despachar un proceso
- Permitir que un proceso se comunice con otros (comunicación entre procesos)
- Cambiarle la prioridad a un proceso.

Razones por las que se puede suspender un proceso:

- Si un sistema esta funcionando mal y es probable que falle se puede suspender los procesos activos para reanudarlos cuando se haya corregido el problema.
- Un usuario que desconfíe de los resultados parciales de un proceso puede suspenderlo en lugar de abortarlo hasta que verifique si el proceso funciona correctamente o no.
- Algunos procesos se pueden suspender como respuesta a las fluctuaciones a corto plazo de la carga del sistema y reanudarse cuando las cargas regresen a niveles normales.

Estados de Suspensión

Se ha añadido dos nuevos estados denominados: `suspendido_listo` y `suspendido_bloqueado`, por lo que ahora contaremos con estados activos que vimos con anterioridad, y estados de suspensión.

Tenemos que, una suspensión puede ser iniciada por el propio proceso o por otro. En un sistema con un solo procesador, el proceso en ejecución puede suspenderse a sí mismo, ningún otro proceso podría estar en ejecución para realizar la suspensión.

Solamente otro proceso puede suspender un proceso listo. La transición sería:

`Suspend(x_proceso):Listo` → `Suspendido_listo`

Un proceso puede hacer que otro proceso que se encuentre en estado `suspendido_listo` pase al estado listo. La transición sería:

Reanudar(x_proceso):Suspendido_Listo \longrightarrow Listo

Un proceso puede suspender a otro que este bloqueado. La transición sería:

Suspender(x_proceso):Bloqueado \longrightarrow Suspendido_Bloqueado

Un proceso puede reanudar otro proceso que este suspendido_bloqueado. La transición sería:

Reanudar(x_proceso):Suspendido_Bloqueado \longrightarrow Bloqueado

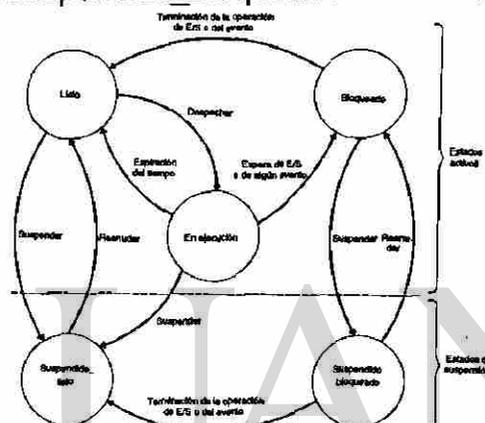
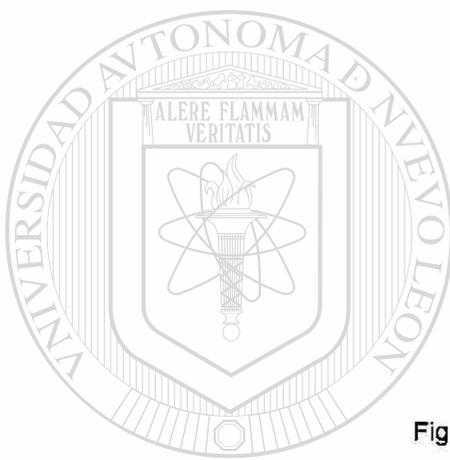


Fig. 2.2. Diagrama de cinco estados de los procesos

Transacciones

Suspender(x_proceso):Listo \longrightarrow Suspendido_Listo

Despachar(x_proceso):Listo \longrightarrow Ejecución

Suspender(x_proceso):Ejecución \longrightarrow Suspendido_Listo

Reanudar(x_proceso):Suspendido_Listo \longrightarrow Listo

Tiempo excedido(x_proceso):Ejecución \longrightarrow Listo

Suspender(x_proceso):Bloqueado \longrightarrow Suspendido_bloqueado

Completar(x_proceso):Bloqueado \longrightarrow Listo

Reanudar(x_proceso):Suspendido_Bloqueado \longrightarrow Bloqueado

Completar(x_proceso):Suspendido_Bloqueado \longrightarrow Suspendido_Listo

Esperar(x_proceso):Ejecución \longrightarrow Bloqueado

Procesamiento de interrupciones

- ♦ **Interrupción.**-Es un evento que altera la secuencia en que el procesador ejecuta las instrucciones. La interrupción es generada por el hardware del sistema.

Cuando se genera una interrupción:

1. El sistema operativo toma el control, es decir, el hardware pasa el control al sistema operativo.
2. El sistema operativo guarda el estado del proceso interrumpido, en muchos sistemas esta información se guarda en el PCB del proceso interrumpido.
3. El sistema operativo analiza la interrupción y transfiere el control a la rutina apropiada para atenderla. En algunos sistemas el hardware se encarga de esto automáticamente.
4. La rutina del manejador de interrupciones procesa la interrupción.
5. Se restablece el estado del proceso interrumpido o del siguiente proceso.
6. Se ejecuta el proceso interrumpido o el siguiente proceso.

Una interrupción puede ser generada específicamente por un proceso en ejecución en cuyo caso se le suele denominar trampa o trap y se dice que esta sincronizada con la operación del proceso o bien puede ser causada por algún evento que puede estar relacionado o no con el proceso en ejecución en cuyo caso se dice que es asíncrona con la operación en proceso. Interrumpir es una forma económica de llamar la atención del CPU.

Tipos de interrupciones

- ♦ **Interrupción SVC (Supervisor Call).**- Son iniciadas por un proceso en ejecución que ejecute una instrucción SVC. Una interrupción SVC es una petición generada por el usuario de un servicio en particular del sistema. Puede ser como ejecutar una operación de E/S, obtener más memoria, o comunicarse con el operador del sistema. Un usuario no puede entrar arbitrariamente al sistema operativo, debe solicitar permiso por medio de una SVC. El sistema operativo esta enterado de los usuarios que intentan

rebasar sus límites y puede rechazar ciertas peticiones si el usuario no cuenta con los privilegios necesarios.

- ◆ **Interrupciones de E/S.-** Son iniciadas por hardware de E/S. Este tipo de interrupciones indican al CPU el cambio de estado de algún canal o un dispositivo. Se producen cuando finaliza una operación de entrada/salida o cuando un dispositivo pasa al estado listo.
- ◆ **Interrupciones externas.-** Son causadas por diversos eventos, incluyendo la expiración de un cuanto (quantum del reloj) que interrumpe, cuando pulsamos la tecla de interrupción de la consola o bien cuando recibimos la señal de otro procesador, esto en el caso de un sistema de múltiples procesadores.
- ◆ **Interrupciones de reinicio.-** Cuando se oprime el botón de reinicio de la consola o si llegara desde otro procesador una señal de reinicio.
- ◆ **Interrupciones de verificación del programa.-** Son causadas por una amplia gama de problemas que pueden ocurrir cuando se ejecutan las instrucciones en lenguaje de máquina de un programa, dichos problemas incluyen:
 - ⇒ Una división entre cero.
 - ⇒ El exceso o defecto de los números que pueden ser manejados por las operaciones aritméticas.
 - ⇒ La presencia de datos con formato erróneo.
 - ⇒ El intento de ejecutar un código de operación inválido.
 - ⇒ El intento de hacer referencia a una localidad de memoria que este fuera de los límites de la memoria real.
 - ⇒ El intento por parte de un usuario no privilegiado de hacer uso de una instrucción privilegiada o el intento de hacer uso de un recurso protegido
- ◆ **Interrupciones por Verificación de Máquina.-** Son ocasionadas por mal funcionamiento del hardware.

Cambio de Contexto

En el esquema de las interrupciones de los procesadores a alta escala se maneja mediante el sistema operativo rutinas denominadas manejadores de interrupciones de primer nivel o FLIH (First Level Interrupt Handlers) para procesar las diferentes clases de interrupciones. Existe un manejador de interrupciones de primer nivel por cada tipo de interrupción.

Cuando ocurre una interrupción, el sistema operativo guarda el estado del proceso interrumpido y transfiere el control al manejador de interrupciones del primer nivel apropiado. Esto se logra mediante una técnica llamada **Cambio de Contexto**. Los manejadores de primer nivel deben distinguir entre interrupciones de la misma clase; el procesamiento de estas interrupciones es realizado entonces por alguno de los manejadores de interrupciones de segundo nivel.

Palabras de Estado o PSW (Program Status Word)

Las palabras de estado del programa controlan el orden de ejecución de las instrucciones y contienen información acerca del estado de un proceso. Hay tres tipos de PSW: nuevas, actuales y antiguas.

La dirección de la siguiente instrucción que se debe ejecutar se almacena en la PSW actual que indica también los tipos de interrupciones habilitadas e inhabilitadas en ese momento. El CPU permite que ocurran las interrupciones habilitadas; las interrupciones inhabilitadas permanecerán pendientes y solo en algunos casos se pasarán por alto.

El procesador nunca puede estar inhabilitado para una interrupción SVC o para una de reinicio o para ciertas interrupciones de verificación del programa.

En un sistema de un procesador solo hay una PSW actual, seis nuevas y seis antiguas, una por cada tipo de interrupción.

La PSW nueva para algún tipo de interrupción contiene la dirección permanente de la memoria principal en la que reside el manejador de interrupciones correspondiente.

Cuando ocurre una interrupción, si el procesador no está inhabilitado para este tipo de interrupción, el hardware automáticamente cambia la PSW como sigue:

1. Se guarda la PSW actual en la PSW antigua de ese tipo de interrupción.
2. Se guarda la PSW nueva de ese tipo de interrupción en la PSW actual.

Después de este intercambio de PSW, la PSW actual contendrá la dirección del manejador de interrupciones apropiada y este ejecutará y procesará la interrupción.

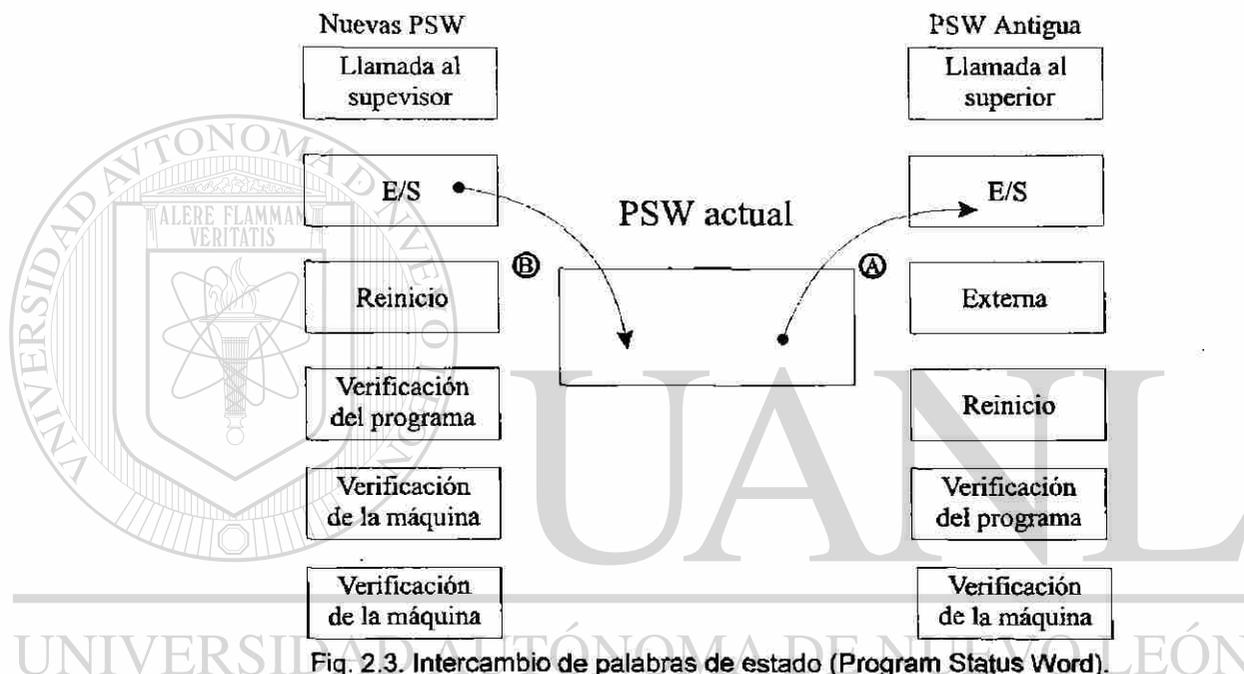


Fig. 2.3. Intercambio de palabras de estado (Program Status Word).

Núcleo del Sistema Operativo (Nucleus/Kernel/Core)

Todas las operaciones en las que participan procesos son controladas por la parte del sistema operativo llamada núcleo. El núcleo normalmente representa solo una pequeña parte de lo que por lo general se piensa que es todo el sistema operativo pero también es el código que más se utiliza. Reside por lo regular en la memoria principal mientras que otras partes del sistema operativo son cargadas solo cuando se necesitan. Una de las funciones más importantes incluidas en ese núcleo es el procesamiento de interrupciones. El núcleo inhabilita las interrupciones cuando atiende una interrupción y las habilita nuevamente una vez que ha completado el procesamiento de la interrupción.

Cuando se da un flujo continuo de interrupciones, es posible que el núcleo deje inhabilitadas las interrupciones por un periodo largo; esto puede ocasionar un elevado tiempo de respuesta a las interrupciones. Por esta razón los núcleos son diseñados para realizar el "mínimo" posible de procesamiento en cada interrupción y dejar que el resto lo realice el proceso apropiado del sistema, que puede operar mientras el núcleo se habilita para atender otras operaciones. Lo que se traduce en que las interrupciones permanecen habilitadas un porcentaje mucho mayor del tiempo y que el sistema responde mejor.

Resumen de las funciones del núcleo del Sistema Operativo

1. Manejo de Interrupciones.
2. Crear y Destruir procesos.
3. El cambio de los estados de los procesos.
4. El despacho de los procesos.
5. Suspender y reanudar un proceso.
6. Sincronización de los procesos.
7. La comunicación entre procesos.
8. El manejo de los bloques de control de procesos.
9. Apoyo a las actividades de entrada/salida.
10. Asignación y liberación de memoria.
11. El apoyo para el sistema de archivos.
12. Apoyo para el mecanismo de llamada y retorno de un procedimiento.
13. Apoyo para ciertas funciones de contabilidad del sistema

Habilitación e Inhabilitación de Interrupciones

Al núcleo del sistema operativo solo puede llegarse mediante una interrupción. El núcleo inhabilita las interrupciones mientras responde a la interrupción que esta procesando. Una vez que se determina la causa de la interrupción, el núcleo pasa el tratamiento de la interrupción a un proceso específico del sistema que se ha diseñado para manejar ese tipo de interrupciones. En algunos sistemas el procesamiento de cada interrupción es realizado por un sistema operativo de una sola pieza.

PLANIFICACION DE TRABAJOS Y DEL PROCESADOR

En la planificación del procesador se estudian los problemas de cuándo asignar procesadores y a cuáles procesos asignarlos, también se considera la admisión de nuevas tareas en el sistema, la suspensión y reactivación de procesos para ajustar la carga del sistema.

Niveles de planificación

- ◆ **Planificación de Alto Nivel.-** Planificación de trabajo. Determina cuáles trabajos podrán competir activamente por los recursos del sistema o bien cuales trabajos deben admitirse por lo que también la llamamos planificación de admisión.
- ◆ **Planificación de Nivel Intermedio.-** Determina que procesos pueden competir por el CPU. Responde a las fluctuaciones temporales en la carga del sistema mediante la suspensión temporal y la activación (reanudación) de procesos para lograr una operación mas fluida del sistema y ayuda a alcanzar ciertas metas globales del rendimiento del sistema. Actúa como amortiguador entre la admisión de los trabajos y la asignación del CPU a esos trabajos.
- ◆ **Planificación de Bajo Nivel.-** Determina a cuál proceso listo se le asignará el CPU cuando éste se encuentre disponible y se encargará de asignar el CPU a ése proceso. Se lleva a cabo mediante el despachador y este debe residir en la memoria principal.

Objetivos de la planificación

En el diseño de una disciplina de planificación deben ser considerados varios objetivos, mismos que suelen caer en conflicto haciendo esto un tanto más complejo. Sin embargo a continuación se listan algunos objetivos:

- ◆ **Ser justa.-**Una disciplina de planificación es justa si todos los procesos se tratan de la misma forma y ningún proceso se queda en aplazamiento indefinido.

- ◆ **Elevar al máximo la producción y el rendimiento.**- Una disciplina de planificación debe de tratar de atender el mayor número posible de procesos por unidad de tiempo.
- ◆ **Aumentar al máximo el número de usuarios interactivos que reciben respuesta en tiempos aceptables (segundos).**
- ◆ **Ser predecible.**- Una tarea debe ejecutarse aproximadamente en el mismo tiempo y casi al mismo costo sea cual sea la carga del sistema.
- ◆ **Reducir al mínimo el gasto extra.**- El gasto extra se considera por lo común un desperdicio de recursos, pero la inversión de cierta porción de los recursos del sistema como gasto extra puede mejorar en gran medida el rendimiento total del sistema.
 - ◆ **Equilibrar el aprovechamiento de los recursos.**- Los mecanismos de planificación deben mantener ocupados los recursos del sistema. Deben favorecerse los procesos que requieren los recursos poco utilizados.
 - ◆ **Lograr un equilibrio entre la respuesta y el aprovechamiento.**- La mejor manera de garantizar tiempos de respuesta adecuados es tener suficientes recursos disponibles en el momento en que son necesarios. El precio que debe pagarse por esta estrategia es que el aprovechamiento global de los recursos será pobre. En los sistemas de tiempo real, las respuestas rápidas son esenciales y el aprovechamiento de los recursos es menos importante. En otros tipos de sistemas, la economía exige un aprovechamiento efectivo de recursos.
- ◆ **Evitar el Aplazamiento Indefinido.**- La mejor manera de evitarlo es el empleo del envejecimiento, es decir, mientras un proceso espera un recurso, su prioridad debe crecer. En algún momento, la prioridad será tan alta que el recurso se asignará al proceso.
- ◆ **Imponer prioridades.**- En los ambientes en que se asignan prioridades a los procesos, los mecanismos de planificación deben favorecer a los procesos de alta prioridad.
- ◆ **Dar preferencia a los procesos que ocupan recursos decisivos.**- Aunque un proceso tenga baja prioridad, podría estar ocupando un recurso

decisivo, y el recurso puede ser requerido por un proceso de alta prioridad. Si el recurso no es apropiado, el mecanismo de planificación debe dar al proceso un trato mejor del que se le daría comúnmente, de tal modo que libere el recurso con más rapidez.

- ◆ **Dar un mejor trato a los procesos que muestran un comportamiento deseable.**- Ejemplo: bajas tasas de paginación.
- ◆ **Degradarse paulatinamente con las cargas pesadas.**- Un mecanismo de planificación no debe desplomarse bajo el peso de una carga fuerte en el sistema. Debe evitar la carga excesiva impidiendo la creación de procesos nuevos cuando la carga es pesada, o bien debe dar servicio a la carga mayor con una reducción moderada del nivel de atención a todos los procesos.

Criterios de planificación

- ◆ **La limitación de un proceso por E/S.**- Cuando un proceso obtiene el CPU ¿lo usará en forma breve antes de generar una petición de E/S?
- ◆ **La limitación de un proceso por el CPU.**- Cuando un proceso obtiene el CPU, ¿tiende a usarlo hasta que expira su cuanto de tiempo?
- ◆ **Si un proceso es por lotes o es interactivo.**- Los usuarios interactivos suelen hacer peticiones “triviales” que deben atenderse de inmediato para garantizar tiempos de respuesta adecuados. Los usuarios por lotes, por lo general pueden tolerar retrasos razonables.
- ◆ **Qué tan urgente es la respuesta.**- Un proceso por lotes que tarda toda la noche no requerirá una respuesta inmediata. Un sistema de control de procesos en un tiempo real que supervise a una refinería de petróleo necesita una respuesta rápida, quizá para evitar una explosión.
- ◆ **Las prioridades de los procesos.**- Los procesos de alta prioridad deben recibir mejor tratamiento que los de baja prioridad.
- ◆ **La frecuencia con la que un proceso está generando fallas de página.**- Teóricamente, un proceso que genera pocas fallas de página tiene acumulados sus conjuntos de trabajo en el almacenamiento principal. Pero

los procesos que experimentan muchas fallas de página no han establecido aún sus conjuntos de trabajo, y lo convencional es favorecer a los procesos que ya los han establecido. Otro punto de vista es que los procesos con altas tasas de fallas de página deben tener preferencia, ya que usan poco el CPU antes de generar una petición de E/S.

- ◆ **Las frecuencias con las que los recursos de un proceso son apropiados por otro de mayor prioridad.**- Los procesos cuyos recursos son apropiados muy a menudo deben recibir un tratamiento menos favorable. La cuestión es que cada vez que el sistema operativo invierte trabajo extra en echar a andar este proceso, el breve lapso de ejecución que se logra antes de la apropiación no justifica el gasto extra necesario para echar a andar el proceso en primer lugar.

- ◆ **Cuanto tiempo real de ejecución ha recibido el proceso.**- Algunos diseñadores opinan que deben favorecerse los procesos con poco tiempo de ejecución. Otro punto de vista es que un proceso que ha recibido mucho tiempo de ejecución debe estar por terminar, debe favorecerse para ayudarlo a que termine y abandone el sistema tan pronto como sea posible.

- ◆ **Cuanto tiempo más necesitará el proceso para terminar.**- Los tiempos de espera promedio pueden reducirse al mínimo ejecutando primero aquellos procesos que requieran los menores tiempos de ejecución para terminar.

Lamentablemente, casi nunca se sabe con exactitud cuanto tiempo durará un proceso.

Planificación no apropiativa y apropiativa

Una disciplina de planificación es no apropiativa si una vez que le ha sido asignado el CPU a un proceso, ya no se le puede arrebatarse. En los sistemas no apropiativos, los trabajos largos retrasan a los cortos, pero el tratamiento para todos los procesos es más justo. Los tiempos de respuesta son más predecibles porque los trabajos nuevos de alta prioridad no pueden desplazar a los trabajos en espera.

Una disciplina de planificación es apropiativa si al proceso se le puede arrebatarse el CPU. La planificación apropiativa es importante en aquellos sistemas de planificación en los cuales los procesos de alta prioridad requieren atención rápida. En los sistemas interactivos de tiempo compartido, la planificación apropiativa es importante para garantizar tiempos de respuesta aceptables.

La apropiación tiene un precio. El cambio de contexto implica un gasto extra. Para que la técnica de apropiación sea efectiva deben mantenerse varios procesos en el almacenamiento principal de manera que el siguiente proceso se encuentre listo cuando quede disponible el CPU. Conservar en el almacenamiento principal programas que no estén en ejecución también implica gasto extra.

Al hacer el diseño de un mecanismo de planificación apropiativa hay que tomar en cuenta la arbitrariedad de casi todos los sistemas de prioridades. Sin embargo es necesario analizar y evaluar cada mecanismo de planificación antes de ser implantado. La sencillez puede ser atractiva, pero si el mecanismo no se puede hacer sencillo, debe tratarse al menos de hacerlo efectivo y significativo.

Prioridades

Las prioridades pueden ser asignadas en forma automática por el sistema, o bien se pueden asignar externamente. Pueden ganarse o comprarse. Pueden ser estáticas o dinámicas. Pueden asignarse en forma racional, o de manera arbitraria en situaciones en las que un mecanismo del sistema necesita distinguir entre procesos pero no le importa cual de ellos es en verdad más importante.

- ◆ **Prioridades estáticas.**- No cambian, son fáciles de llevar a la práctica e implican un gasto extra relativamente bajo. No responden a cambios en el ambiente que podrían hacer necesario un ajuste de prioridades. Es decir se mantienen constantes mientras dura el proceso.
- ◆ **Prioridades dinámicas.**- Responden a los cambios. La prioridad inicial asignada a un proceso tiene una duración corta, después de lo cual se ajusta a un valor más apropiado. Este esquema es más complejo e implica

más gasto extra que los esquemas estáticos, pero este queda justificado por el aumento de sensibilidad del sistema. Cambian en respuesta a los cambios de las condiciones del sistema.

- ◆ **Prioridades compradas.-** Un sistema operativo debe proporcionar un servicio competente y razonable a una gran comunidad de usuarios, pero también debe manejar las situaciones en las cuales un miembro de la comunidad necesite un trato especial. Un usuario con un trabajo urgente puede estar dispuesto a pagar extra, esto es, comprar prioridad, por un nivel más alto de servicio. Este pago extra es obligatorio debido a que puede ser necesario arrebatar recursos con otros usuarios que también pagan. Si no hubiera un pago extra, entonces todos los usuarios pedirían un nivel más alto de servicio.

TIPOS DE PLANIFICACION

Planificación a Plazo Fijo.- Se programan ciertos trabajos para terminarse en un tiempo específico. Los trabajos pueden tener un gran valor si son entregados a tiempo y carecer de él si son entregados fuera del plazo, por lo que algún usuario puede estar dispuesto a pagar extra para asegurar que sus trabajos sean entregados a tiempo. Este tipo de planificación es compleja por varias razones que debemos considerar:

- El usuario debe informar por adelantado las necesidades precisas de recursos de la tarea. Esta información no suele estar disponible.
- El sistema debe ejecutar la tarea en un plazo determinado sin degradar el servicio a otros usuarios.
- El sistema debe planificar cuidadosamente sus necesidades de recursos dentro del plazo. Esto puede ser difícil por la llegada de nuevos procesos que impongan demandas impredecibles al sistema.
- Si hay muchas tareas a plazo fijo activas al mismo tiempo podría ser necesario la utilización de métodos de optimización para cumplir con los plazos.

- La administración intensiva de recursos requerida por ésta planificación puede ocasionar un gasto extra sustancial. Aunque los usuarios estén dispuestos a pagar una cuota alta por los servicios recibidos, el consumo neto de los recursos del sistema puede ser tan alto que el resto de la comunidad puede sufrir degradación del servicio.

Planificación Primeras Entradas Primeras Salidas (PEPS o First Input First Output FIFO).- Es la disciplina más simple. Los procesos se despachan de acuerdo a su tiempo de llegada a la cola de procesos listos. Es una disciplina no apropiativa. Es justa en el sentido formal pero es injusta con los procesos cortos que tienen que esperar a que los trabajos largos se ejecuten o bien que los trabajos importantes tienen que esperar a que se terminen los de menor importancia. Es la más predecible de las planificaciones. Sin embargo no es útil en la planificación para usuarios interactivos porque no puede garantizar buenos tiempos de respuesta.

Planificación por turno (Round Robin [RR]).- Es una disciplina apropiativa. Los procesos se despachan en forma PEPS, pero se les asigna una cantidad limitada de tiempo de CPU conocida como cuanto o quantum. Si un proceso no termina antes de que expire su tiempo de CPU, se le quitará el CPU y éste se le asignará al siguiente proceso en espera, el proceso desposeído se colocará al final de la cola de procesos listos. Es efectiva en ambientes de tiempo compartido en los que se necesita garantizar tiempos de respuesta razonables para usuarios interactivos. El gasto extra debido a la apropiación es bajo gracias a eficientes mecanismos de cambio de contexto y a la asignación de suficiente memoria para que los procesos residan en la memoria al mismo tiempo.

☆ **Quantum.**- La determinación del tamaño de quantum es vital para lograr una buena utilización del sistema y tiempos de respuesta razonable. Un tamaño de quantum muy grande hará que cualquier disciplina apropiativa se aproxime a su contraparte no apropiativa. Un quantum muy pequeño puede desperdiciar tiempo de CPU al obligar a un excesivo cambio de contexto

entre procesos. Por lo que se debe de elegir lo bastante grande para que la mayoría de las solicitudes triviales terminen en un quantum. Ejemplo: en un sistema limitado de E/S, el quantum es lo bastante grande para que la mayor parte de los procesos puedan realizar una petición de E/S antes de que expire su quantum.

Planificación por Prioridad del Trabajo más Corto Primero (Shortest-job-first SJF).- Es una disciplina no apropiativa utilizada sobre todo para trabajos por lotes. Según esta disciplina se ejecuta primero el trabajo (o proceso) en espera que tiene el menor tiempo estimado de ejecución hasta terminar. Reduce al mínimo el tiempo promedio de espera pero los trabajos largos pueden verse sometidos a largas esperas. El problema obvio con SJF es que exige conocer con exactitud el tiempo que tardará en ejecutarse un trabajo o proceso, y esa información no suele estar disponible; lo mejor que se puede hacer es basarse en los tiempos de ejecución estimados por el usuario.

Planificación del Tiempo Restante mas Corto (Shortest-remaining-time-scheduling SRT).- Es la contraparte apropiativa de SJF. En SRT, el proceso con el menor tiempo estimado de ejecución para terminar es el primero en ejecutarse, incluyendo los procesos nuevos. Un proceso en ejecución puede ser despojado por un proceso nuevo con un tiempo estimado de ejecución más pequeño; implica un gasto extra mayor que SJF, pero proporciona un mejor servicio a los trabajos nuevos cortos. Reduce más los tiempos promedio de espera de todos los trabajos pero los trabajos largos pueden sufrir retrasos mucho mayores que en SJF.

Planificación por Prioridad de la Tasa de Respuesta mas Alta (Highest-response-ratio-next HRN).- Corrige algunos defectos de SJF, particularmente la excesiva predisposición contra los trabajos largos y el favoritismo de trabajos cortos nuevos. Es una disciplina no apropiativa en la cual la prioridad de cada trabajo no sólo es función del tiempo de servicio, sino también del tiempo que ha esperado el trabajo para ser atendido. Cuando un trabajo obtiene el

procesador, se ejecuta hasta terminar. Las prioridades dinámicas se calculan con la siguiente fórmula:

$$\text{Prioridad} = \frac{\text{Tiempo de espera} + \text{Tiempo de servicio}}{\text{Tiempo de servicio}}$$

Como el tiempo de servicio aparece en el denominador, los trabajos cortos tendrán preferencia. Como el tiempo de espera aparece en el numerador, los trabajos largos que han esperado también tendrán un trato favorable. El tiempo de respuesta del sistema para el trabajo si este se inicia de inmediato se representa por esta suma:

$$\text{Tiempo de Respuesta} = \text{Tiempo de espera} + \text{Tiempo de servicio}$$

Procesos Concurrentes Asíncronos

- ◆ **Proceso Concurrente.-** Los procesos son concurrentes si existen simultáneamente. Los procesos concurrentes pueden funcionar en forma totalmente independiente unos de otros o pueden ser asíncronos, lo cual significa que en ocasiones requieren cierta sincronización y cooperación.

Procesamiento en Paralelo

A medida que el hardware siga disminuyendo en tamaño y costo irán apareciendo tendencias definidas hacia el multiprocesamiento, el procesamiento distribuido y el paralelismo a gran escala. Si algunas operaciones pueden ejecutarse de manera lógica en paralelo, los computadores también podrán realizarlas físicamente en paralelo, aun sí el nivel de paralelismo es de miles o tal vez de millones de actividades concurrentes. Ello puede significar mejoras en el rendimiento muy superiores a las que son posibles en los computadores secuenciales. El procesamiento paralelo resulta interesante y complejo por diversas razones. Resulta más fácil para la gente concentrar su atención en una sola actividad a la vez que pensar en paralelo.

Ejemplo.- Intente leer dos libros al mismo tiempo, leyendo una línea del primero y una del segundo y así sucesivamente.

Determinar qué actividades pueden realizarse en paralelo es difícil y lleva mucho tiempo. Los programas en paralelo son mucho más difíciles de depurar que los secuenciales. **Ejemplo.-** Después de corregir un error, puede resultar imposible reconstruir la secuencia de eventos que provocaron su aparición original, por lo que sería impropio asegurar que ya se ha corregido.

Una Estructura de Control para Indicar Paralelismo: Parbegin/Parend

- Existen muchas construcciones de lenguajes de programación para indicar paralelismo. Normalmente se trata de parejas de enunciados como a continuación se describen:
- Un enunciado para indicar que la ejecución secuencial se dividirá en varias secuencias de ejecución paralelas (hilos de control).
- Un enunciado para indicar la fusión de ciertas secuencias de ejecución paralelas y la reanudación de la ejecución secuencial.

Estos enunciados aparecen en pareja y se conocen como parbegin/parend (para la ejecución en paralelo) o cobegin/coend (para la ejecución concurrente) para indicar el principio y fin.

```
parbegin
enunciado 1;
enunciado 2;
```

```
enunciado n
```

```
parend
```

Ejemplo.- Supongamos que un programa se está ejecutando una sola secuencia de instrucciones encuentra la construcción parbegin. Eso hace que el hilo de control único se divida en n hilos separados; uno para cada enunciado de la construcción parbegin/parend. Puede tratarse de enunciados simples, llamadas a procedimientos, bloques de enunciados secuenciales delimitados por un par begin/end, o combinaciones de ellos. Cada uno de los hilos de control terminará en algún momento y llegará a parend. Cuando terminen todos los hilos de control, reanudará un hilo de control único y el sistema ejecutará el enunciado que sigue al parend.

Ejemplo.- Considérese el cálculo de una raíz de una ecuación cuadrática:

$$x = (-b + (b^2 - 4ac)^{.5}) / (2a)$$

Procesamiento Secuencial	Procesamiento Paralelo
1 b^2	1 <code>parbegin</code>
2 $4a$	<code>temp1 = -b;</code>
3 $(4a)^c$	<code>temp2 = b^2;</code>
4 $(b^2) - (4ac)$	<code>temp3 = 4a;</code>
5 $(b^2 - 4ac)^{.5}$	<code>temp4 = 2a</code>
6 $-b$	<code>parend;</code>
7 $(-b) + ((b^2 - 4ac)^{.5})$	2 <code>temp5 = temp3 * c;</code>
8 $2a$	3 <code>temp5 = temp2 - temp5;</code>
9 $(-b + (b^2 - 4ac)^{.5}) / (2a)$	4 <code>temp5 = temp5^{.5};</code>
	5 <code>temp5 = temp1 + temp5;</code>
	6 <code>x = temp5 / temp4</code>

Las cuatro operaciones contenidas dentro del `parbegin/parend` se evalúan en paralelo y las cinco restantes en secuencia, reduciendo considerablemente el tiempo.

♦ **Exclusión Mutua.**- Se llama exclusión mutua a la situación en la cual los procesos cooperan de tal forma que, mientras un proceso obtiene acceso a datos compartidos modificables, los demás procesos no pueden hacer lo mismo.

♦ **Secciones Críticas.**- Cuando un proceso obtiene acceso a datos compartidos modificables se dice que se encuentra en una sección crítica o región crítica. Las secciones críticas deben ejecutarse lo más rápido posible; un proceso no debe bloquearse en su sección crítica y las secciones críticas deben ser codificadas cuidadosamente para evitar la posibilidad de ciclos infinitos por ejemplo.

♦ **Algoritmo de Dekker.**- Es una realización en software de las primitivas de exclusión mutua y tiene las siguientes propiedades:

- No requiere ninguna instrucción especial de hardware.
- Un proceso que se encuentra fuera de su sección crítica no puede evitar que otro proceso entre en su propia sección crítica.

Un proceso que desea entrar en su sección crítica puede hacerlo sin que haya posibilidad de un aplazamiento indefinido.

```

program algoritmo_dekker;
var proceso_favorecido: (primero, segundo);
    p1deseaentrar, p2deseaentrar: boolean;
procedure proceso_uno;
begin
    while true do
        begin
            p1deseaentrar := true;
            while p2deseaentrar do
                if proceso_favorecido = segundo then
                    begin
                        p1deseaentrar:= false;
                        while proceso_favorecido = segundo do;
                        p1deseaentrar := true
                    end;
                sección_crítica_uno;
                proceso_favorecido:= segundo;
                p1deseaentrar:= false;
                otras_tareas_uno
            end
        end;
procedure proceso_dos;
begin
    while true do
        begin
            p2deseaentrar := true;
            while p1deseaentrar do
                if proceso_favorecido = primero then
                    begin
                        p2deseaentrar:= false;
                        while proceso_favorecido = primero do;
                        p2deseaentrar := true
                    end;
                sección_crítica_dos;
                proceso_favorecido:= primero;
                p2deseaentrar:= false;
                otras_tareas_dos
            end
        end;
begin
    p1deseaentrar := false;
    p2deseaentrar := false;
    proceso_favorecido := primero;
    parbegin
        proceso_uno;
        proceso_dos
    parend
end.

```

El algoritmo Dekker se aplica para la exclusión mutua de n pasos; estas suelen ser soluciones muy complejas. Sin embargo durante muchos años representó el último adelanto en cuestión de algoritmos de espera activa para manejar la exclusión mutua.

En 1981, G.L. Peterson publicó un algoritmo mucho más sencillo para manejar la exclusión mutua de dos procesos con espera activa.

```

program algoritmo_peterson;
var proceso_favorecido: (primero, segundo);
    p1deseaentrar, p2deseaentrar: boolean;
procedure proceso_uno;
begin
    while true do
        begin
            p1deseaentrar := true;
            proceso_favorecido := segundo;
            while p2deseaentrar
                and proceso_favorecido = segundo do;
            sección_crítica_uno;
            p1deseaentrar:= false;
            otras_tareas_uno
        end
    end;
procedure proceso_dos;
begin
    while true do
        begin
            p2deseaentrar := true;
            proceso_favorecido := primero;
            while p1deseaentrar
                and proceso_favorecido = primero do;
            sección_crítica_dos:= false;
            otras_tareas_dos
        end
    end;
begin
    p1deseaentrar := false;
    p2deseaentrar := false;
    proceso_favorecido := primero;
parbegin
    proceso_uno;
    proceso_dos
parend
end.

```

Semáforos

Dijkstra extrajo los conceptos fundamentales de la exclusión mutua en su concepto de semáforos. Un semáforo es una variable protegida cuyo valor solo puede ser leído y alterado mediante las operaciones P y V y una operación de asignación de valores iniciales. Los semáforos binarios pueden tener solamente los valores 0 o 1. Los semáforos contadores o semáforos generales pueden tener tan solo valores enteros no negativos.

La operación P sobre el semáforo S, escrita P(S), opera como sigue:

```

Sí S > 0
entonces S := S - 1
sí no (esperar S)

```

La operación V sobre el semáforo S , escrita $V(S)$, opera como sigue:

Si (uno o más procesos esperan S)
entonces (dejar que prosiga uno de esos procesos)
si no $S := S + 1$

Se adoptará una disciplina de colas de primeras entradas-primeras salidas para los procesos que esperan terminar una operación $P(S)$.

P y V son indivisibles. La exclusión mutua sobre el semáforo S se implanta dentro de $P(S)$ y $V(S)$. Si varios procesos desean ejecutar una operación $P(S)$ de manera simultánea, solamente se permitirá la ejecución de uno de ellos. Los otros quedaran en espera, pero la manera como se realizan P y V garantiza que los procesos no se aplazarán en forma indefinida.

Los semáforos pueden llevarse a la práctica en software o hardware. Se suelen incluir en el núcleo del sistema operativo donde se controla la comunicación de los estados de los procesos.

Los semáforos pueden servir para llevar a la práctica un mecanismo de sincronización bloquear/despertar: un proceso se bloquea a sí mismo mediante $P(S)$, con $S=0$ inicialmente, para esperar que ocurra un evento; otro proceso detecta el evento y despierta al proceso bloqueado mediante $V(S)$.

En una relación productor-consumidor, un proceso productor, genera información que utiliza un segundo proceso consumidor (comunicación entre procesos). Si estos procesos se comunican por medio de un buffer compartido, el productor no debe producir cuando el buffer esta lleno y el consumidor no debe consumir cuando el buffer esta vacío (sincronización de procesos).

Los semáforos contadores son particularmente útiles cuando un recurso debe asignarse a partir de un banco de recursos idénticos. Cada operación P indica que se ha asignado un recurso, en tanto que una operación V indica que se ha devuelto un recurso al banco.

Las operaciones de los semáforos pueden llevarse a la práctica con espera activa, pero esto puede ocasionar desperdicio (pueden incluirse en el núcleo para evitar la espera activa).

Programación Concurrente

Los métodos anteriores poseen en su estructura varios defectos. Son lo bastante primitivos que es difícil expresar las soluciones a los problemas de concurrencia más complejos aumentando el ya de por sí difícil problema de probar la corrección de los programas. De hecho el mal uso de estas primitivas podría corromper la operación de un sistema concurrente.

En el método de los semáforos, si omitimos P, no obtenemos la exclusión mutua; si omitimos V, las tareas que están esperando por causa de operaciones P podrían quedar en bloqueo permanente. Una vez que comienza una operación P, el usuario no puede arrepentirse y seguir otro curso de acción mientras el semáforo esta en uso. Una tarea puede esperar solo un semáforo a la vez, lo cual pudiera originar un bloqueo mutuo en situaciones de asignación de recursos.

Ha surgido gran interés en los lenguajes de programación concurrente porque permiten expresar de una forma natural las soluciones a ciertos problemas de carácter inherentemente paralelo, y también gracias al verdadero paralelismo de hardware que es posible lograr con los multiprocesadores y sistemas distribuidos. Los programas concurrentes son más difíciles de escribir, depurar, modificar y probar que son correctos, sin embargo, tienen muchas aplicaciones posibles. Los sistemas operativos en sí son ejemplos importantes de sistemas concurrentes, como también lo son los sistemas de control de tráfico aéreo, los sistemas de control de tiempo real (los que controlan las refinerías, plantas de productos químicos, procesadoras de alimentos, etc.). Los robots son sistemas altamente concurrentes, cuando un robot camina debe ser capaz de ver, oír, tocar, saborear y oler al mismo tiempo. De hecho, el proceso de caminar, es preciso coordinar con cuidado la operación de cada motor y cada parte de los miembros para lograr incluso los movimientos más sencillos.

ADA es el primer lenguaje concurrente de uso generalizado, pero pasará algún tiempo antes de que haya un número significativo de sistemas realizados por completo en ADA.

Regiones Críticas y Regiones Críticas Condicionales

La noción de región crítica se desarrollo para expresar de forma sencilla la exclusión mutua. Por ejemplo para indicar que alguna acción se realizará con acceso exclusivo a ciertos datos_ compartidos se escribe la siguiente instrucción en Pascal:

región datos_compartidos do acción ↔ Región crítica

La noción de región crítica condicional permite especificar la sincronización, además de la exclusión mutua cuyo ejemplo es:

región datos_compartidos do begin await condición; acción end ↔ Región crítica condicional

Es posible realizar acción con acceso exclusivo a datos_compartidos cuando se cumple condición.

MONITORES

◆ **Monitor.-** Es una construcción para concurrencia que contienen tanto los datos como los procedimientos necesarios para asignar un recurso compartido específico y reutilizable en serie o en grupos de estos recursos.

Para ejecutar una función de asignación de recursos, un proceso debe llamar a una entrada del monitor específica. Muchos procesos pueden desear entrar al monitor en diversos momentos, pero la exclusión mutua se mantiene de manera estricta en los límites del monitor ya que solo se permite la entrada de un proceso a la vez. Los procesos que desean entrar en el monitor cuando ya esta en uso deben esperar; el monitor controla automáticamente esta espera, de esta forma la exclusión mutua esta garantizada.

Los datos que se encuentran dentro del monitor pueden ser globales con respecto de todos los procedimientos del monitor o locales con respecto de un procedimiento específico, Todos esos datos son solo accesibles dentro del monitor; un proceso fuera del monitor no puede tener acceso a esos datos, a esto le llamamos ocultación de información, que es una técnica de estructuración de sistemas que facilita mucho el desarrollo de sistemas de software confiables.

Si un proceso que esta llamando a la entrada del monitor encuentra que el recurso ya ha sido asignado, el procedimiento monitor llama a la función esperar. El proceso podría permanecer dentro del monitor pero ello violaría la exclusión mutua si otro proceso entrara en el monitor. Por lo tanto el proceso que llamó a *esperar* debe aguardar fuera del monitor hasta que se libere el recurso. En algún momento, el proceso que tiene el recurso llamará a una entrada al monitor para devolver el recurso al sistema. Esta entrada podría limitarse a aceptar el recurso devuelto y esperar a que llegue otro proceso solicitante. Pero puede haber procesos esperando el recurso, así que la entrada del monitor invoca a la función señalar para permitir que uno de los procesos en espera adquiera el recurso y salga del monitor.

Si un proceso señala la liberación del recurso y ningún proceso esta esperando, la señal no tendrá efecto, pero el monitor habrá recuperado un recurso. Es claro que un proceso que esta en espera de un recurso deberá hacerlo afuera para permitir que otro proceso entre al monitor a devolver el recurso. Se da mayor prioridad a los procesos en espera para asegurar que un proceso que esta en espera de un recurso termine por obtenerlo, de otra forma un proceso nuevo podría obtener ese recurso y ocasionar un problema de aplazamiento indefinido.

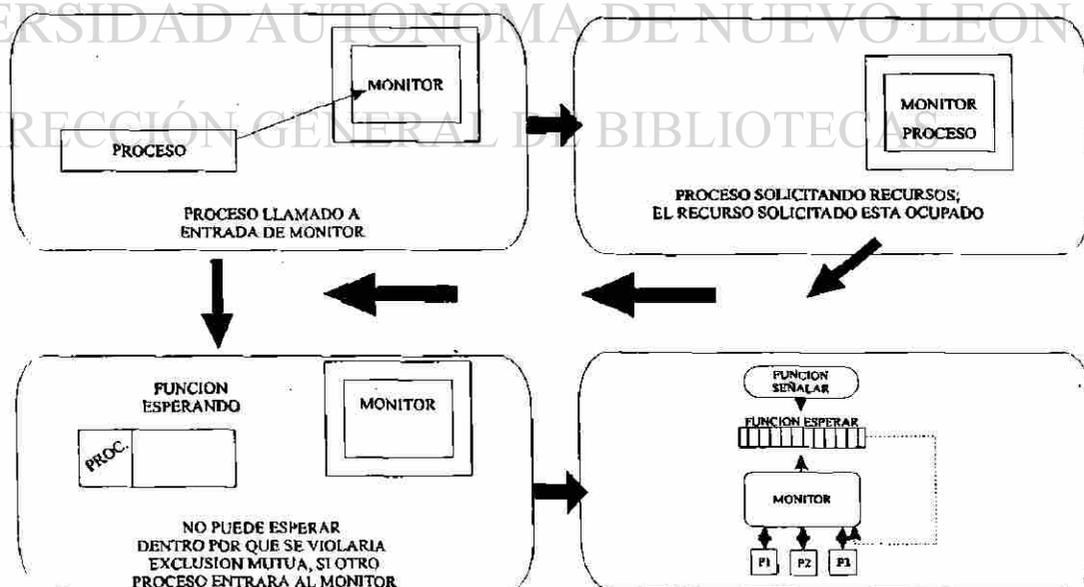


Fig. 2.4. Pasos que sigue un proceso al solicitar recursos al monitor.

- ◆ **Variables Condicionales.-** Las variables condicionales son diferentes de las variables convencionales con las que estamos familiarizadas. Se asocia una variable condicional diferente a cada una de las razones por las que un proceso puede verse obligado a esperar. Las operaciones de esperar y señalar se modifican para incluir el nombre de la variable condicional que esta esperando o señalando.

ESPERAR (nombre de variable condicional)

SEÑALAR (nombre de variable condicional)

Cuando se define una variable condicional se crea una cola. Un proceso que llama a la función esperar se agrega a la cola; un proceso que llama a la función señalar hace que un proceso en espera sea sacado de la cola y entre en el monitor.

A continuación se muestra un monitor sencillo para el manejo de la asignación de recursos. El atractivo de la función asignador_de_recursos radica en que funciona exactamente como un semáforo binario; la función obtener_recurso trabaja como la operación P y la función devolver_recurso trabaja como la operación V.

```
monitor asignador_de_recursos;
var recurso_en_uso:boolean;
recurso_libre:condition;
```

```
procedure obtener_recurso;
begin
if recurso_en_uso then
esperar(recurso_libre);
recurso_en_uso := true
end;
```

```
procedure devolver_recurso;
begin
recurso_en_uso := false;
señalar (recurso_libre);
end;
begin
recurso_en_uso := false
end;
```

Función P



Función V



- ◆ **Buffer circular (Ring Buffer, Bounded Buffer o Buffer Acotado).-** Un ejemplo de Monitor. Se utiliza en situaciones en las que un proceso productor pasa datos a un proceso consumidor. Por lo que debemos considerar de vital importancia la sincronización entre los procesos productores y consumidores.

Los sistemas operativos asignan a menudo una cantidad fija razonable de memoria para las comunicaciones mediante buffer entre los procesos productores y consumidores. Esto se puede simular mediante una tabla de tamaño indicado. El productor deposita los datos en elementos sucesivos de la tabla y el consumidor los saca en el orden en que se depositaron. El productor puede haber producido varios elementos que aun no ha utilizado el consumidor. En algún momento el productor llenará el último elemento de la tabla, cuando produzca mas datos deberá dar la vuelta y comenzar a depositar los datos en el primer elemento de la tabla, dando por asentado que el consumidor haya sacado los datos previamente colocados ahí por el productor, la tabla se cierra efectivamente en circulo, de ahí el termino de buffer circular.

Debido al tamaño fijo del buffer circular el productor puede encontrar a veces ocupados todos los elementos de la tabla, en este caso, el productor debe esperar hasta que el consumidor vacíe algunos elementos de la tabla, del mismo modo en que habrá ocasiones en que el consumidor deberá esperar hasta que el productor deposite algunos datos en algunos elementos de la tabla.

Un mecanismo de buffer circular es adecuado para implantar un mecanismo de spool en los sistemas operativos. Cuando un proceso genera líneas que se deben imprimir en un dispositivo de salida relativamente lento, como una impresora. Como el proceso puede producir las líneas mucho más rápidamente de lo que la impresora puede imprimirlas y como también se desea que el proceso se ejecute lo mas rápidamente posible, las líneas de salida del proceso se envían a un mecanismo de buffer circular. El buffer circular puede estar en el almacenamiento primario, pero es más probable que este en disco. A este proceso se le llama spooler.

Otro proceso lee las líneas del buffer circular y las escribe en la impresora trabajando a la velocidad de la impresora a este proceso a menudo se le llama despooler.

El buffer circular tiene suficiente espacio para absorber el desajuste resultante de la diferencia de velocidad entre spooler y despooler. Desde luego

suponemos que el sistema no genera indefinidamente, líneas más rápido de lo que la impresora pueda imprimirlas; si así fuera el buffer siempre estaría lleno y sería de poco valor mantener la continuidad de la operación de impresión.

- ◆ **Llamadas anidadas a monitores.-** Es posible que un sistema que utilice monitores permita a los procedimientos de un monitor llamar a procedimientos que estén en otro, con lo que se originan las llamadas anidadas a monitores.
- ◆ **Expresiones de ruta.-** La noción de expresiones de ruta se introdujo para poder especificar el orden en que se ejecutarán los procedimientos al tiempo que se mantienen separados de esta especificación los procedimientos.

Ejemplos:

`path leer end`

`path comenzar_lectura, leer, lectura_terminada end`

`path C:\DOS\WINDOWF_PROT96`

Cuando una expresión de ruta especifica una secuencia de llamadas, no implica que el mismo invocador tenga que hacer cada una de las llamadas.

- ◆ **Paso de mensajes.-** Las dos formas existentes de comunicación entre procesos son:

⇒ La memoria compartida

⇒ El paso de mensajes.

Los procesos envían y reciben mensajes mediante llamadas como las siguientes:

`enviar (proceso_receptor, mensaje);`

`recibir(proceso_transmisor, mensaje);`

Las llamadas enviar y recibir se realizan normalmente como llamadas al sistema operativo accesibles desde muchos ambientes de lenguajes de programación. La llamada a enviar puede señalar explícitamente el nombre del proceso receptor, o puede omitir el nombre, indicando así que el mensaje será difundido a todos los procesos.

- ◆ **Envío con bloqueo.-** Debe esperar a que el receptor reciba el mensaje (comunicación sincrónica). Una llamada a recibir con bloqueo obligará al receptor a esperar.
- ◆ **Envío sin bloqueo.-** Permite al transmisor continuar con otro procesamiento aunque el receptor no haya recibido aún el mensaje (comunicación asíncrona). Este tipo de envíos requiere un mecanismo de buffer para almacenar el mensaje hasta que lo reciba el receptor.

La comunicación asíncrona sin bloqueo aumenta el nivel de concurrencia. Una llamada a recibir sin bloqueo permite que el receptor continúe con otro procesamiento antes de volver a intentar la recepción.

Cuando se habló de comunicación entre procesos en el mismo computador siempre damos por asentado una transmisión sin errores, sin embargo, en los sistemas distribuidos la transmisión puede tener errores, incluso puede perderse, por ello los transmisores y receptores cooperan muchas veces utilizando un protocolo de reconocimiento para confirmar la recepción correcta de cada transmisión.

El transmisor que espera un reconocimiento del receptor puede utilizar un mecanismo de tiempo, cuando expira el tiempo si es que no ha llegado el reconocimiento el transmisor puede retransmitir el mensaje.

Una complicación de los sistemas distribuidos con paso de mensajes es la necesidad de dar nombres en forma no ambigua a los procesos, de forma que las llamadas explícitas a enviar y recibir hagan referencia a los procesos correctos. La creación y destrucción de procesos se puede coordinar mediante algún mecanismo centralizado de asignación de nombres pero tal cosa puede introducir una sobrecarga considerable de transmisión ya que cada máquina deberá solicitar permiso para utilizar nuevos nombres.

Una alternativa es asegurar que cada computador asigne nombres únicos a sus propios procesos; de esta manera puede hacerse referencia a los procesos combinando el nombre del computador con el nombre del proceso. Lograrlo, requiere un control centralizado para determinar un nombre único para cada computador de un sistema distribuido, pero esto, no representa una sobrecarga

importante, pues los computadores se añaden o eliminan de las redes con muy poca frecuencia.

Las comunicaciones a base de mensajes en los sistemas distribuidos presentan serios problemas de seguridad, uno de ellos es el problema de comprobar la autenticidad ya que siendo receptor o transmisor en realidad no se sabe si son impostores los que transmiten o los que reciben los mensajes.

Buzones y puertos

La comunicación a base de mensajes se puede establecer directamente entre receptor y transmisor, o pueden utilizar colas intermedias. En el caso de las colas intermedias, el transmisor y receptor especifican la cola, en lugar del proceso, con que se van a comunicar.

- ◆ **Buzón.-** Es una cola de mensajes que puede ser utilizada por múltiples transmisores y receptores. En un sistema distribuido, los receptores pueden estar situados en muchos computadores, por lo cual requieren dos transmisiones para la mayor parte de los mensajes, una del transmisor al buzón y otra del buzón al receptor. El problema se resuelve dando a cada receptor un puerto.
- ◆ **Puerto.-** Se define como un buzón utilizado por múltiples transmisores y un solo receptor. En los sistemas distribuidos, cada despachador tiene normalmente un puerto en el cual recibe solicitudes de muchos clientes.
- ◆ **Conductos.-** Unix introdujo la noción de conductos o pipes como un esquema para manejar comunicaciones entre procesos. Conducto es en esencia un buzón que permite extraer un número especificado de caracteres a la vez. No tiene la noción de mensaje, de modo que, si el usuario desea simular un buzón de mensajes con un conducto, deberá leer cada vez el número apropiado de caracteres (el tamaño del mensaje). Los conductos se utilizan con frecuencia para manejar las comunicaciones entre un proceso productor único y un solo consumidor.
- ◆ **Llamadas a procedimientos remotos.-** La noción de llamadas a procedimientos remotos o RPC (remote procedure calls) se introdujo con el

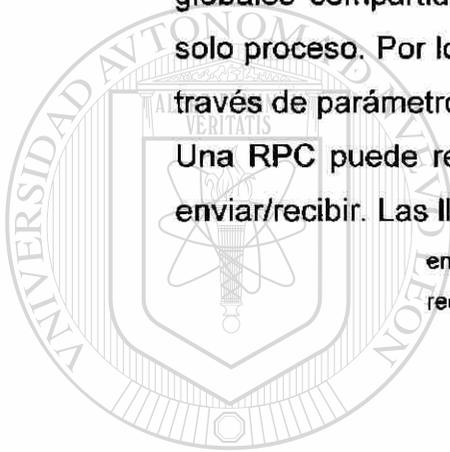
objeto de ofrecer un mecanismo estructurado de alto nivel para realizar comunicaciones entre procesos en sistemas distribuidos.

Con una llamada a un procedimiento remoto, un proceso de un sistema puede llamar a un procedimiento de un proceso de otro sistema. El proceso que llama se bloquea esperando el retorno desde el procedimiento llamado en el sistema remoto y después continúa su ejecución desde el punto que sigue inmediatamente a la llamada.

El procedimiento llamado y el que llama residen en procesos distintos, con espacios de direcciones distintos, por lo cual no existe la noción de variables globales compartidas, como en los procedimientos normales dentro de un solo proceso. Por lo tanto, las RPC transfieren la información estrictamente a través de parámetros de la llamada.

Una RPC puede realizarse aprovechando un mecanismo existente del tipo enviar/recibir. Las llamadas al procedimiento remoto sería:

```
enviar ( proceso_remoto, parámetros_de_entrada);  
recibir (proceso_remoto, parámetros_de_salida);
```



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

Unidad III Bloqueo Mutuo (Deadlocks)

Objetivo de esta unidad.- Durante el desarrollo de esta unidad, se analizarán los bloqueos de los procesos. El alumno comprenderá: ¿Qué es un bloqueo?, las condiciones que lo propician, las áreas de investigación de los bloqueos y temas asociados.

Bloqueos

En los sistemas de multiprogramación, el compartimiento de recursos es uno de los principales objetivos del sistema operativo. Cuando se comparten los recursos entre una población de usuarios, cada uno de los cuales mantiene un control exclusivo sobre ciertos recursos asignados a él, es posible que se produzcan bloqueos mutuos en que nunca podrán terminar los procesos de algunos usuarios.

Un bloqueo mutuo de tráfico, por ejemplo podríamos citar a las calles del centro de Monterrey:

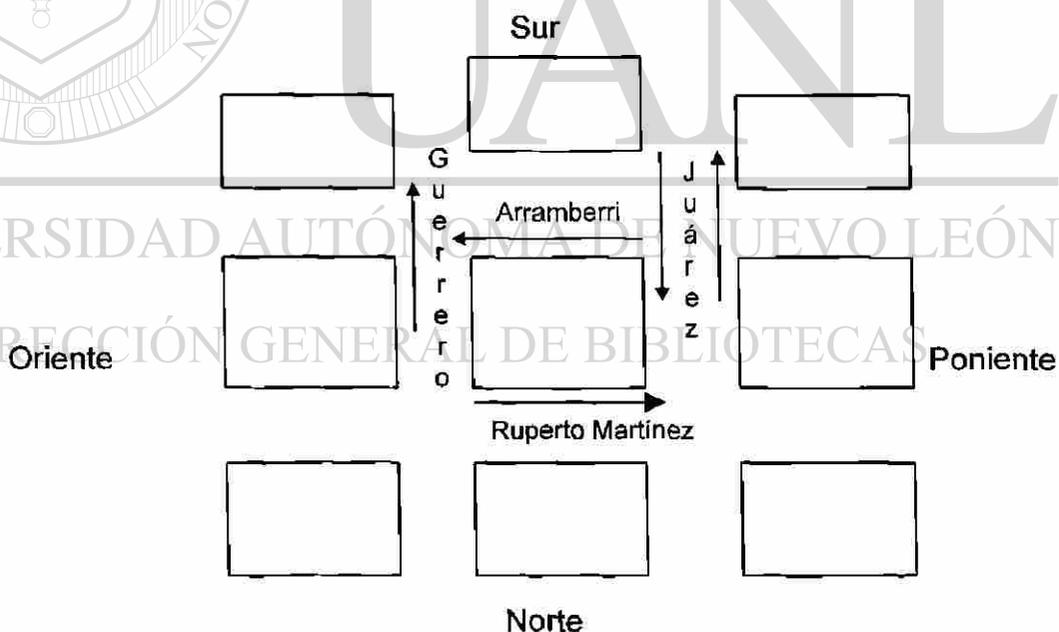


Fig. 3.1. Primer cuadro de la Cd. de Monterrey

El tráfico se detiene por completo, de poco o nada sirven los semáforos controladores del tráfico, siendo necesario la intervención del agente de tránsito para solucionar el embrollo alejando lenta y cuidadosamente los autos y

camiones que circulan por el área congestionada. El tráfico comienza a fluir normalmente, no sin antes haber provocado molestias, movilizaciones y una considerable pérdida de tiempo.

La mayor parte de los bloqueos mutuos de los sistemas operativos se presentan a causa de una competencia normal por los recursos dedicados (recursos que sólo pueden ser utilizados por un usuario a la vez, o sea, recursos reutilizables en serie).

Cada proceso espera que el otro libere un recurso que no liberará hasta que el otro libere su recurso, etc. Esta espera circular es característica de los sistemas en bloqueo mutuo. Dado que la tenaz retención de los recursos puede provocar un bloqueo mutuo, este a veces recibe el nombre de abrazo mortal.

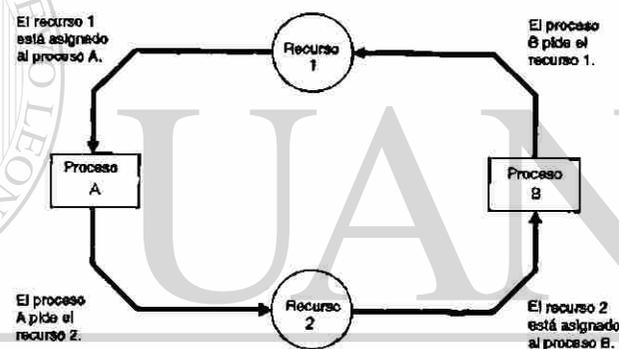


Fig.3.2. Abrazo Mortal

Bloqueo mutuo en un sistema de spool.- Un sistema de spool se utiliza para mejorar la producción de un sistema haciendo que un programa sea independiente de la baja velocidad de operación de los dispositivos tales como las impresoras.

Si un programa que envía líneas a la impresora debe esperar a que cada línea se imprima antes de poder enviar la siguiente, el programa se ejecutará lentamente. Para acelerar la ejecución del programa, las líneas de salida se envían a un dispositivo mucho más rápido, como un disco, donde se almacenan en forma temporal hasta su impresión. En algunos sistemas de spool la salida completa de un programa debe estar disponible antes de que comience realmente la acción de imprimir. De este modo, varias tareas

incompletas que se encuentren generando líneas de impresión hacia un archivo de spool pueden bloquearse mutuamente si el espacio se termina antes de que acabe cualquiera de las tareas.

La recuperación de un sistema por un bloqueo mutuo de esa naturaleza puede requerir el inicio completo del sistema con la consecuente pérdida de todo trabajo realizado hasta entonces. Si se bloquea de manera que el operador pueda tomar el control, existe la posibilidad de realizar una recuperación menos drástica mediante la desactivación o muerte de una o más tareas hasta disponer del espacio de spool suficiente para que puedan completarse los otros trabajos.

Cuando un diseñador de sistemas genera un sistema operativo, especifica el espacio para los archivos de spool. Una manera de reducir la posibilidad de un bloqueo mutuo en los sistemas de spool es reservar un espacio mucho mayor para los archivos spool que el considerado indispensable. Si el espacio es insuficiente no siempre es posible esta solución. Una solución más común es colocar un impedimento en la entrada del spooler para que no acepte más tareas en el momento en que los archivos de spool se acerquen a un umbral de saturación que podría ser, un 75% del espacio total por ejemplo. Esto reduciría

el rendimiento del sistema que es el precio que hay que pagar por reducir la posibilidad del bloqueo mutuo.

En cualquier sistema que mantenga los procesos en espera mientras se les asigna un recurso o se toman decisiones de planificación, la programación de un proceso puede postergarse indefinidamente mientras otro recibe la atención del sistema. Esta situación se le conoce como Aplazamiento indefinido, Postergación indefinida o Inanición, y es tan peligrosa como un bloqueo mutuo.

Puede ocurrir debido a predisposiciones en las políticas de planificación de recursos del sistema. Cuando los recursos se planifican por prioridad, es posible que un proceso dado espere en forma indefinida un recurso porque siguen llegando otros procesos con mayor prioridad. En algunos sistemas, el aplazamiento indefinido se evita aumentando la prioridad del proceso mientras

espera, a esto se le conoce como envejecimiento; en algún momento la prioridad de este superará la prioridad de otros procesos entrantes y el proceso en espera será atendido.

Conceptos de Recursos

Un sistema operativo es ante todo un administrador de recursos. Se encarga de la asignación de un amplio conjunto de recursos de varios tipos.

Recursos apropiables

Se consideran los recursos apropiables como el CPU y memoria principal. Un programa de usuario que ocupa un intervalo determinado de localidades dentro de la memoria principal puede ser expulsado por otro programa, el cual se apropia de la memoria. Un programa de usuario que haya realizado una solicitud de entrada/salida quizá no pueda aprovechar de manera efectiva la memoria principal durante el largo periodo en que se completa la operación de E/S. El CPU es quizá el recurso más apropiable de un sistema y debe conmutarse rápidamente entre un gran número de procesos que compiten por la atención del sistema a fin de que todos avancen a una velocidad razonable.

En el momento en que un proceso en particular no pueda utilizar de manera efectiva el CPU, perderá el control de este en favor de otro proceso.

La apropiación es de enorme importancia para el éxito de un sistema de cómputo multiprogramado.

Recursos no apropiables

No pueden arrebatarse al proceso al que han sido asignados. Por ejemplo, las unidades de disco, unidades de cinta que están asignadas a un proceso en particular por periodos de varios minutos u horas.

Algunos recursos se pueden compartir entre varios procesos y otros están dedicados a un solo proceso. Las unidades de disco están dedicadas a un solo proceso, pero a menudo contienen archivos pertenecientes a varios procesos. La memoria principal y el CPU son compartidos por muchos procesos; aunque

un solo CPU normalmente sólo puede pertenecer a un proceso a la vez, la conmutación de un CPU entre varios procesos crea la ilusión de un compartimiento simultáneo.

Los datos y programas son sin duda recursos que es necesario controlar y asignar. En los sistemas de multiprogramación, varios usuarios podrían utilizar al mismo tiempo un programa editor, siendo un desperdicio de memoria tener una copia del editor para cada programa, en lugar de ello se lleva a la memoria una sola copia del código y se hacen varias copias de los datos, una para cada usuario. El código no debe cambiar, pues pueden estarlo usando gran cantidad de personas al mismo tiempo. El código que no se puede modificar mientras se usa se llama reentrante. El código reentrante puede ser compartido simultáneamente por varios procesos.

El código que se puede modificar pero que vuelve a su forma original cada vez que se utiliza se llama reutilizable en serie. El reutilizable en serie solo puede ser utilizado por un proceso a la vez.

Cuando se dice que ciertos recursos son compartidos, debe especificarse si van a ser utilizados por varios procesos de manera simultánea o si pueden ser utilizados por varios procesos, pero sólo uno a la vez, siendo estos últimos los recursos que tienden a participar en bloqueos mutuos.

Cuatro condiciones necesarias para que exista un bloqueo mutuo

1. **Exclusión Mutua.-** Los procesos exigen un control exclusivo de los recursos que necesitan.
2. **Espera.-** Los procesos mantienen la posesión de los recursos ya asignados a ellos mientras esperan recursos adicionales.
3. **No-apropiación.-** Los recursos no pueden arrebatarse a los procesos a los que están asignados hasta que no termine su ejecución o su utilización.
4. **Espera circular.-** Existe una cadena circular de procesos en la cual cada proceso tiene uno o más recursos que son requeridos por el siguiente proceso en la cadena.

La existencia de un bloqueo mutuo implica que se han dado todas y cada una de las cuatro condiciones.

Áreas de investigación en los bloqueos mutuos

1. **Prevención.-** En la prevención de un bloqueo mutuo vamos a ajustar el sistema para eliminar toda la posibilidad de que ocurra un bloqueo mutuo pero esto puede declinar el aprovechamiento de los recursos o bien empobrecerlos.
2. **Evasión.-** En las técnicas para evitar el bloqueo mutuo el objetivo es imponer condiciones menos restrictivas que en la prevención para tratar de obtener un mejor aprovechamiento de los recursos. No implica ajustar previamente el sistema para eliminar todas las posibilidades de que se produzca aquel sino que permite la posibilidad de que ocurra el bloqueo mutuo pero se esquiva cuando esta a punto de suceder.
3. **Detección.-** Los métodos de detección del bloqueo mutuo se utilizan en sistemas que permiten la ocurrencia de los bloqueos mutuos ya sea de manera voluntaria e involuntaria. El objetivo es determinar si ha ocurrido un bloqueo mutuo y saber exactamente cuales son los procesos y recursos implicados en él, una vez que se determina, el bloqueo mutuo puede eliminarse del sistema.
4. **Recuperación.-** Los métodos de recuperación ante un bloqueo mutuo sirven para eliminar los bloqueos mutuos de un sistema para que pueda seguir trabajando y para que los procesos implicados puedan terminar su ejecución y liberen los recursos. Esto viene a ser un problema complejo ya que la recuperación se viene logrando en el mejor de los casos eliminando completamente uno o varios procesos. Después, se inician de nuevo los procesos eliminados, perdiéndose la mayor parte o todo el trabajo previo realizado por el proceso.

1^{er} Área de Investigación en los Bloqueos Mutuos: Prevención

La técnica más popular más popular empleada por los diseñadores para tratar el bloqueo mutuo es la prevención

J.W. Havender llegó a la conclusión de que si falta una de las cuatro condiciones necesarias no puede haber bloqueo mutuo y para negarlas sugiere lo siguiente:

1. Cada proceso deberá pedir todos sus recursos al mismo tiempo y no podrá continuar su ejecución hasta que no reciba todos.
2. Si a un proceso que tiene ciertos recursos se le niegan los demás, este proceso deberá liberar los recursos y, en caso necesario pedirlos de nuevo junto con los recursos adicionales.
3. Se impondrá un ordenamiento lineal de los tipos de recursos en todos los procesos, esto es, si a un proceso se le han asignado un tipo de recursos específico, en lo sucesivo solo podrá pedir aquellos recursos que siguen en el ordenamiento.

Nótese que se presentan tres estrategias y no cuatro debido a que no es deseable impedir la primera de las condiciones (exclusión mutua), porque específicamente queremos permitir la existencia de recursos dedicados.

1^{er} Estrategia de Havender.- Negación de la Condición de Espera.

La primera estrategia de Havender requiere que sean pedidos de una sola vez todos los recursos que un proceso necesita. El sistema debe proporcionarlos bajo el principio de todo o nada. Si no está disponible el conjunto completo de recursos, el proceso debe esperar. Mientras el proceso espera no puede tener ningún recurso, con esto se evita la condición de espera y no puede ocurrir el bloqueo mutuo, pero esto nos va a llevar a un grave desperdicio de recursos.

Ejemplo.- Considérese el caso en el cual un programa necesita solo una unidad de cinta al comienzo de su ejecución o peor aún ninguna y no necesitará del resto por varias horas. El requisito de que el programa pueda pedir y recibir todos los recursos antes de comenzar la ejecución significa que

tendremos recursos importantes ociosos durante horas. Dividir el programa en varios pasos de ejecución de manera relativamente independiente es una técnica empleada con frecuencia para conseguir una mejor utilización de los recursos bajo estas circunstancias. De esta forma, la asignación de recursos puede controlarse por etapas en lugar de hacerlo para todo el proceso, esto reduce efectivamente el desperdicio pero implica un trabajo extra en el diseño del sistema y complica la ejecución, nos puede provocar un aplazamiento indefinido ya que los recursos pueden no estar disponibles al mismo tiempo. El sistema debe permitir que un número suficiente de tareas terminen y liberen sus recursos antes de poder seguir ejecutando el proceso en espera. Mientras se acumulan los recursos no se pueden asignar a otros trabajos, motivo por el cual se desperdician.

2ª Estrategia de Havender.- Negación de la Condición de No Apropiación.

Supóngase que un sistema permite a un proceso conservar recursos mientras sigue pidiendo otros. En tanto haya suficientes recursos disponibles para satisfacer todas las peticiones, el sistema no puede entrar en bloqueo mutuo. Consideremos aquí cuando no se puede atender una petición, en ese momento, el proceso tiene recursos que un segundo proceso puede necesitar para continuar su ejecución, mientras el segundo proceso puede tener recursos que el primero necesita, dando origen a un bloqueo mutuo.

Esta segunda estrategia de Havender requiere que cuando a un proceso que tiene recursos se le niegan recursos adicionales, dicho proceso debe liberar los recursos que tiene y, si es necesario, pedirlos de nuevo junto con los recursos adicionales, anulando efectivamente la condición de no-apropiación. Los recursos se le puede quitar al proceso que los tiene antes de la terminación del proceso. Cuando un proceso libera recursos puede perder el trabajo realizado hasta ese momento lo cual nos puede llevar a pagar un costo demasiado alto. Otra consecuencia seria es el aplazamiento indefinido. Un proceso puede aplazarse indefinidamente mientras pide y libera muchas veces los recursos, si

esto llegara a ocurrir el proceso podría eliminarse por el sistema a fin de que otros procesos puedan ejecutarse.

3ª Estrategia de Havender.- Negación de la Condición de Espera Circular

Como todos los recursos tienen una numeración única y como los procesos deben pedir en orden lineal ascendente es imposible que se presente una espera circular.

Los recursos deben pedirse en orden ascendente por número de recurso. El número de recurso es asignado por la instalación y debe tener un tiempo de vida largo (meses o años); si se agregan nuevos tipos de recursos, puede ser necesario reescribir los programas y sistemas.

Por lógica, cuando se asigna el número de recurso, estos deben reflejar el orden normal en que los usan la mayoría de las tareas; si espera que las tareas que siguen este orden tengan una operación eficiente. Pero las tareas que necesitan los recursos en forma diferente que el previsto en la instalación, los deberán adquirir y conservar, quizá durante mucho tiempo antes de utilizarlo realmente, lo cual se puede traducir en desperdicio.

Uno de los objetivos de más importancia en los sistemas operativos actuales es crear ambientes amables con el usuario, los usuarios deben ser capaces de desarrollar sus aplicaciones con un mínimo de interferencia en sus ambientes por las molestas restricciones de hardware y software. El ordenamiento lineal de Havender, elimina la posibilidad de una espera circular, pero afecta la capacidad del usuario para escribir libre y fácilmente el código de sus aplicaciones.

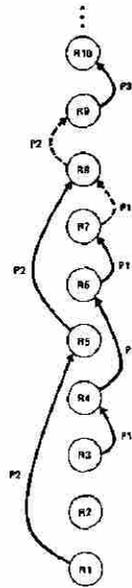


Fig. 3.3. Ordenamiento lineal

2ª ÁREA DE INVESTIGACIÓN EN LOS BLOQUEOS MUTUOS: Evasión

Si se presentan las condiciones necesarias para que ocurra un bloqueo mutuo, es posible aún evitarlo mediante una cuidadosa asignación de recursos. El algoritmo del banquero de Dijkstra cuyo nombre atañe precisamente las funciones de un banquero que otorga préstamos y recibe pagos a partir de una determinada fuente de capital. Aquí se implementa en el contexto de la asignación de recursos.

El algoritmo del banquero.

El algoritmo del banquero puede generalizarse a lotes de recursos del mismo tipo.

Ejemplo. Consideremos la asignación de una cantidad, t , de unidades de cinta idénticas.

1. Un sistema operativo comparte un número fijo, t , de unidades de cinta equivalente entre un número fijo, u , de usuarios. Cada usuario especifica por adelantado el número máximo de unidades de cinta que necesitará durante la ejecución de su trabajo en el sistema.

2. El sistema operativo aceptará la petición de un usuario si la necesidad máxima de ese usuario no es mayor que t .
3. Un usuario puede obtener o liberar unidades de cinta una a una. Algunas veces un usuario puede verse obligado a esperar para obtener una unidad adicional, pero el sistema operativo garantiza una espera finita. El número real de unidades de cinta asignadas a un usuario nunca será superior a la necesidad máxima declarada por ese usuario.
4. Si el sistema operativo es capaz de satisfacer la necesidad máxima del usuario, entonces éste debe garantizar al sistema operativo que las unidades de cinta serán utilizadas y liberadas en un tiempo finito.
5. Se dice que el estado actual del sistema es seguro si el sistema operativo puede permitir que todos sus usuarios actuales terminen sus trabajos en un tiempo finito, de no ser así el estado es inseguro (estamos suponiendo que las unidades de cinta son los únicos recursos requeridos por el usuario).

Supóngase que existen n usuarios.

- Sea **préstamo(i)** la representación del **préstamo** actual de unidades de cinta para el usuario i ; si el usuario 5, por ejemplo, tiene asignadas cuatro unidades de cinta, entonces **préstamo(5) = 4**.
- Sea **máx(i)** la necesidad máxima del usuario i , de manera que si el usuario 3 tiene una necesidad máxima de dos unidades de cinta, entonces **máx(3) = 2**.
- Sea **petición(i)** la petición actual del usuario, la cual es igual a su máxima necesidad menos su préstamo actual.

Si el usuario 7, por ejemplo tiene una necesidad máxima de 6 unidades y un préstamo actual de 4, entonces

$$\text{petición}(7) = \text{máx}(7) - \text{préstamo}(7) = 6 - 4 = 2$$

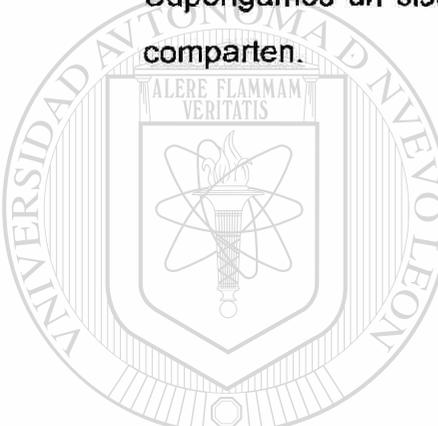
El sistema operativo controla t unidades de cinta.

- Sea a el número de unidades de cinta todavía disponibles para asignar. Entonces a es igual a t menos la suma de todos los préstamos a todos los usuarios.

El algoritmo del banquero permite la asignación de recursos cuando esta conduzca a estados seguros. Un estado seguro es aquel en el que la situación total de recursos es tal que todos los usuarios serán capaces de terminar en algún momento.

Ejemplo de estado seguro.

Supongamos un sistema con doce unidades de cinta y tres usuarios que las comparten.



	Préstamo Actual	Necesidad Máxima
Usuario 1	1	4
Usuario 2	4	6
Usuario 3	5	8
Disponibles = 2		

Estado seguro en el algoritmo del banquero Dijkstra

Es seguro porque es posible que terminen los tres usuarios. Si las unidades disponibles se le asignan al usuario 2, al terminar liberará 6 unidades con lo que es posible que terminen los demás usuarios. Es decir, la clave para que un estado sea seguro es que exista al menos una forma de que terminen todos los usuarios.

	Préstamo Actual	Necesidad Máxima
Usuario1	8	10
Usuario2	2	5
Usuario3	1	3
Disponibles = 1		

Estado inseguro en el algoritmo del banquero Dijkstra

Supongamos el mismo sistema con doce unidades de cinta y tres usuarios que las comparten. En este caso once de las unidades de cinta están en uso y solo queda una unidad disponible para ser asignada. No importa a cual usuario se le asigne, no se puede garantizar que terminen los tres usuarios. Es importante señalar que un estado inseguro no implica la existencia, ni siquiera eventual, de un bloqueo mutuo. Lo que sí implica un estado inseguro es simplemente que alguna secuencia desafortunada de eventos podría llevar al bloqueo mutuo.

	Préstamo Actual	Necesidad Máxima
Usuario1	1	4
Usuario2	4	6
Usuario3	5	8
	2	

SEGURO

La clave para que un estado sea seguro es que exista al menos una forma de que terminen todos los usuarios.

	Préstamo Actual	Necesidad Máxima
Usuario1	8	10
Usuario2	2	5
Usuario3	1	3
	1	

INSEGURO

Un estado inseguro no implica la existencia ni siquiera eventual de un bloqueo mutuo. Lo que sí implica un estado inseguro es simplemente que alguna secuencia desafortunada de eventos podría llevarnos a un bloqueo mutuo.

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
DIRECCIÓN GENERAL DE BIBLIOTECAS

Saber que un estado es seguro no implica que serán seguros todos los estados futuros. La política de asignación de recursos debe considerar todas las peticiones antes de satisfacerlas. Por ejemplo, si en el cuadro del estado seguro en lugar de satisfacer la necesidad del usuario 2 se le otorgarán al usuario 3 o al usuario 1, ya no se podría garantizar la terminación de todos los procesos.

Asignación de recursos con el algoritmo del banquero

Están permitidas las condiciones de espera circular, espera y no-apropiación, pero los procesos si exigen el uso exclusivo de los recursos que requieren. Los procesos pueden conservar recursos mientras piden y esperan recursos adicionales. Los recursos no pueden arrebatare a los procesos que los tienen. El sistema puede satisfacer o rechazar cada petición. Si una petición es rechazada, el usuario conserva los recursos ya asignados y espera un tiempo finito a que se satisfaga su petición. El sistema solo satisface peticiones que lleven a estados seguros ya que una petición que condujese a un estado inseguro se rechazaría repetidamente hasta que pudiera quedar satisfecha. Como el sistema siempre se mantiene en estado seguro, tarde o temprano todas las peticiones podrán ser atendidas y los usuarios terminaran.

Defectos del algoritmo del banquero

Permite la ejecución de tareas que tendrían que esperar en una situación de prevención del bloqueo mutuo como las siguientes:

1. El algoritmo requiere un número fijo de recursos asignables.
2. Requiere de una población constante de usuarios.
3. El algoritmo requiere que el banquero satisfaga todas las peticiones en un tiempo finito. Es evidente que los sistemas reales se necesitan mayores garantías.
4. El algoritmo requiere que los clientes (trabajos) salden sus préstamos en un tiempo finito. Se necesitan mayores garantías en los sistemas de tiempo reales.
5. El algoritmo requiere que los usuarios declaren por anticipado sus necesidades máximas.

3ª ÁREA DE INVESTIGACIÓN: DETECCIÓN

La detección del bloqueo mutuo es el proceso de determinar si realmente existe un bloqueo mutuo e identificar los procesos y recursos implicados en él. Los

algoritmos de detección del bloqueo mutuo determinan por lo general si existe una espera circular y son demasiado costosos.

Para facilitar la detección de bloqueos mutuos, se utilizará una notación en la cual una gráfica dirigida indica las asignaciones y peticiones de recursos. Los cuadrados representan procesos, los círculos grandes, clases de dispositivos idénticos; los círculos pequeños en el interior de los grandes indican el número de dispositivos de cada clase.

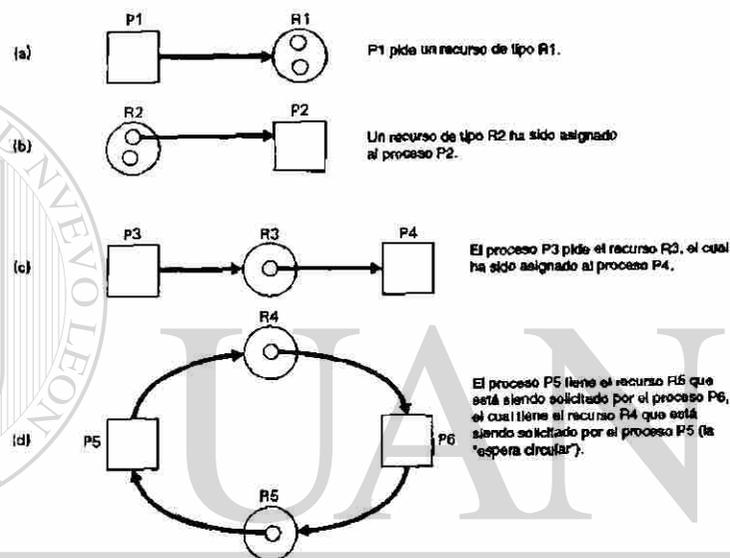


Fig. 3.4. Gráficas de Asignación de recursos

Las gráficas de asignación y solicitud de recursos cambian cuando los procesos piden recursos, los adquieren y luego los devuelven al sistema operativo.

Reducción de las Gráficas de Asignación de Recursos

Una técnica útil para detectar los bloqueos mutuos implica reducciones de gráficas en las que se determinan los procesos que pueden terminar su ejecución y los que se encuentran en bloqueo mutuo.

Si pueden atenderse las peticiones de recursos de un proceso, se dice que la gráfica puede ser reducida por ese proceso. Es equivalente a mostrar la gráfica como si el proceso hubiera terminado su ejecución y devueltos los

recursos al sistema. La reducción se muestra eliminando las flechas de los recursos hacia el proceso eliminando las flechas de ese proceso hacia los recursos.

Si una gráfica puede ser reducida por todos sus procesos, entonces no hay bloqueo mutuo. Si una gráfica no puede ser reducida por todos sus procesos, los procesos irreducibles constituyen el conjunto de procesos en bloqueo mutuo de la gráfica.

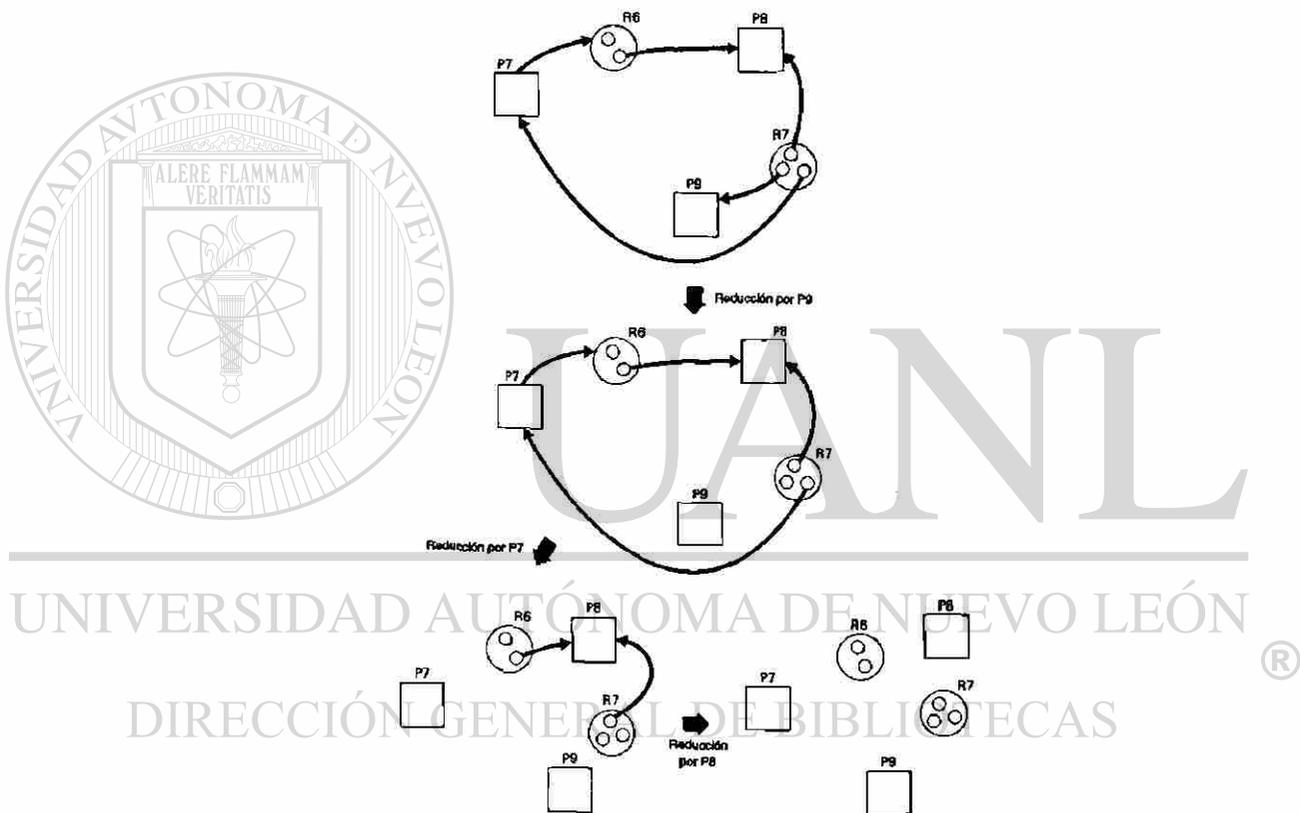


Fig. 3.5. Reducción de gráficas de asignación

Es importante señalar aquí que no importa el orden en el cual se realizan las reducciones el resultado final es siempre el mismo.

4ª ÁREA DE INVESTIGACIÓN: Recuperación

El bloqueo mutuo debe romperse mediante la eliminación de uno o más de las condiciones necesarias. Por lo general, varios procesos perderán una parte o la

totalidad del trabajo que se ha efectuado pero es el precio pagado que puede ser pequeño comparado con las consecuencias de permitir que nuestro proceso siga bloqueado. La recuperación después de un bloqueo mutuo puede ser complicada debido a:

1. Puede no estar claro que el sistema se haya bloqueado.
2. La mayor parte de los sistemas tienen medios muy deficientes para suspender indefinidamente un proceso, eliminarlo del sistema y reanudarlo más tarde. De hecho, algunos procesos como los de tiempo real, que deben funcionar en forma continua, sencillamente no se pueden suspender y reanudar.
3. Aún cuando existieran medios efectivos de suspensión/reanudación, con toda seguridad implicarían un gasto extra considerable y requerirían la atención de un operador altamente calificado. No siempre es posible encontrar un operador así.
4. La recuperación después de un bloqueo mutuo de dimensiones modestas puede significar una cantidad razonable de trabajo, en tanto que un bloqueo mutuo a gran escala puede requerir una cantidad enorme de trabajo.

En los sistemas actuales, la recuperación se suele realizar eliminando un proceso y arrebatándole sus recursos. Generalmente, el proceso eliminado se pierde, pero es posible concluir los procesos restantes. Algunas veces es necesario eliminar varios procesos hasta que se haya liberado los recursos suficientes para que terminen los procesos restantes.

El término "recuperación" es un tanto inapropiado ya que se aniquilan procesos en beneficio de otros. Los procesos pueden eliminarse de acuerdo con algún orden de prioridad, pero también se presentan algunas dificultades, tales como:

1. Es posible que no existan prioridades entre los procesos bloqueados, de modo que el operador necesita tomar una decisión arbitraria.
2. Las prioridades pueden ser incorrectas o un poco confusas debido a consideraciones especiales como la planificación a plazo fijo, en la cual un

proceso de prioridad relativamente baja tiene una prioridad temporalmente alta a causa de un fin de plazo inminente.

3. La determinación de una decisión óptima sobre los procesos por eliminar puede requerir un esfuerzo considerable.

Ejemplo.- Podría ser necesario apagar temporalmente un sistema y reiniciarlo a partir de ese punto sin pérdida de trabajo productivo. La suspensión/reanudación sería muy útil en este caso. Las características de un punto de verificación/reinicio disponibles en muchos sistemas facilitan la suspensión/reanudación sólo con la pérdida de trabajo desde el último punto de verificación (la última grabación del estado del sistema).



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

Unidad IV

ALMACENAMIENTO REAL

Objetivo de esta unidad.- Durante el desarrollo de esta unidad, se analizarán las técnicas más comunes en las asignaciones y relocalizaciones de programas y datos en memoria principal. El alumno entenderá las técnicas para el manejo de memoria real y su aplicación para sistemas operativos de un solo usuario o para sistemas de memoria virtual; deberá comprender los conceptos de paginación, segmentación y memoria asociativa, para entender el funcionamiento de la memoria virtual en un sistema operativo.

Por sencillez consideremos:

Memoria = Almacenamiento

Memoria Principal = Memoria Real = Memoria Primaria

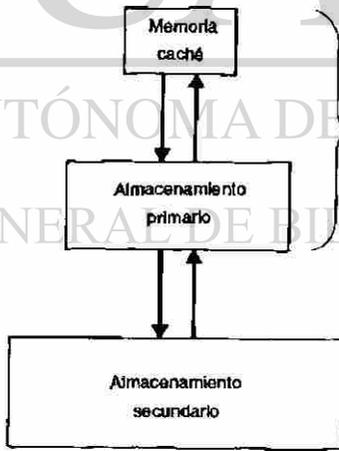
Organización de almacenamiento se entiende que es la forma en que se considera el almacenamiento principal.

Disminuye el tiempo de acceso al almacenamiento.

Aumenta la velocidad de acceso al almacenamiento.

Aumenta el costo de almacenamiento por bit.

Disminuye la capacidad de almacenamiento.



La UCP puede referirse directamente a los programas y datos.

Los programas y datos se deben pasar primero al almacenamiento principal para que la UCP pueda referirse a ellos.

Fig. 4.1. Jerarquía del almacenamiento

Estrategias de Administración de Almacenamiento

1. Estrategias de Obtención
2. Estrategias de Colocación
3. Estrategias de Reemplazo

- ◆ **Estrategias de Obtención.-** Determinan cuando debe obtenerse la siguiente parte del programa o los datos que se van a transferir del almacenamiento secundario al principal. La estrategia convencional es la **obtención por demanda** en la cual la siguiente parte del programa o de los datos se transfiere al almacenamiento principal cuando un programa en ejecución hace referencia a ella.

Se tiene la creencia de que como generalmente no se puede predecir hacia donde pasará el control de un programa, el trabajo extra para hacer superposiciones y anticipar el futuro superaría por mucho los beneficios esperados aunque se dice que mejorará el rendimiento de los sistemas la **obtención anticipada**.

- ◆ **Estrategias de Colocación.-** Tienen que ver con la determinación de la parte del almacenamiento principal donde se colocara un programa entrante (Primer ajuste, Mejor ajuste y Peor ajuste).
- ◆ **Estrategias de Reemplazo.-** Están relacionadas con la determinación de qué parte del programa o de los datos se debe desalojar para dejar espacio a los programas entrantes.

Asignación de almacenamiento contiguo y no contiguo

Los primeros sistemas de computo requerían una asignación de almacenamiento contiguo, es decir, cada programa tenía que ocupar un único bloque contiguo de localidades de memoria.

En la asignación de almacenamiento no contiguo un programa se divide en varios bloques o segmentos que se pueden colocar en el almacenamiento principal en fragmentos que no necesitan ser adyacentes. Para el sistema operativo es más difícil controlar esta asignación pero la ventaja es que dispone de varios huecos pequeños en lugar de un solo hueco grande.

Asignación de almacenamiento contiguo para un usuario único

Los primeros sistemas de computo permitían que solo un usuario a la vez utilizara la máquina, todos los recursos estaban a su disposición. Como el usuario disponía de toda la máquina, se le cobraban todos los recursos. El mecanismo de cobro normal se basaba en el tiempo de reloj.

Originalmente cada usuario escribía todo el código necesario para realizar una aplicación específica detallando entradas/salidas, etc. Muy pronto el código de entrada/salida requerido se consolidó en un sistema de control de entrada/salida. Los usuarios que querían hacer E/S ya no tenían la necesidad de codificar directamente las instrucciones, sino que llamaban a la rutina de IOCS para que ejecutara el trabajo.

El tamaño de los programas está limitado por la cantidad de memoria principal pero es posible ejecutar programas más grandes utilizando el concepto de superposiciones. Si una sección de un programa no se necesita durante el resto de la ejecución del programa se puede traer del almacenamiento secundario otra sección para ocupar el espacio utilizado por la que ya no se necesite.

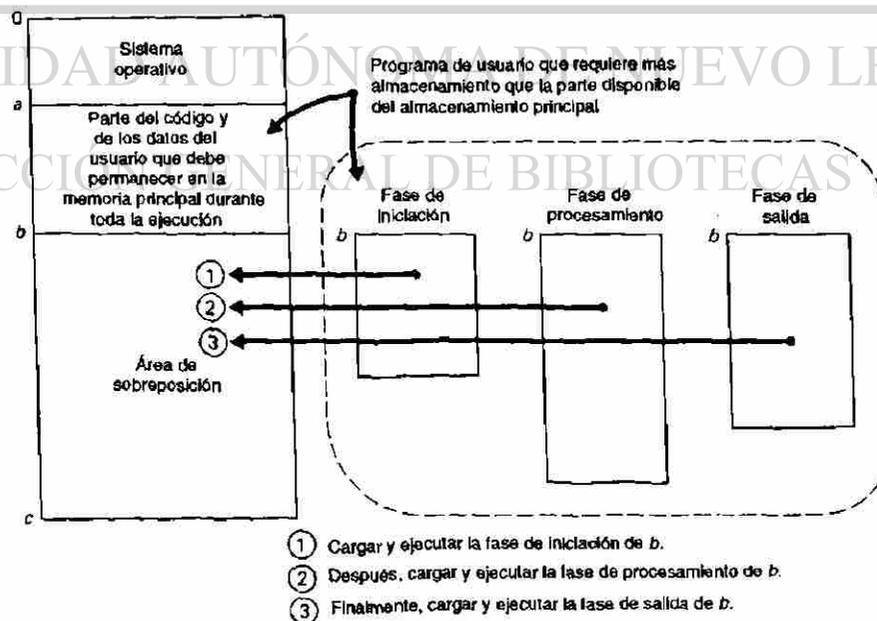


Fig. 4.2. Funcionamiento del concepto de superposiciones

Protección de los sistemas de usuario único

Como se tiene control absoluto sobre todo el almacenamiento principal, si el programa del usuario se desvía puede destruir al sistema operativo, si esto resulta fatal y el usuario no puede proseguir sabrá que algo anda mal, terminará la ejecución, arreglará el problema y volverá a intentar la ejecución de su programa.

Supóngase que el usuario destruye al sistema operativo en forma más sutil, por ejemplo, supóngase que ciertas rutinas de E/S son modificadas por accidente, de tal modo que pueden estar truncados todos los registros de salida, el trabajo puede seguir su ejecución pero los resultados no serán examinados sino hasta que se termina el trabajo, entonces se estarán desperdiciando recursos de máquina y en el peor de los casos ese daño a nuestro sistema operativo puede ser que nos produzcan errores poco evidentes. La protección se realiza con un registro de límites único integrado en el CPU. Cada vez que un programa de usuario hace referencia a una dirección de memoria se revisa el registro de límites para verificar que el usuario no está a punto de destruir el sistema operativo. Si el usuario intenta entrar al sistema operativo se intercepta la instrucción y el trabajo termina con un mensaje de error apropiado.

El usuario necesita entrar de vez en cuando al sistema operativo para utilizar ciertos servicios, como por ejemplo la E/S, el problema se resuelve dando al usuario una instrucción específica con la cual puede solicitar los servicios del sistema operativo, es decir, una llamada al supervisor.



Fig. 4.3. Protección de los sistemas de usuario único

Procesamiento por lotes de flujo único

Los trabajos en general requerían un tiempo de preparación, se cargaba el sistema operativo, se montaban las cintas y los discos, se preparaban impresoras, etc y también requerían de largos tiempos de descarga. El computador estaba ocioso todo ese tiempo. Los diseñadores se dieron cuenta que podían optimizar esos tiempos de transición entre procesos o trabajos pudiendo reducir la cantidad de tiempo que se desperdiciaba en los trabajos, esto condujo al desarrollo de los sistemas de procesamiento por lotes (BATCH).

En el procesamiento por lotes de flujo único, los trabajos se agrupan en lotes cargándolos consecutivamente en cinta o en disco. Un procesador de flujo de trabajos lee las instrucciones en JCL (Job Control Language) y facilita la preparación para el siguiente trabajo. Cuando termina un trabajo, el lector de flujo de trabajos lee automáticamente las instrucciones del lenguaje de control para el siguiente trabajo y realiza las acciones de mantenimiento apropiadas para facilitar la transición del siguiente trabajo.

Multiprogramación de particiones fijas

Los diseñadores ven una vez más que se puede aprovechar el CPU notablemente mediante una administración intensiva. Esta vez decidieron administrar sistemas de multiprogramación en los cuales los usuarios compiten al mismo tiempo con los recursos del sistema. El trabajo que esta esperando la terminación de un proceso de E/S cederá el CPU a otro trabajo que este listo para realizar cálculos si es que hay uno en espera, de este modo pueden efectuarse simultáneamente las operaciones de E/S en los cálculos del CPU.

Para aprovechar al máximo la multiprogramación es necesario que varios trabajos residan al mismo tiempo en el almacenamiento principal del computador, de este modo cuando un trabajo solicita E/S el CPU puede conmutarse a otro y realizar cálculos sin retraso.

Multiprogramación con particiones fijas: traducción y cargas absolutas

En cada partición podía haber solo un trabajo, el CPU se conmutaba rápidamente entre los usuarios para crear la ilusión de simultaneidad. Los trabajos se traducían con compiladores y ensambladores absolutos para ejecutarse solo en una partición específica. Si un trabajo estaba listo para ser ejecutado y su partición estaba ocupada tenía que esperar aunque estuvieran libres otras particiones, esto nos provoca desperdicio de memoria o de almacenamiento.

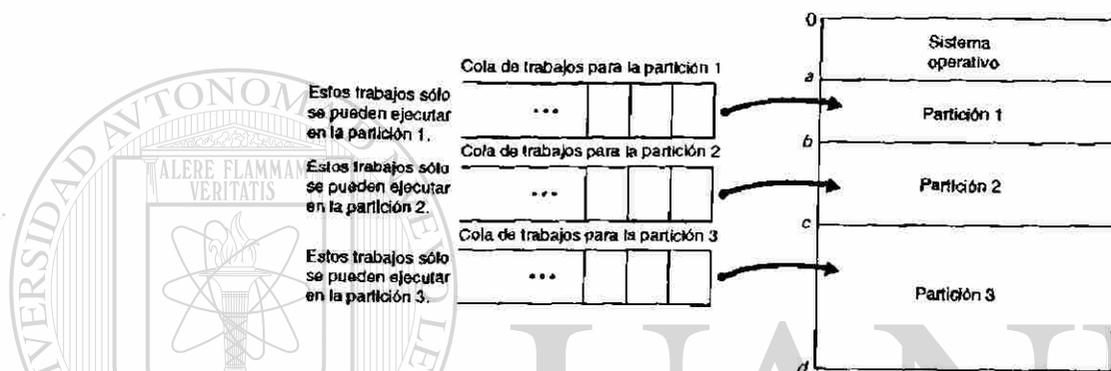
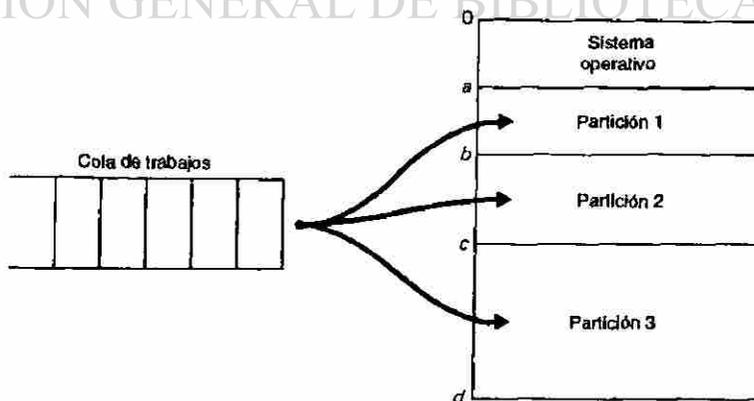


Fig.4.4. Multiprogramación con particiones fijas: traducción y cargas absolutas

Multiprogramación con particiones fijas: traducción y carga con reubicación

Los compiladores, ensambladores y cargadores con reubicación sirven para producir programas que se puedan ejecutar en cualquier partición disponible que sea lo suficientemente grande para contenerlos.



Los trabajos se pueden colocar en cualquier partición disponible en la que quepan.

Fig.4.4. Multiprogramación con particiones fijas: traducción y cargas con reubicación

Protección en los sistemas con multiprogramación

Esto se logra a menudo con varios registros límites. Con dos registros se pueden establecer los límites superior e inferior de la partición de un usuario o bien el límite superior o el inferior y la longitud de la región. El usuario que necesita llamar al sistema operativo utiliza una instrucción de llamada al supervisor para hacerlo, esto permite al usuario cruzar el límite del sistema operativo y solicitar sus servicios sin poner en peligro la seguridad del sistema operativo.

Fragmentación en la multiprogramación con particiones fijas

En todos los sistemas de computo se presenta la fragmentación, sea la organización de almacenamiento que sea, en los sistemas de multiprogramación y particiones fijas la fragmentación ocurre porque los trabajos de los usuarios no llenan por completo las particiones designadas o porque una partición es demasiado pequeña para contener un trabajo en espera.

Multiprogramación con particiones variables

Proporciona una mejora evidente al permitir que los trabajos ocupen tanto espacio como necesiten pero no todo el almacenamiento real. No tiene límites fijos.

Solo se tratarán los esquemas de asignación contigua, un trabajo debe ocupar localidades de memoria contiguas. En este tipo de multiprogramación no se hacen suposiciones con el tamaño. Si cuando llega un trabajo los mecanismos de planificación deciden que debe proseguir, se le asigna tanto espacio como necesite, en teoría no hay desperdicio.

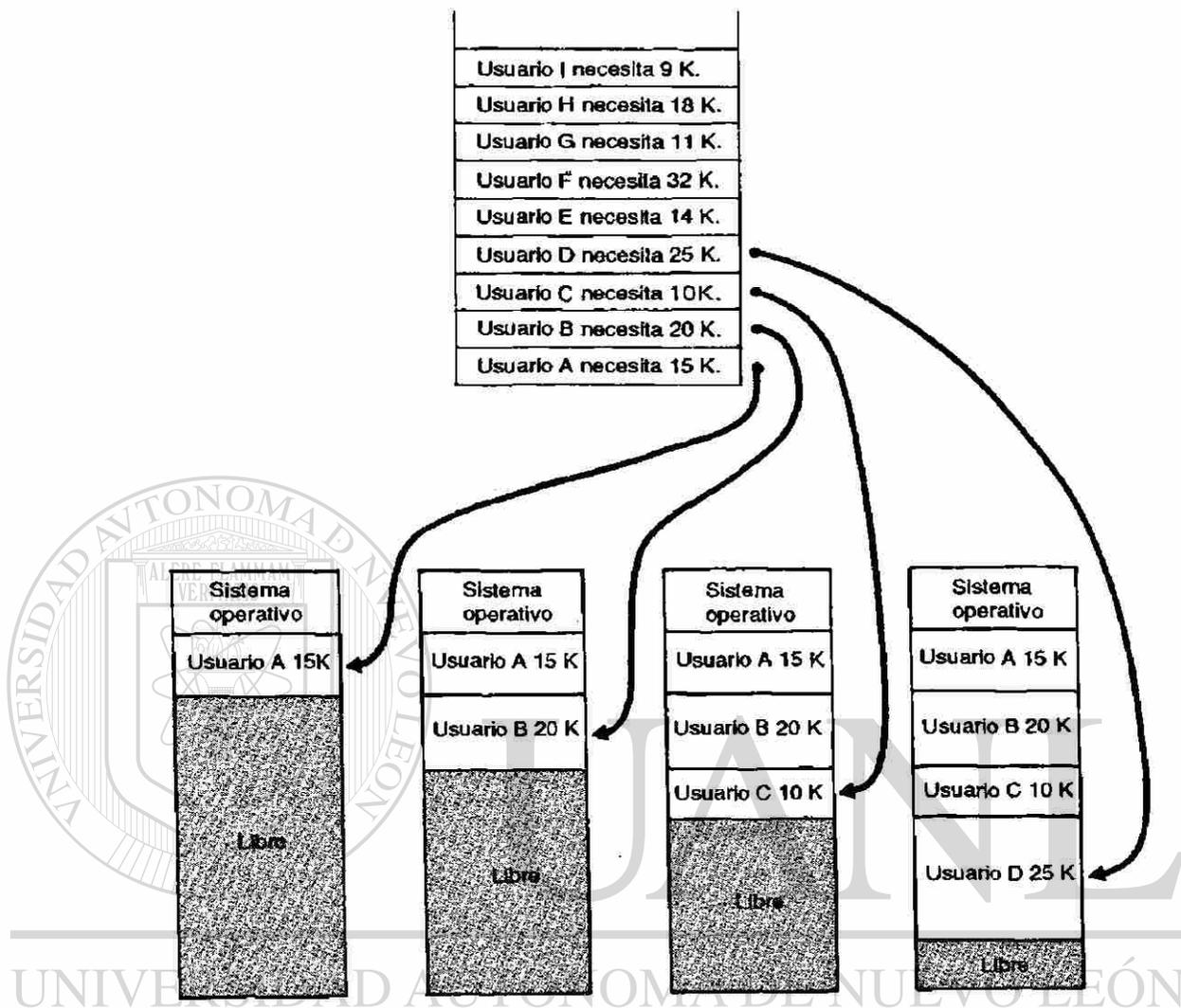


Fig. 4.6. Multiprogramación con particiones variables

DIRECCIÓN GENERAL DE BIBLIOTECAS

Condensación de huecos

Cuando termina un trabajo en un sistema de multiprogramación con particiones variables se puede comprobar si el almacenamiento liberado colinda con otras áreas libres de almacenamiento, comúnmente denominados huecos.

El proceso de fusionar huecos adyacentes para formar un solo hueco más grande se le denomina condensación. Mediante la condensación de huecos se puede recuperar los bloques contiguos de almacenamiento más grande que sea posible.

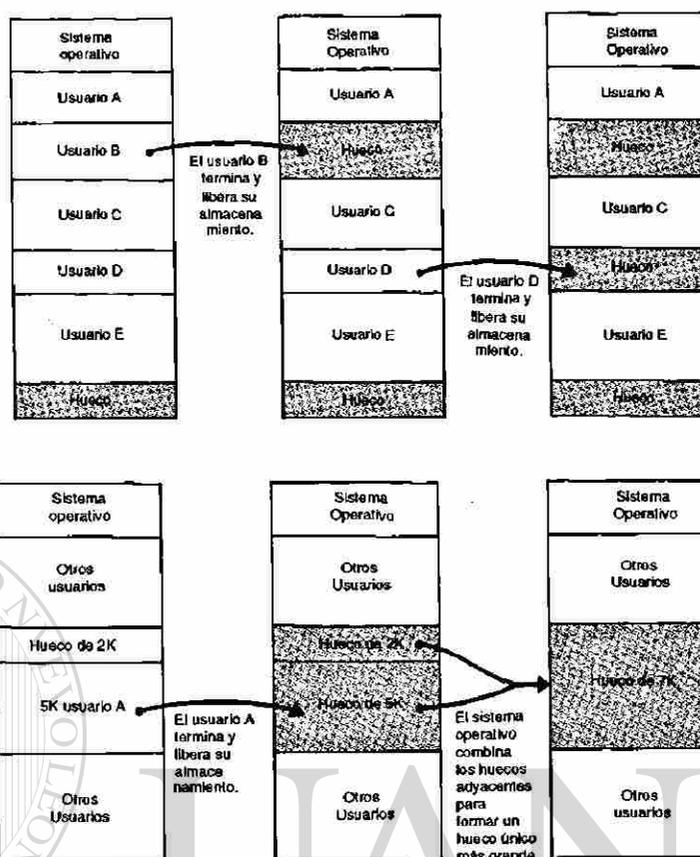


Fig. 4.7. Condensación de huecos

Compactación del almacenamiento

La técnica de computación del almacenamiento implica trasladar todas las áreas ocupadas del almacenamiento en algún extremo de la memoria principal. Esto deja un gran hueco único de almacenamiento libre, en lugar de numerosos huecos pequeños característicos de la multiprogramación con particiones variables. Ahora todo el almacenamiento libre está contiguo, así que un trabajo en espera puede ejecutarse si sus necesidades de memoria son satisfechas por el hueco único resultante. De manera pintoresca se le llama eructo de almacenamiento (Burping Storage) o recolección de basura (Garbage Collection).

Inconvenientes:

- Consume recursos del sistema que podrían utilizarse en forma más productiva.

- El sistema debe detener todas sus actividades mientras realiza la compactación provocando tiempos de respuesta irregulares para usuarios interactivos y podría ser devastador en sistemas de tiempo real.
- La compactación implica reubicar los trabajos que estén en el almacenamiento. Esto significa que la información requerida para realizar la reubicación que se pierde por lo regular cuando se carga un programa, debe mantenerse accesible ahora.
- Con una combinación normal de trabajos que cambia rápidamente, a menudo es necesario compactar. Los recursos del sistema que se consuman quizá no justifiquen las ventajas de la compactación

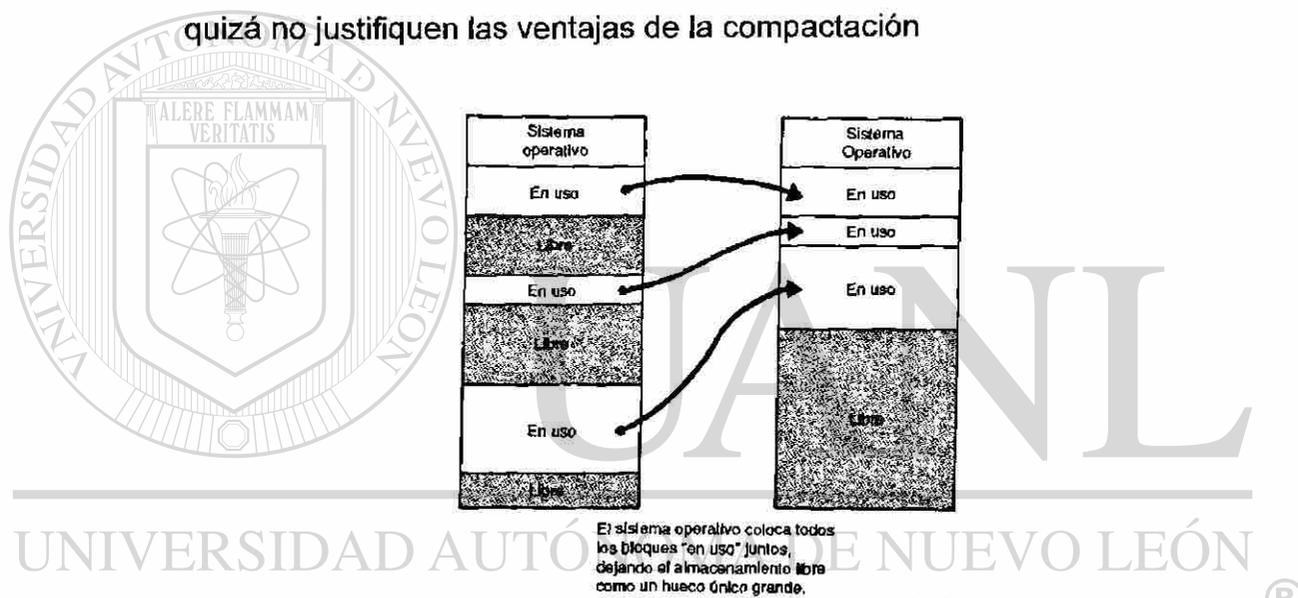


Fig. 4.8. Compactación del almacenamiento

Estrategias de colocación en el almacenamiento

Las estrategias de colocación en el almacenamiento sirven para determinar en qué lugar del almacenamiento principal se deben colocar los programas o los datos entrantes.

- ◆ **Estrategia del primer ajuste.**- Un trabajo que entre al sistema se coloca en el almacenamiento principal en el primer hueco disponible lo bastante grande para contenerlo. Intuitivamente es una estrategia atractiva porque permite decidir con rapidez en donde se coloca el trabajo.

- ◆ **Estrategia del mejor ajuste.-** Un trabajo que entre al sistema se colocará en el almacenamiento principal en el hueco que quepa mejor y quede la menor cantidad posible de espacio sin utilizar. Intuitivamente es la estrategia más atractiva.
- ◆ **Estrategia del peor ajuste.-** A primera vista es una elección extravagante, pero tras un examen cuidadoso también posee su atractivo. El peor ajuste consiste en colocar un programa en el almacenamiento primario en el hueco donde peor se ajusta, es decir, en el hueco más grande. Su atractivo de manera sencilla consiste en que después de colocar el programa en ese gran hueco, el hueco restante también será grande y por consiguiente también podrá contener un nuevo programa relativamente grande.

Una variante de la estrategia del primer ajuste es la llamada estrategia del siguiente ajuste. Comienza cada búsqueda de un hueco disponible en el lugar donde terminó la búsqueda anterior.

Multiprogramación con intercambio de almacenamiento

En algunos sistemas de intercambio (SWAPPING) un trabajo que ocupa el almacenamiento principal de una vez. Ese trabajo se ejecuta hasta que no puede continuar y entonces cede el almacenamiento y el CPU al siguiente trabajo. Así todo el almacenamiento está dedicado a un solo trabajo por un periodo breve, después se saca ese trabajo (se intercambia con el de fuera) y se carga el siguiente (se intercambia con el de adentro). Un trabajo normalmente será intercambiado en ambos sentidos repetidas veces antes de completarse.

ORGANIZACION DEL ALMACENAMIENTO VIRTUAL

REAL	REAL		VIRTUAL		
Sistemas dedicados a un solo usuario	Almacenamiento real en sistemas de multiprogramación		Almacenamiento virtual en sistemas de multiprogramación		
	Multiprogramación con particiones fijas	Multiprogramación con particiones variables	Paginación pura	Segmentación pura	Paginación/Segmentación combinada
	Absolutas	Reubicables			

Fig. 4.9. Evolución de las organizaciones de almacenamiento

Todos los sistemas de almacenamiento virtual tienen la propiedad de que las direcciones calculadas o programas en ejecución no son necesariamente disponibles en el almacenamiento primario, de hecho, las direcciones virtuales se seleccionan por lo regular de un conjunto de direcciones mucho más grande que el disponible en el almacenamiento primario.

Conceptos básicos

El concepto clave de almacenamiento virtual es la disociación de las direcciones a las que hace referencia un proceso en ejecución de las direcciones disponibles en el almacenamiento primario.

Las direcciones a las que hace referencia un proceso en ejecución se conocen como direcciones virtuales. Las direcciones disponibles en el almacenamiento primario se conocen como direcciones reales.

El intervalo de las direcciones virtuales a las que se puede hacer referencia en un proceso en ejecución se le conoce como espacio de direcciones virtuales V . El intervalo de direcciones reales disponibles en un sistema de cómputo se conoce como espacio de direcciones reales R del computador.

El número de direcciones en V se denota como $|V|$. El número de direcciones en R se denota como $|R|$.

Aunque los procesos solo hacen referencia a direcciones virtuales deben ejecutarse en almacenamiento real. Por lo tanto, es preciso establecer la

correspondencia entre direcciones reales y direcciones virtuales durante la ejecución de un proceso. Se han desarrollado varios métodos para asociar las direcciones virtuales con las reales. Los mecanismos de traducción dinámica de direcciones (Dynamic Address Translation) convierten las direcciones virtuales en direcciones reales mientras se ejecuta un proceso.

Todos estos sistemas tienen la propiedad de que las direcciones contiguas en el espacio de direcciones virtuales de un proceso no son necesariamente contiguas en el almacenamiento real, a esto se le conoce como **contigüidad artificial**.

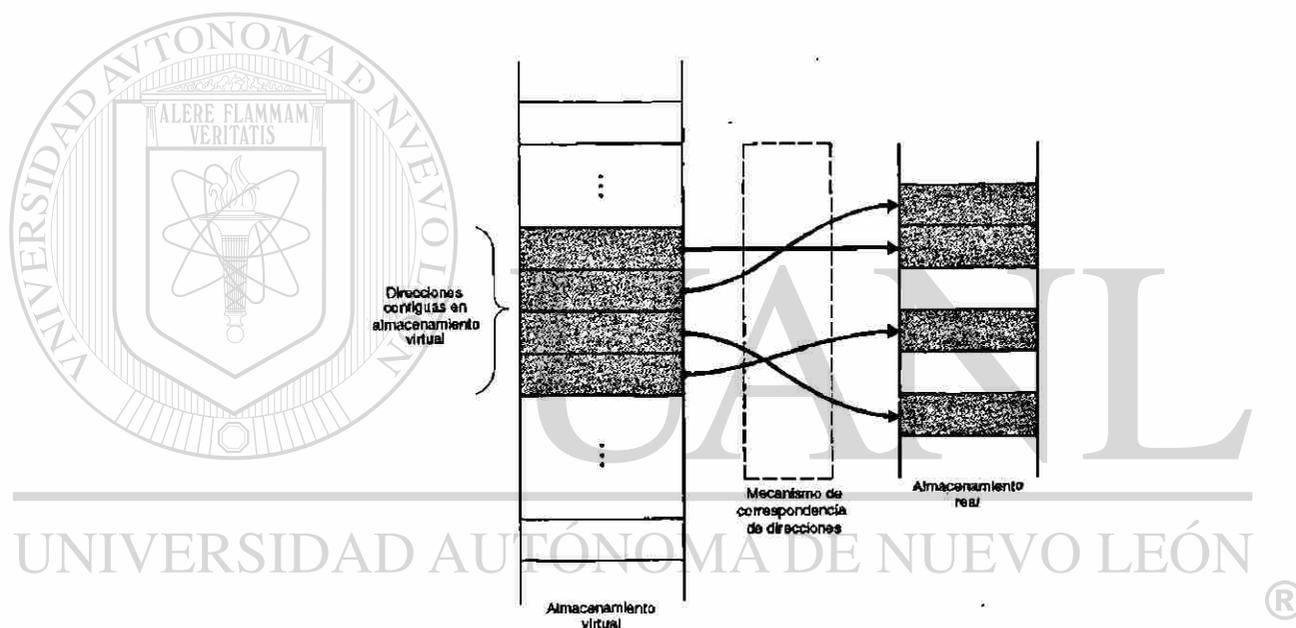


Fig. 4.10. Contigüidad artificial

Organización en almacenamiento con múltiples niveles

Si se va a permitir que el espacio de direcciones virtual del usuario que sea más grande que el espacio de direcciones reales y sobre todo si se va a multiprogramar un sistema en el cual muchos usuarios comparten el recurso del almacenamiento real hay que proporcionar lo necesario para proporcionar lo necesario para obtener los programas y datos en un almacenamiento auxiliar grande.

El primer nivel es el almacenamiento real en el cual se ejecutan los procesos y en el cual deben de encontrarse los datos para que un proceso en ejecución

pueda hacer referencia a ellos. El segundo nivel consiste en medios de almacenamiento de gran capacidad como discos y tambores capaces de almacenar programas que no quepan en el almacenamiento real en un momento dado, se conoce también como almacenamiento auxiliar o secundario.

Correspondencia de bloques

Los mecanismos de traducción dinámica de direcciones deben tener mapas de correspondencias de traducción de direcciones que muestren cuales localidades del almacenamiento virtual están en el almacenamiento real en un momento dado y donde se encuentran. Si esta correspondencia se estableciera palabra por palabra o byte por byte, es decir, si existiera una entrada en el mapa de correspondencias por cada entrada en V , entonces la información de correspondencias sería tan voluminosa que requeriría tanto espacio del almacenamiento real como los propios procesos o más.

Como no podemos darnos el lujo de establecer una correspondencia de este tipo la información se agrupará en bloques y el sistema lleva un registro del lugar del almacenamiento real en donde se han colocado los diversos bloques

de almacenamiento virtual, cuanto mayor sea el bloque menor será la fracción de almacenamiento real dedicada a guardar la información de correspondencia.

Al hacer bloques más grandes se reduce el gasto extra de almacenamiento del mecanismo de correspondencias pero los bloques grandes tardan más en ser transferidos del almacenamiento secundario al primario y consumen más almacenamiento real por lo que limitan el número de procesos que pueden compartir el almacenamiento real.

Número de bloque b	Desplazamiento d
-------------------------	-----------------------

Fig. 4.11. Representación gráfica de la dirección virtual $v=(b,d)$

La traducción de una dirección de almacenamiento virtual $v = (b,d)$ a una dirección real r se ejecuta de la siguiente forma:

1. Cada proceso tiene su propia tabla de correspondencia de bloques mantenida por el sistema dentro del almacenamiento real.
2. Un registro especial dentro de la unidad de procesamiento llamado registro de origen de la tabla de correspondencia de bloques se carga con la dirección real, a , de la tabla de correspondencia de bloques del proceso durante la conmutación de contexto (cambio de contexto).
3. La tabla contiene una entrada por cada bloque del proceso y las entradas siguen un orden secuencial (b_0, b_1, b_2, \dots , etc.)
4. Ahora se suma el número de bloque "b" a la dirección base "a" de la tabla de bloques para formar la dirección real de la entrada del bloque "b" en la tabla de correspondencia de bloques. Esta entrada contiene la dirección "b" del inicio del bloque "b".
5. El desplazamiento "d" se suma a la dirección de inicio del bloque "b" para formar la dirección real deseada $r = (b' + d)$.

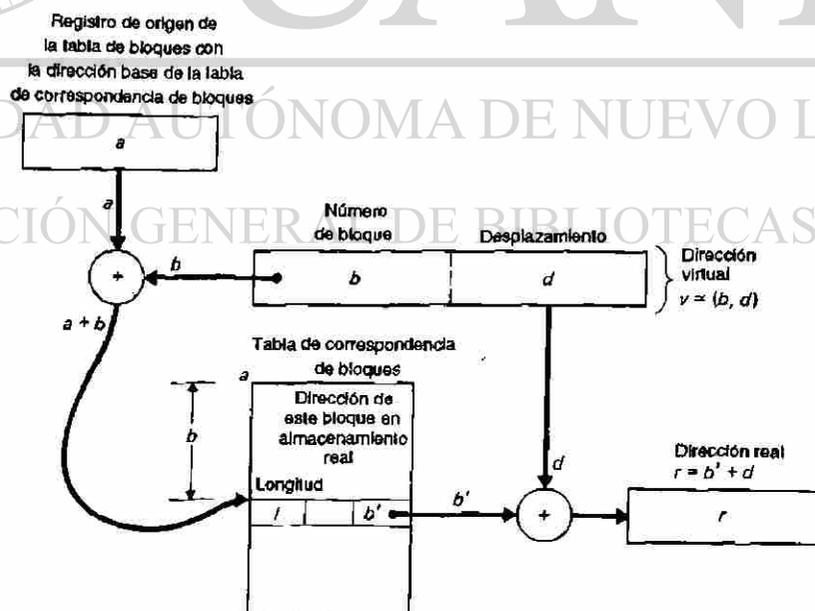


Fig.4.12. Traducción de una dirección virtual $v=(b,d)$ a una real r

Conceptos básicos de paginación

Los bloques de tamaño fijo se llaman páginas y la organización del almacenamiento virtual correspondiente se le conoce como paginación. Una dirección virtual en un sistema de paginación es un par ordenado (p,d) en el cual "p" es el número de página del almacenamiento virtual en el que reside el elemento al que se hace referencia y "d" es el desplazamiento dentro de la página "p" donde se localiza dicho elemento.

Número de página p	Desplazamiento d
-----------------------	---------------------

Fig. 4.13. Representación gráfica de la dirección virtual $v=(p,d)$

Un proceso puede ejecutarse si su página activa se encuentra en el almacenamiento primario. Las páginas se transfieren del almacenamiento secundario al primario y se colocan dentro de bloques llamados marcos de página que tienen el mismo tamaño. Los marcos de página comienzan en direcciones de almacenamiento real que son múltiplos enteros de tamaño fijo de la página. Una página entrante puede colocarse en cualquier marco de página disponible.

La traducción dinámica de direcciones en la paginación se realiza de la siguiente manera:

Un proceso en ejecución hace referencia a una dirección de almacenamiento virtual $v = (p,d)$. Un mecanismo de correspondencia de páginas busca la página "p" en una tabla de correspondencias de páginas y determina que la página "p" se encuentra en el marco de página "p' ". La dirección del almacenamiento real se forma concatenando "p' " y "d".

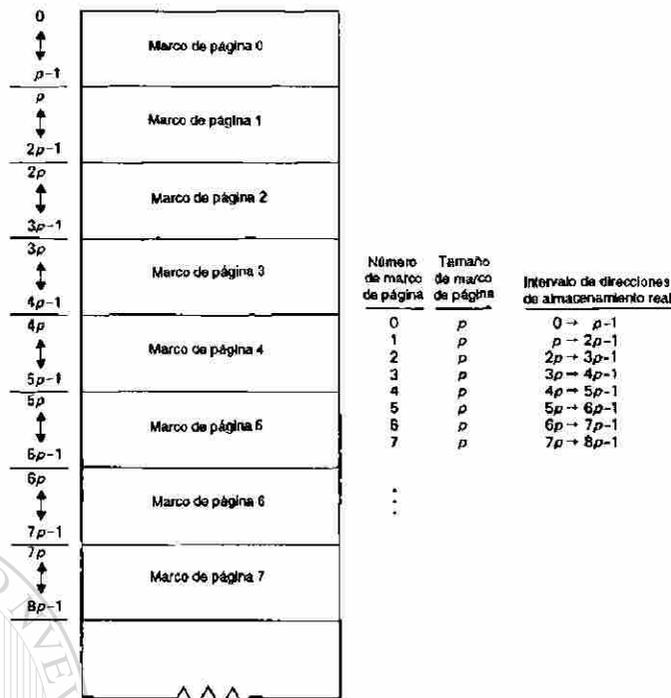
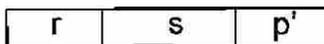


Fig. 4.14 División del almacenamiento en marcos de página

Debido a que normalmente no todas las páginas del proceso se encuentran en el almacenamiento primario al mismo tiempo la tabla de correspondencia de páginas debe indicar si la página a la que se hace referencia se encuentra o no en el almacenamiento primario, si sí esta donde se encuentra y si no donde puede encontrarse en el secundario.



r = bit de residencia de página
 s = dirección en almacenamiento secundario
 p' = número de página

Fig. 4.15. Entrada de la tabla de correspondencia de páginas

Un bit de residencia de página “r” tiene valor cero si la página no se encuentra en el almacenamiento primario y valor 1 en caso contrario, si la página no se encuentra en el almacenamiento primario, entonces “s” es su dirección en el almacenamiento secundario, si la página se encuentra en el almacenamiento secundario, si la página se encuentra en el almacenamiento primario, entonces “ p’ “ es un número de marco de página.

La traducción dinámica de direcciones en la paginación implica la correspondencia entre el número de página "p" y el marco de página "p' ". La correspondencia puede ser directa en cuyo caso se mantiene en una tabla de correspondencia de páginas completa en el almacenamiento primario o en una memoria cache de acceso rápido. La correspondencia puede ser también puramente asociativa en cuyo caso la tabla de correspondencia se mantiene en un almacenamiento asociativo de acceso rápido. Debido al alto costo de los almacenamientos asociativos y caché la correspondencia puede ser una combinación asociativa directa donde solo se mantienen en el almacenamiento asociativo de alta velocidad las partes más recientes y se recurre a una correspondencia directa en almacenamiento primario cuando falla la búsqueda asociativa.



Fig.4.16. Traducción de una dirección virtual $v=(p,d)$ a una real r

Compartimiento en un sistema de paginación

El compartimiento en un sistema con paginación se logra al hacer que las entradas de las tablas de correspondencia de procesos diferentes apunten al mismo marco de página. Esta forma de compartir es inconveniente porque en

realidad los procesos comparten entidades lógicas como lo son procedimientos y datos que pueden ocupar varias páginas y crecer o decrecer mientras se ejecuta el proceso.

Conceptos básicos de segmentación

Emplea bloques de tamaño variable y cada bloque es tan grande como sea necesario dentro de su límite para contener una entidad lógica como procedimientos y datos. Los segmentos de un proceso no necesitan estar todos al mismo tiempo en el almacenamiento principal ni tienen por fuerza que ser contiguos.

Número de segmento s	Desplazamiento d
------------------------	--------------------

Fig. 4.17. Representación gráfica de la dirección virtual $v=(s,d)$

Una dirección virtual en un sistema de segmentación o con segmentación es un par ordenado $v = (s,d)$ donde "s" es el segmento donde se encuentra "v" y "d" es el desplazamiento de "v" dentro de "s". La traducción dinámica puede realizarse con los mismos tipos de correspondencia directa, asociativa o combinada.

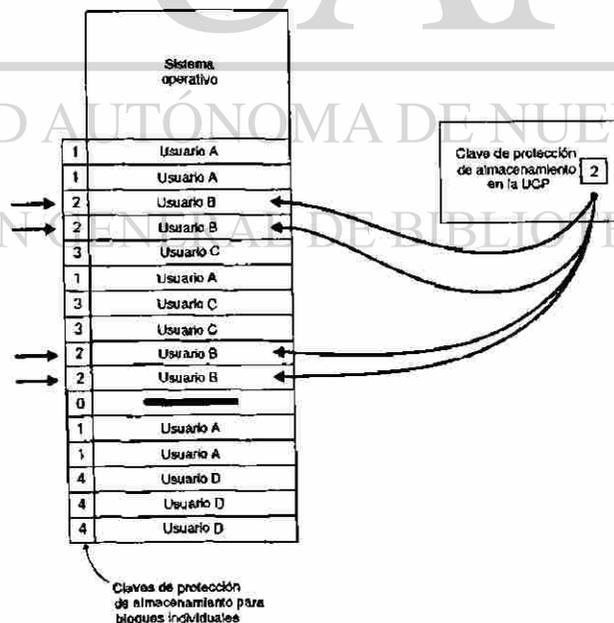


Fig. 4.18 Manejo de claves para protección de bloques

La protección en los sistemas con segmentación es más natural que en la paginación ya que se protegen entidades lógicas no físicas. A los procesos se le otorgan varias combinaciones de acceso de lectura, escritura, ejecución y adición a los distintos segmentos.

Tipos de acceso	Abreviatura	Explicación
Lectura	R	Este segmento se puede leer
Escritura	W	Este segmento se puede modificar
Ejecución	E	Este segmento se puede ejecutar
Adición	A	Se puede agregar información al final de este segmento

Fig. 4.19. Tipos de control de acceso

r	a	l	R	W	E	A	s'
---	---	---	---	---	---	---	----

r = Bit Residencia
 l = Long. Segmento
 a = Direcc. Almac. Sec

s' = Direcc. Base Del Segmento
 R, W, E y A= Bit Protección

Fig. 4.20. Entrada de la tabla de correspondencia de segmentos

Un bit de residencia "r" indicara si el segmento se encuentra o no en el almacenamiento primario, si sí se encuentra en el almacenamiento primario "s'" es la dirección. Si no esta en el almacenamiento primario, entonces "a" es la dirección en el almacenamiento secundario donde debe leerse ese segmento antes de que el proceso pueda continuar con su ejecución. Todas las referencias al segmento deben verificarse contra la longitud del segmento "l" para asegurarse de que se encuentra dentro de él. En cada referencia al segmento se revisan también los bits de protección para ver si esta permitida la operación que intenta.

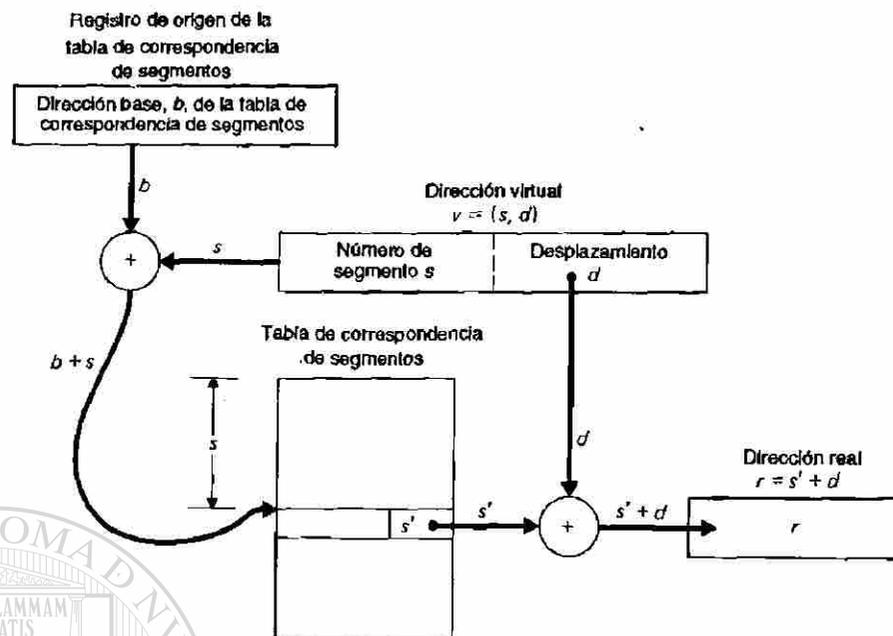


Fig.4.20. Traducción de una dirección virtual $v=(s,d)$ a una real r

Durante la traducción dinámica de direcciones una vez localizada la tabla de correspondencia, la entrada para el segmento "s" se examina primero "r" para comprobar si el segmento está en el almacenamiento primario, si el segmento no se encuentra allí se genera **falla por falta de segmento**, lo cual hace que el sistema operativo asuma el control y cargue el segmento al que se hizo referencia desde la dirección "a" del almacenamiento secundario.

Cuando el segmento está cargado continúa la traducción de la dirección comprobando si el desplazamiento "d" es menor o igual a la longitud del segmento "l", si no es igual se genera una **excepción por desbordamiento de segmento** lo que también provoca que el sistema operativo tome el control y termine con la ejecución del proceso. El último paso es sumar la dirección "s'" del almacenamiento primario al desplazamiento "d" para formar la dirección del almacenamiento real $r = s' + d$, si esto no está permitido se genera una **excepción por protección del segmento** lo que hace que el sistema operativo tome el control y termine la ejecución.

Paginación/segmentación combinada o sistemas de paginación segmentación

Una dirección virtual con sistema de paginación/segmentación es una tripleta ordenada $V=(s,p,d)$, donde "s" es el segmento en el que reside V, "p" es la página de "s" que contiene a "V" y "d" es el desplazamiento dentro de "V" dentro de "p". En estos sistemas casi siempre se usa un almacenamiento asociativo de acceso rápido o una memoria caché para evitar que la traducción dinámica de direcciones degrade seriamente el rendimiento.

Número de segmento	Número de página	Desplazamiento
s	p	d

Fig. 4.22. Representación gráfica de la dirección virtual $v=(s,p,d)$

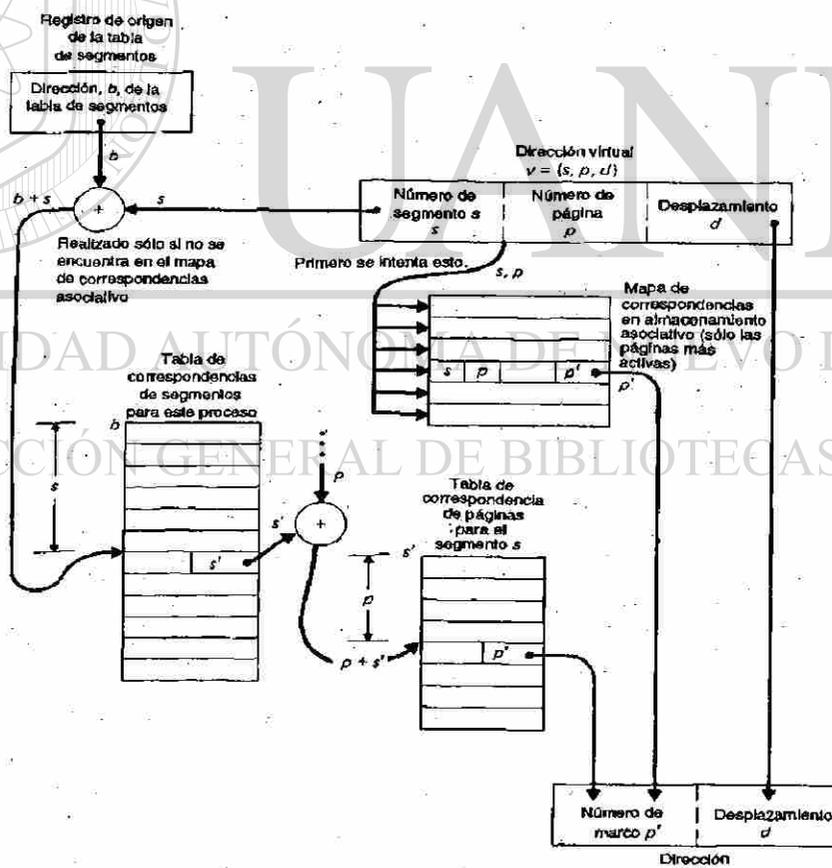


Fig.4.23. Traducción de una dirección virtual $v=(s,p,d)$ a una real r

ADMINISTRACION DEL ALMACENAMIENTO VIRTUAL

Estrategias de Reemplazo:

1. Principio de Optimalidad.
2. Reemplazo aleatorio de páginas.
3. Primeras Entradas Primeras Salidas (FIFO).
4. La menos recientemente utilizada (LRU).
5. La menos frecuentemente utilizada (LFU).
6. La no utilizada recientemente (NUR).
7. Segunda Oportunidad
8. Por Reloj
9. Conjuntos de Trabajo
10. Frecuencia de fallas de página (PFF).

Principio de Optimalidad.

Establece que para obtener un rendimiento óptimo la página que se debe reemplazar es aquella que tardará más tiempo en volver a ser utilizada.

Reemplazo de Páginas Aleatorio.

Si se busca una estrategia de reemplazo de páginas con un gasto extra reducido y que no discrimina ciertos usuarios, el reemplazo de páginas aleatorio es una técnica sencilla para lograrlo ya que todas las páginas que se encuentran en el almacenamiento principal tienen la misma oportunidad de ser elegidas para el reemplazo. Esta estrategia puede seleccionar cualquier página para ser reemplazada incluyendo la página a la que se hará referencia enseguida.

Primeras Entradas Primeras Salidas.

Aquí, para cada página se registra el instante en el que entró al almacenamiento primario. Cuando se necesite reemplazar una página se escoge a la que haya permanecido en el almacenamiento primario el mayor

tiempo. Esto parece razonable ya que cada página ha tenido su oportunidad y es tiempo de darle oportunidad a otra página, lamentablemente es probable que se reemplace páginas muy utilizadas ya que si una página permanece en el almacenamiento por mucho tiempo, puede deberse a que está en uso constantemente.

Anomalia PEPS

Parece razonable pensar que cuantos más marcos de página se asignen a un proceso menos fallos de página experimentará éste. Sin embargo, se descubrió que al utilizar el reemplazo de páginas PEPS ciertos patrones de referencia a páginas originan más fallas de página cuando aumenta el número de marcos de página asignados a un proceso, a este fenómeno se le llama anomalía PEPS o anomalía Belady.

La Menos Recientemente Utilizada (LRU).

Se selecciona para su reemplazo a aquella página que no ha sido utilizada el mayor tiempo, la estrategia se basa en la heurística de localidad según la cual el pasado reciente es un buen indicador del futuro cercano de modo que LRU exige que se marque cada página con el instante en que se hace referencia a ella, lo que origina mucho trabajo adicional por lo que pese a ser atractiva no se usa en los sistemas actuales, lo que se hace es utilizar estrategias que se aproximen a la LRU y que no ocasionen grandes costos.

La página menos recientemente utilizada podría ser la siguiente en ser utilizada, si un programa ocupa un ciclo importante que ocupe varias páginas. Si se reemplazara la página LRU se encontraría a sí mismo transfiriendo esa página de nuevo al almacenamiento principal casi de inmediato.

La Menos Frecuentemente Utilizada (LFU).

Es una aproximación a la LRU en la que es importante la intensidad con la que se ha utilizado una página, la página por reemplazar aquella que se le ha usado menos frecuentemente ó la que se ha hecho referencia con menos frecuencia.

La página menos frecuentemente utilizada podría ser aquella que se trajo del almacenamiento secundario al principal más recientemente. Esta página ha sido utilizada una sola vez mientras que las demás páginas que se encuentran en el almacenamiento principal pudieron haber sido utilizadas en varias ocasiones. Este mecanismo de reemplazo de página reemplazará esta página aunque tenga una probabilidad de ser usada de inmediato.

La No Utilizada Recientemente (NUR).

Es una aproximación a LRU con un poco de trabajo extra. No es probable que sean utilizadas pronto, las páginas que no se han utilizado recientemente por lo que puede reemplazarse con páginas reentrantes.

Es deseable reemplazar una página que no ha sido modificada mientras estaba en el almacenamiento primario, la estrategia NUR se llevará a la práctica con dos bit de hardware, estos son: el bit de referencia y el bit de modificación; en el bit de referencia el valor va a ser 0 si no se ha hecho referencia a la página o 1, si sí se ha hecho referencia, el bit de modificación va a ser 0 si no ha sido modificada y 1, si sí ha sido modificada. Al bit de modificación se le llama "bit sucio".

Segunda Oportunidad.

Es una variante de PEPS. Examina el bit de referencia de la página más antigua, si este vale cero se selecciona de inmediato para ser reemplazado. Si el bit de referencia vale 1 se le asigna el valor cero y la página se pasa al final de la lista y se considera en esencia como una página nueva, ésta página gradualmente irá avanzando hacia el inicio de la lista donde será reemplazada solamente si su bit de referencia vale cero (0).

Por Reloj.

La variación por reloj del algoritmo de la segunda oportunidad dispone las páginas en una lista circular en lugar de una lista lineal, un apuntador a la lista se desplaza alrededor de la lista circular en la misma forma que giran las manecillas de un reloj cuando el bit de referencia de una página toma el valor de cero (0) el apuntador se mueve al siguiente elemento de la lista.

Localidad.- El concepto de localidad indica que los procesos tienden a hacer referencia a la memoria en patrones no uniformes y altamente localizados. La localidad se manifiesta tanto en el espacio como en el tiempo. Es una propiedad empírica observada más que teórica.

Localidad Temporal.- Es la localidad en el tiempo. Significa que es probable que las localidades de memoria a las que se haya hecho referencia recientemente sean objeto de otra referencia en un futuro cercano, como apoyo de esta observación se puede mencionar:

- ◆ Los ciclos.
- ◆ Las subrutinas.
- ◆ Las pilas.
- ◆ Las variables de cuenta y totalización.

Localidad Espacial.- Significa que los elementos cercanos tienden a ser similares. También significa que las referencias a memoria tienden a estar concentradas y una vez que se hace referencia a una localidad es muy probable que se haga referencia a las localidades cercanas, como apoyo a esta observación:

- ◆ Los recorridos de los arreglos.
- ◆ La ejecución secuencial de código.
- ◆ La tendencia de algunos programadores a colocar las definiciones de variables afines próximas unas de otras.

Conjuntos de trabajo.

Denning desarrolló el conjunto de programas de trabajo para explicar el compartimiento de las páginas de los programas en términos de localidades. Los conjuntos de trabajo intentan mantener el conjunto de trabajo de un proceso (las páginas a las que se hizo referencia recientemente) en el almacenamiento primario de tal manera que el proceso se ejecutará con rapidez. Los procesos nuevos se pueden iniciar solo si hay espacio en el almacenamiento primario para sus conjuntos de trabajo. Los procesos que intentan ejecutarse sin espacio suficiente para sus conjuntos de trabajo a menudo experimentan hiperpaginación, un fenómeno en el cual se reemplaza continuamente páginas que son devueltas de inmediato al almacenamiento primario.

Reemplazo de páginas por frecuencia de fallas de página

El algoritmo de frecuencia de fallas de página (page fault frequency) ajusta el conjunto de páginas residentes de un proceso basándose en la frecuencia con que el proceso tiene fallas de página o según el tiempo que hay entre fallas de página.

El PFF registra el tiempo entre la última falla de página y el vigente. Si ese tiempo es mayor que un límite superior, entonces se liberan todas las páginas a las que no se haya hecho referencia en ese intervalo. Si el tiempo es menor que un valor límite inferior, la página entrante se convierte en un miembro del conjunto de páginas residentes del proceso.

Paginación por demanda

Es la más popular por:

1. Los resultados de la teoría de la computabilidad indican que no se pueden lograr esquemas de paginación anticipada totalmente preciso, debido a que no se puede predecir el futuro.
2. Se traen al almacenamiento primario solo las páginas que necesita un proceso.

3. El trabajo extra de búsqueda es insignificante.

Liberación de Páginas

Esta ayuda a sacar del almacenamiento primario las páginas que ya no se necesitan.

Tamaño de Páginas

Consideraciones que afectan la determinación del tamaño óptimo de página para un sistema dado:

1. Un tamaño de página pequeño requiere tablas de páginas más grandes con la consecuente fragmentación de tablas.
2. Un tamaño de página grande hace que se transfieran al almacenamiento primario instrucciones y datos a los que no se hará referencia.
3. Las transferencias de entrada/salida son más eficientes con páginas grandes.
4. La localidad tiende a ser pequeña.
5. La fragmentación interna se reduce con páginas pequeñas.

Los diseñadores sugieren el uso de páginas pequeñas.

Muchos experimentos han examinado el comportamiento de los sistemas de computo con paginación y los resultados han sido interesantes:

1. Cuando un proceso comienza a ejecutarse, por lo regular hace referencia rápidamente a un gran porcentaje de sus páginas.
2. El número de fallas experimentadas por un proceso en ejecución tiende a crecer a medida que aumenta el tamaño de la página suponiendo que permanece constante el almacenamiento primario asignado al proceso.
3. El tiempo entre fallos experimentado por un proceso en ejecución crece a medida que aumenta la cantidad de marcos de página asignados al proceso, una vez asignado al proceso un número suficiente de marcos de páginas para contener su conjunto de trabajo disminuye la tasa de incremento.
4. El número de instrucciones ejecutadas en una página antes de que el control salga de la página tiende a ser pequeño.

Unidad V

SISTEMAS DE ARCHIVO

Objetivo de esta unidad.- Durante el desarrollo de esta unidad, se analizará como almacenar una gran cantidad de información, que no se pierda hasta cuando el proceso que la generó termina su ejecución y que dos o más procesos tengan acceso concurrente a la información. El alumno comprenderá la forma de uso de los archivos y las propiedades que estos poseen; la utilización de los directorios, su organización, sus propiedades y las operaciones que se lleven a cabo con ellos; la implantación de un sistema de archivo, la seguridad de la información y mecanismos de protección debe de tener un sistema de archivos.

◆ **Archivo.**-Es un conjunto de datos al cual se le asigna un nombre y reside, por lo regular, en el almacenamiento secundario ya sea en cinta ó en disco, se puede manejar como una unidad mediante operaciones como:

- abrir
- cerrar
- destruir
- copiar
- renombrar
- listar

A los elementos individuales de información dentro del archivo se le pueden aplicar las siguientes operaciones:

- leer
- escribir
- modificar
- insertar
- eliminar

Los archivos pueden caracterizarse por su:

- ◆ **Volatilidad.**- Se refiere a la frecuencia con la que se añade información a un archivo y se borra de él.
- ◆ **Actividad.**- Se refiere al porcentaje de los registros de un archivo al cual se tuvo acceso durante un periodo dado.

- ◆ **Tamaño.-** Se refiere a la cantidad de información almacenada en el archivo.

Sistemas de Archivos

Los sistemas de archivo contienen:

- ◆ **Métodos de acceso.-** Se ocupan de la forma en que se obtiene acceso a los datos almacenados en archivos.
- ◆ **Administración de archivos.-** Se ocupa de ofrecer mecanismos para almacenar, compartir y asegurar archivos, y hacer referencia a ellos.
- ◆ **Administración de almacenamiento secundario.-** Se ocupa de asignar espacio a los archivos en dispositivos de almacenamiento secundario.
- ◆ **Mecanismos de integridad de los archivos.-** Se ocupan de garantizar que no se corrompa la información en un archivo. Cuando se asegura la integridad de los archivos, cualquier información que deba estar en un archivo estará ahí.

El sistema de archivos se ocupa primordialmente de administrar el espacio de almacenamiento secundario, sobre todo el espacio en disco.

Funciones del Sistemas de Archivos

Además de que los usuarios deben poder crear, modificar y eliminar archivos las funciones normalmente atribuidas a los sistemas de archivos son:

- ◆ **Compartir información.-** Los usuarios deben ser capaces de compartir sus archivos entre sí en forma cuidadosamente controlada para aprovechar y continuar el trabajo de los demás.
- ◆ **Mantener privada la información, cifrarla o descifrarla.-** En ambientes delicados en los cuales la información debe mantenerse segura y privada, como los sistemas de transferencia de fondos, sistemas de expedientes criminales, sistemas de expedientes médicos, etc, es conveniente ofrecer estas funciones, lo cual hace que la información solo resulte útil a quienes esta destinada.

- ◆ **Obtener acceso a la información.-** El mecanismo para compartir debe ofrecer varios tipos de acceso controlado, como acceso para lectura, acceso para escritura, acceso para ejecución o las diversas combinaciones de estos.
- ◆ **Respaldo y recuperación de la información.-** Deben ofrecerse recursos de respaldo y recuperación para evitar la pérdida accidental o la destrucción mal intencionada de información.
- ◆ **Independencia con respecto a los dispositivos.-** Los usuarios deben poder hacer referencia a sus archivos mediante nombres simbólicos en vez de tener que usar nombres de dispositivos físicos.

Lo más importante de todo es que el sistema de archivos debe ofrecer una interfaz amable con el usuario. Debe proporcionar a los usuarios una vista lógica de sus datos y las funciones que puede realizar con ellos, más que una vista física. El usuario no debe preocuparse de los dispositivos específicos en los cuales está almacenada la información, la forma que adoptan los datos en esos dispositivos o los mecanismos físicos para transferir datos de esos dispositivos o hacia ellos.

Jerarquía de los Datos

La estructura de datos está formada por entidades de complejidad creciente: ®

bits → bytes → campos → registros → archivos → bases de datos

División en Bloques y Empleo de Buffer

Registro físico o bloque físico.- Es la unidad de información que se lee realmente de un dispositivo o se graba en él.

Registro lógico o bloque lógico.- Es un conjunto de datos manejado como una unidad desde el punto de vista del usuario.

Registro sin Bloques.- Cuando cada registro físico contiene sólo un registro lógico se dice que el archivo está formado por registros sin bloques.

Registro en bloques.- Cuando cada registro físico puede contener varios registros lógicos se dice que el archivo está formado por registros en bloques.

En un archivo con registros de longitud fija, todos los registros tienen el mismo tamaño; el tamaño de bloque casi siempre es un múltiplo entero de la longitud del registro.

En un archivo con registros de longitud variable, el tamaño del registro puede variar, sin rebasar el tamaño del bloque.

El empleo de Buffer (áreas de almacenamiento temporal) hace posible que el cómputo se efectúe de forma simultánea con la entrada/salida. Se reservan espacios en almacenamiento primario para guardar varios bloques físicos de un archivo a la vez, cada uno de estos espacios se llama buffer. El método más común son los buffers dobles y funciona como siguen (para la salida):

1. Existen dos buffers
2. Los registros generados por un proceso se depositan en el primer buffer hasta que se llena.
3. Se inicia la transferencia del bloque en el primer buffer al almacenamiento secundario.
4. Mientras se realiza esta transferencia, el proceso continúa generando registros que se depositan en el segundo buffer
5. Cuando se llena el segundo buffer y una vez finalizada la transferencia del primer buffer, se inicia la transferencia del segundo buffer.

El proceso sigue generando registros, los cuales se depositan ahora en el primer buffer. Esta alternación de buffers permite que se lleve a cabo la entrada/salida en paralelo con los cálculos de un proceso.

Organización de los Archivos

La organización de los archivos se refiere a la forma como se acomodan los registros de un archivo en almacenamiento secundario.

Sistemas de Organización de Archivos más comunes:

- ◆ **Secuencial.-** Los registros se colocan en orden físico. Es una organización natural para archivos grabados en cinta magnética, medio de almacenamiento que por su naturaleza es secuencial. Los archivos en disco también se pueden organizar secuencialmente, aunque por diversas razones, los registros de un archivo secuencial de disco no se almacenan en forma contigua por fuerza.
- ◆ **Directo.-** Se obtiene acceso directo (aleatorio) a los registros por su dirección física en un dispositivo de almacenamiento de acceso directo (DASD, direct access storage device)
- ◆ **Secuencial indexado.-** Los registros se acomodan en secuencia lógica de acuerdo con una clave contenida en cada registro. El sistema mantiene un índice con las direcciones físicas de ciertos registros principales. El acceso a los registros secuencial indexado puede obtenerse secuencialmente por orden de clave o de manera directa, mediante una búsqueda dentro del índice creado por el sistema. Los archivos secuencial indexado casi siempre se almacenan en discos.
- ◆ **De partición.-** Este es en esencia un archivo de subarchivos secuenciales. Cada subarchivo secuencial se llama miembro. La dirección inicial de cada miembro se almacena en el directorio del archivo.

El término volumen se usa para referirse al medio de grabación de cada dispositivo auxiliar de almacenamiento en particular. El volumen empleado en una unidad de cinta es el carrete de cinta y el volumen empleado en una unidad de disco es el disco.

Métodos de acceso

El acceso a los archivos se logra mediante funciones de los sistemas operativos llamados métodos de acceso, estos se agrupan en dos categorías:

- ◆ **Método de acceso por colas.-** Se usan con archivos de organización secuencial y secuencial indexado; realizan transferencias anticipadas a

buffers y programan la E/S además de ofrecer agrupación en bloques y separación de bloques en forma automática.

- ◆ **Métodos de acceso básicos.-** Se utilizan por lo regular cuando no es posible anticipar el orden en el que se van a procesar los registros, sobre todo en el caso del acceso directo. En los métodos básicos, solo es posible leer y escribir bloques físicos; si se requiere en la aplicación, el usuario se encarga de la formación y división de bloques.

Asignación y Liberación de Espacio

Con la asignación contigua cada archivo se asigna a un área única del almacenamiento secundario. La asignación contigua facilita el acceso rápido pero adolece de serios problemas de fragmentación.

Si la asignación es no contigua, el archivo puede estar disperso en varias áreas de almacenamiento secundario. La asignación no contigua es más flexible pero puede requerir búsquedas frecuentes.

Descriptor de Archivos

Llamado también Bloque de Control de Archivo. Es un bloque de control con información que el sistema necesita para administrar un archivo. Un descriptor de archivo representativo incluye lo siguiente:

1. Nombre simbólico del archivo.
2. Localización del archivo en el almacenamiento secundario.
3. Organización del archivo.
4. Tipo de dispositivo
5. Datos para el control de acceso.
6. Tipo (Sí es archivo de datos, programa objeto, o programa fuente).
7. Tratamiento (Temporal o permanente).
8. Fecha y hora de creación.
9. Fecha de destrucción.
10. Fecha y hora de la última modificación.

11. Conteo de la actividad de acceso.

Los descriptors de archivo se mantienen en el almacenamiento secundario y se transfieren al almacenamiento primario cuando se abre un archivo. El descriptor de archivo es controlado por el sistema operativo, el usuario no puede hacer referencia directa a él.

Tipos de Control de Acceso

El control de acceso a los archivos puede manejarse mediante una matriz para control de acceso que indica cuáles usuarios tienen, qué tipos de acceso a cuáles archivos. Lo más frecuente es que el control de acceso se maneje por clases de usuarios donde por ejemplo se puede conceder acceso al propietario ó a un usuario específico, a un miembro de un grupo ó a un miembro del público en general.

Respaldo y Recuperación

Son funciones muy importantes de cualquier sistema de archivos. La técnica más común es el respaldo periódico. Otra técnica es el vaciado por incrementos.

Vaciado por Incrementos.- Los archivos modificados por un usuario durante una sesión determinada se respaldan cuando ese usuario sale del sistema.

Bitácora de Transacciones.- Todas las líneas tecleadas por un usuario se copian en un archivo de bitácora. La recuperación implicaría literalmente una nueva aplicación de todas las transacciones realizadas después del último respaldo periódico principal.

Bases de Datos

Es un conjunto integrado de datos controlados centralmente. Algunas de las ventajas principales son:

1. Se puede reducir la redundancia.
2. Evitar la inconsistencia.

3. Compartir datos.
4. Imponer Normas.
5. Se puede aplicar restricciones de seguridad.
6. Mantener la integridad.
7. Se puede equilibrar requerimientos en conflicto.

Independencia de los datos.- Hace posible modificar una aplicación y desarrollar nuevas aplicaciones sin tener que alterar la estructura del almacenamiento de los datos y la estrategia de acceso.

Bases de datos distribuida.- Está distribuida o dispersa en todos los sistemas de cómputo mediante una red.

Tipos de organización de bases de datos

1. Jerárquica
2. Red
3. Relacional

En el enfoque jerárquico los datos se organizan según interrelaciones padre-hijo, cada hijo tiene un solo padre y cada padre puede tener muchos hijos.

Búsquedas y mantenimientos fáciles, pero es limitada la flexibilidad del usuario para diseñar interdependencias complejas de los datos.

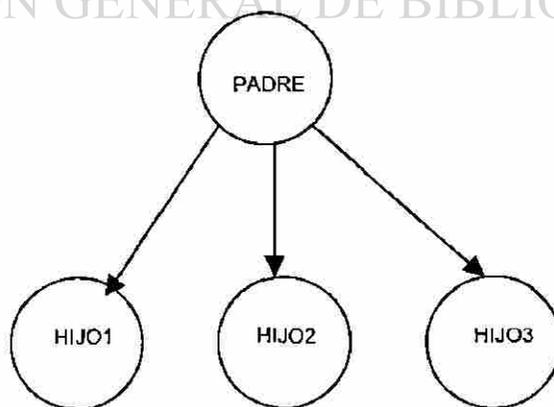


Fig. 5.1. Enfoque jerárquico

En el enfoque de red se pueden expresar en forma conveniente interdependencias muy generales, pero las estructuras resultantes pueden ser difíciles de entender, modificar o reconstruir en caso de falla (Red).

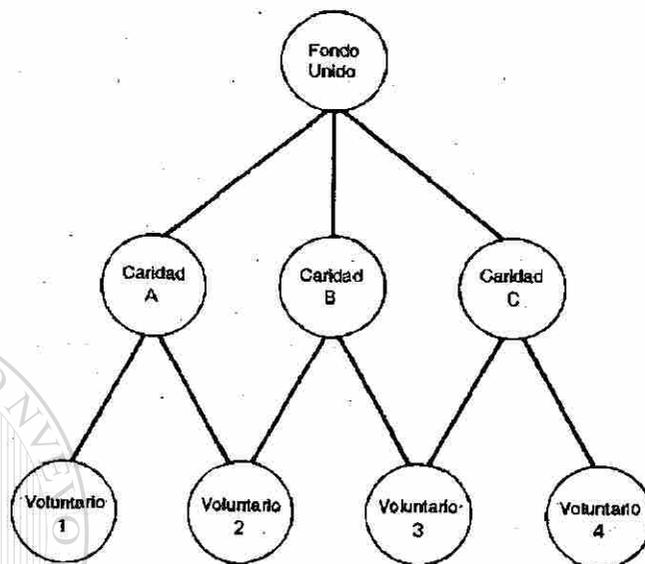


Fig.5.2. Enfoque de red

El enfoque relacional ofrece muchas ventajas con respecto a las anteriores, la representación tabular es más fácil de comprender y llevar a la práctica. Otros esquemas se pueden convertir con mucha facilidad a la organización relacional.

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

Relación: EMPLEADO

Número	Nombre	Departamento	Salario	Localización
23603	RAMOS, A.	413	1100	SEGOVIA
24568	CORTÉS, R.	413	2000	SEGOVIA
34589	LÓPEZ, P.	642	1800	ZARAGOZA
35761	MENDOZA, B.	611	1400	MADRID
47132	NOGUERA, C.	413	9000	SEGOVIA
78321	SALAS, T.	611	8500	MADRID

Una tupla { 34589 LÓPEZ, P. 642 1800 ZARAGOZA }

Clave primaria { 34589 LÓPEZ, P. 642 1800 ZARAGOZA }

Un dominio { 642 1800 ZARAGOZA }

Fig. 5.3 Enfoque relacional

Operaciones útiles como la proyección y la reunión facilitan la creación de nuevas relaciones. Los datos delicados pueden asegurarse colocándolos en relaciones separadas. Las búsquedas son más directas y rápidas, modificaciones directas y mejora la claridad y visibilidad de los datos.

SEGURIDAD EN LOS SISTEMAS OPERATIVOS

La seguridad y el compartimiento son objetivos en conflicto.

- ◆ **Seguridad externa.**- Se ocupa de proteger el sistema de computo de intrusos y desastres como incendios e inundaciones, etc.
- ◆ **Seguridad de interfaz con el usuario.**- Se ocupa de establecer la identidad de un usuario antes de que se le conceda el acceso a un sistema.
- ◆ **Seguridad interna.**- Se ocupa de garantizar el funcionamiento confiable y sin corrupción del sistema de computo y la integridad de los programas y datos.
- ◆ **Autorización.**- Determina qué acceso se permite a qué entidades.
- ◆ **División de responsabilidades.**- Asigna al personal subconjuntos distintos de deberes; ningún empleado se encarga de una porción grande de la operación de un sistema, por lo que un ataque a la seguridad tendría que implicar a varios empleados.
- ◆ **Vigilancia.**- Se ocupa de supervisar el sistema y realizar auditorías así como de verificar la autenticidad de los usuarios.
- ◆ **Supervisión de amenazas.**- El sistema operativo controla operaciones delicadas en lugar de ceder el control directamente a los usuarios. Los programas de vigilancia ejecutan las operaciones delicadas. Se habla de amplificación cuando éstos programas requieren un acceso más amplio para atender las solicitudes de los usuarios.

Protección por contraseña

Hay tres clases de elementos de la verificación de autenticidad con los cuales puede establecerse la identidad de una persona:

1. Algo característico de la persona (huellas digitales, patrones de voz, fotografías y firmas)
2. Algo que la persona posee (credenciales, tarjetas de identificación y claves).
3. Algo que sabe la persona (contraseñas, combinaciones de candados, el apellido de su maestra de tercer año de primaria).

El ciframiento de la lista maestra de contraseñas ayuda a mantener la seguridad de estas aun cuando haya penetración al sistema. Se recomienda cambiar con frecuencia las contraseñas.

- ◆ **Auditoría.-** Se realiza por lo general en sistemas manuales a posteriori. Se convocan auditores periódicamente para examinar las transacciones recientes de una organización y determinar si se han realizado actividades fraudulentas. En los sistemas de computo puede implicar un procesamiento inmediato en el computador para revisar las transacciones que se acaban de realizar.
- ◆ **Bitácora de auditoría.-** Es un registro permanente de eventos importantes que ocurren en el sistema de computo. Se produce automáticamente cada vez que sucede un evento así y se almacena en un área protegida del sistema; si el sistema se ve comprometido, la bitácora deberá permanecer intacta. Es un mecanismo de detección importante. Aunque logren penetrar las defensas de un sistema, las personas pueden refrenar sus deseos de hacerlo si temen una detección posterior.
- ◆ **Controles de acceso.-** La clave para la seguridad interna es controlar el acceso a los datos almacenados. Los derechos de acceso definen qué acceso tienen varios sujetos a diversos objetos. Los objetos se protegen contra los sujetos. Los accesos más comunes son los de lectura, escritura y ejecución.
- ◆ **Núcleos de seguridad.-** Las medidas de seguridad más vitales se ponen en práctica en el núcleo, el cual se mantiene a propósito lo más pequeño

posible. Esto hace más razonable la revisión cuidadosa del núcleo para detectar fallas y demostrar formalmente que este correcto.

La seguridad de un sistema operativo depende sobre todo de asegurar las funciones que se encargan del control de acceso, las entradas al sistema, la supervisión, funciones que administran el almacenamiento real, el almacenamiento virtual y el sistema de archivos.

- ◆ **Sistemas tolerantes a fallas.-** Un sistema de cómputo tolerante a fallas continua funcionando aún después de haber fallado uno o más de sus componentes. La tolerancia a fallas se facilita mediante la incorporación de mecanismos a prueba de fallas, el empleo de multiprocesamiento transparente, el uso de múltiples subsistemas de E/S, la incorporación en hardware de gran parte del sistema operativo y la incorporación en hardware de mecanismos para la detección de fallas.

Capacidades y Sistemas Orientados a Objetos

Un derecho de acceso permite a algún sujeto obtener acceso a un objeto de una manera predeterminada. Los sujetos son usuarios de sistemas de cómputo o entidades que actúan a nombre de los usuarios o del sistema. Los objetos son recursos dentro del sistema. Los sujetos pueden ser cosas como tareas, procesos y procedimientos. Los objetos pueden ser archivos, programas, semáforos, directorios, terminales, canales, controladores, dispositivos, pistas de disco, bloques de almacenamiento primario, etc. Los sujetos también se consideran como objetos del sistema, de modo que un sujeto puede tener acceso a otro. Los sujetos son entidades activas; los objetos son pasivos.

Un dominio de protección define los derechos de acceso que tiene un sujeto a los diversos objetos del sistema. Es el conjunto de capacidades pertenecientes a un sujeto. Es importante que los dominios de protección sean pequeños para hacer cumplir el principio de menor privilegio. Para que un sujeto obtenga acceso a un objeto específico debe poseer una capacidad para ello. El problema del objeto perdido se refiere, precisamente, a lo que sucede cuando

se elimina la capacidad para tener acceso a un objeto. La renovación de capacidades puede ser difícil, una capacidad podría haber sido copiadas muchas veces. Las capacidades por lo regular no se modifican, pero pueden reproducirse.

Criptografía

- ◆ **Criptografía.-** Es el empleo de transformaciones de los datos a fin de hacerlos incomprensibles para todos con excepción de sus usuarios autorizados.
- ◆ **Problema de intimidad.-** Se ocupa de evitar la extracción no autorizada de información de un canal de comunicación.
- ◆ **Problema de verificación de autenticidad.-** Se ocupa de evitar que algún enemigo modifique una transmisión o inserte datos falsos en una transmisión.
- ◆ **Problemas de disputa.-** Se ocupa de ofrecer al receptor de un mensaje una prueba legal de la identidad del remitente o sea el equivalente electrónico de una firma escrita. Como funciona: un remitente cifra texto simple para crear texto cifrado el cual se transmite a un receptor a través de un canal no seguro incluso vigilado por un espía. El receptor descifra el texto cifrado para reconstruir el texto simple original.
- ◆ **Criptoanálisis.-** Es el proceso de intentar regenerar un texto simple a partir del texto cifrado pero sin conocer la clave de desciframiento.
- ◆ **Clave pública.-** En los sistemas de clave pública las funciones de ciframiento y desciframiento están separadas, cada una requiere una clave distinta, la clave se hace pública si la otra permanece en secreto. Cuando se cifra un mensaje con clave pública de un usuario sólo ese usuario puede descifrar el mensaje. Con los sistemas de clave pública es posible llevar a la práctica firmas digitales que garanticen la autenticidad de un mensaje.

Los esquemas DES y RSA

Dos de los esquemas criptográficos más importantes son la norma de ciframiento de datos (DES) y el esquema Rivest, Shamir y Adleman (RSA).

DES es un esquema simétrico de ciframiento en el cual se usa una sola clave para cifrar y descifrar y RSA es un esquema asimétrico en el cual se utilizan claves distintas para estos propósitos.

El uso más común en los sistemas operativos actuales es para proteger la lista maestra de contraseñas de un sistema. El ciframiento también es usado para proteger datos almacenados en archivos y para proteger datos que se transmiten a través de una red. Las cintas y discos de respaldo cifrado no necesitan cuidarse con tanto celo como los no cifrados. El ciframiento de enlaces se ocupa del ciframiento/desciframiento en cada nodo de una red de computadores.

Ejemplo: Si se usa ciframiento de extremo a extremo los mensajes se cifran sólo en su punto de origen y se descifran sólo en su punto destino.

Mediante un procedimiento de reto y respuesta, un sistema puede verificar la autenticidad de un usuario cuando trate de entrar, sin necesidad de transmitir una clave.

Penetración en el Sistema Operativo

Las defensas de un sistema operativo deben ser capaces de resistir un intento de penetración por parte de un usuario no privilegiado; hacer que un sistema operativo sea impenetrable es una tarea imposible, lo que podemos esperar es que sea altamente resistente a la penetración.

Defectos Funcionales Genéricos de los Sistemas

Se han encontrado varios defectos comunes a muchos sistemas de computo. Entre ellos están:

- ◆ **Verificación de autenticidad.-** En muchos sistemas, los usuarios no pueden determinar si el equipo y los programas son lo que deberían de ser. Esto hace que un penetrador pueda reemplazar con facilidad un programa sin que se entere el usuario. Ejemplo.- Un usuario podría dar su contraseña a un programa falso de entrada al sistema.

- ◆ **El ciframiento.-** La lista maestra de contraseñas debe almacenarse en forma cifrada. A veces no se hace.
- ◆ **Realización.-** Un diseño bien pensado para un mecanismo de seguridad puede llevarse a la práctica en forma inadecuada.
- ◆ **Confianza implícita.-** Una rutina supone que otra está funcionando correctamente, en vez de examinar con cuidado los parámetros suministrados por la otra.
- ◆ **Compartimiento implícito.-** El sistema puede depositar sin darse cuenta información vital del sistema en el espacio de direcciones de un usuario.
- ◆ **Comunicación entre procesos.-** El penetrador puede usar un mecanismo de transmisión/recepción para probar diversas posibilidades. Ejemplo.- El penetrador puede solicitar un recurso del sistema y suministrar una contraseña; la información de vuelta puede indicar “contraseña correcta”, confirmando la contraseña adivinada por el penetrador.
- ◆ **Comprobación de legalidad.-** El sistema quizá no verifique lo suficiente la validez de los parámetros del usuario.
- ◆ **Desconexión de línea.-** En sistemas de tiempo compartido y redes, cuando se pierde la línea el sistema operativo deberá clausurar de inmediato la sesión del usuario o poner a éste en un estado tal que sea necesaria una nueva autorización para otorgarle el control. Un penetrador podría obtener control del proceso y utilizar los recursos a los cuales puede tener acceso este último.
- ◆ **Descuido del operador.-** Un penetrador puede engañar a un operador para que monte un disco de sistema operativo falso.
- ◆ **Paso de parámetros por referencia en vez de por valor.-** Es más seguro pasar parámetros directamente en registros y no hacer que los registros apunten a localidades donde están los parámetros. El paso por referencia puede conducir a una situación en la cual los parámetros siguen en el espacio de direcciones del usuario después de haberse realizado la verificación de legalidad; así, el usuario podría suministrar parámetros

legítimos, hacer que sean verificados y después modificarlos justo antes de que los utilice el sistema.

- ◆ **Contraseñas.-** A menudo las contraseñas son fáciles de adivinar o de obtener por intentos repetidos.
- ◆ **Trampas para el penetrador.-** Los sistemas deben incluir mecanismos de trampa para atraer al intruso inexperto, pues constituyen una buena primera línea de detección. La mayor parte de los sistemas tienen mecanismos de trampa inadecuados.
- ◆ **Privilegios.-** En algunos sistemas, son demasiados los programas que tienen demasiados privilegios. Esto va contra el principio del menor privilegio.
 - ◆ **Confinamiento de programas.-** Un programa prestado por otro usuario puede actuar como Caballo de Troya; podría robar o alterar los archivos de quien lo pidió prestado.
 - ◆ **Prohibición.-** Muchas veces se indica a los usuarios que se abstengan de usar ciertas funciones porque los resultados pueden ser “indeterminados”. No obstante, estas funciones siguen siendo accesibles para los usuarios.
 - ◆ **Residuo.-** El penetrador puede encontrar una lista de contraseñas examinando el cesto de la basura. En ocasiones se deja residuo en el almacenamiento después de ejecutarse una rutina del sistema. Toda información confidencial deberá reemplazarse o destruirse antes de liberar o desechar el medio que ocupa.
- ◆ **Blindaje.-** Un alambre con corriente en un alambre genera un campo magnético alrededor de este; los penetradores pueden intervenir de hecho una línea de transmisión o un sistema de computo sin hacer contacto físico. El blindaje eléctrico puede servir para evitar estas “intrusiones invisibles”.
- ◆ **Valores de umbral.-** El propósito de estos es refrenar intentos repetidos de entrar en el sistema. Ejemplo: Después de un cierto número de intentos de entrada no válidos, ese usuario deberá bloquearse, notificando al

administrador del sistema. Muchos sistemas no cuentan con esta característica.

Ataques Genéricos a los Sistemas Operativos

Metodologías de penetración:

- ◆ **Asincronía.-** Cuando varios procesos avanzan en forma asíncrona, es posible que un proceso modifique parámetros cuya validez ha sido verificada por otro, aunque este último todavía no los haya usado; así, un proceso puede pasar valores malos a otro aunque el segundo realice una verificación exhaustiva.
- ◆ **Hojeo.-** Un usuario revisa el sistema de computo intentando localizar información privilegiada.
- ◆ **Entre líneas.-** Se usa una terminal especial para intervenir una línea de comunicaciones empleada por un usuario inactivo que ya haya entrado en el sistema.
- ◆ **Código clandestino.-** Se instala un parche con la pretensión de corregir un error en el sistema operativo; el código contiene escotillones, a través de los cuales se puede entrar después en el sistema sin autorización.
- ◆ **Rechazo de acceso.-** Un usuario escribe un programa para hacer que se caiga el sistema, para ponerlo en un ciclo infinito o para monopolizar sus recursos. La intención en este caso es impedir que usuarios legítimos obtengan acceso o servicio.
- ◆ **Interacción de procesos sincronizados.-** Los procesos usan las primitivas de sincronización del sistema para compartir o pasar información entre ellos.
- ◆ **Desconexión de línea.-** El penetrador intenta obtener acceso al trabajo de un usuario después de una desconexión de línea, pero antes de que el sistema reconozca la desconexión.
- ◆ **Disfraz.-** El penetrador asume la identidad de un usuario legítimo después de haber obtenido la identificación correcta por medios clandestinos.

- ◆ **Ataque NAK.-** Muchos sistemas permiten a un usuario interrumpir un proceso en ejecución(utilizando la tecla “negative acknowledge”), realizar otra operación y después continuar el proceso interrumpido. El penetrador puede “atrapar” al sistema en un estado no protegido y adueñarse del control con facilidad.
- ◆ **Engaño del operador.-** Un penetrador astuto a menudo puede engañar al operador del computador para que realice una acción que ponga en peligro la seguridad del sistema.
- ◆ **Parásito.-** El penetrador utiliza una terminal especial para intervenir una línea de comunicación. El penetrador intercepta mensajes entre el usuario y el procesador y los modifica o bien los reemplaza por completo.
- ◆ **Caballo de Troya.-** El penetrador coloca código en el sistema que le permitirá un acceso posterior no autorizado. El Caballo de Troya puede dejarse en el sistema permanentemente o puede borrar todos los indicios de su existencia después de una penetración.
- ◆ **Parámetros inesperados.-** El penetrador suministra valores inesperados en una llamada al supervisor para aprovechar un punto débil en los mecanismos de verificación de legalidad del sistema.

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

La confidencialidad de un computador no existe.

Aquí la idea es:

Si no quieres que lean algo

No lo metas en el sistema.

Unidad VI

ENTRADA / SALIDA

Objetivo de esta unidad.- Durante el desarrollo de esta unidad, se analizará la forma de controlar los dispositivos de entrada/salida de la computadora.

El alumno deberá comprender los principios de hardware de E/S y de software de E/S.

Dispositivos de E/S

De manera general se dividen en dos categorías:

- ◆ **Dispositivo de bloque.-** La propiedad esencial de un dispositivo de bloque es la posibilidad de leer o escribir en un bloque de forma independiente de los demás, es decir, en todo momento, el programa puede leer o escribir en cualquiera de los bloques. Los discos son dispositivos de bloque.
- ◆ **Dispositivos de caracter.-** Un dispositivo de caracter envía o recibe un flujo de caracteres, sin sujetarse a una estructura de bloques. No se pueden utilizar direcciones ni tienen operación de búsqueda. Las terminales, impresoras de línea, cintas de papel, tarjetas perforadas, interfaces de red, mouse y otros dispositivos no parecidos a los discos son dispositivos de caracter.

Controladores de dispositivos.

Las unidades de E/S cuentan por lo general con un componente mecánico y un componente electrónico. El componente electrónico se llama controlador de dispositivo o adaptador, este toma con frecuencia la forma de tarjeta de circuitos impresos que se puede insertar en la computadora. El componente mecánico es el dispositivo mismo.

La labor del controlador es convertir el flujo de bits en serie en un bloque de bytes y llevar a cabo cualquier corrección de errores necesaria. Lo común es que el bloque de bytes se ensamble, bit a bit, en un buffer dentro del controlador. Después de verificar la suma y declarar al bloque libre de errores, se le puede copiar en la memoria principal.

El controlador de una terminal CRT también funciona como un dispositivo serial de bits en un nivel igual de bajo. Lee bytes que contienen los caracteres a exhibir en la memoria y genera las señales utilizadas para modular la luz CRT para que esta escriba en pantalla. El controlador también genera señales para que la luz CRT vuelva a realizar un trazo horizontal después de terminar una línea de rastreo, así como las señales para que vuelva a hacer un trazo vertical después de rastrear en toda la pantalla. De no ser por el controlador CRT, el programador del sistema operativo tendría que programar en forma explícita el rastreo análogo del tubo de rayos catódicos.

Con el controlador el sistema operativo inicializa éste con pocos parámetros, tales como el número de caracteres por línea y el número de líneas en la pantalla, para dejar que el controlador se encargue de dirigir en realidad el rayo de luz. Cada controlador tiene unos cuantos registros que utiliza para la comunicación con el CPU. En ciertas computadoras, estos registros son parte del espacio normal de direcciones de la memoria. Este esquema se llama E/S mapeada a memoria. El manejador de disco es la única parte del sistema operativo que conoce el número de registros de un controlador de disco y el uso de que tienen estos. El se encarga de los sectores, pistas, cilindros, cabezas, movimiento del brazo, factores de separación, control de motor, tiempos de descenso de la cabeza y el resto de la mecánica del funcionamiento adecuado del disco.

Acceso Directo a Memoria (DMA)

Muchos controladores, en particular los correspondientes a los dispositivos de bloques, permiten el acceso directo a memoria, si no se utilizara éste, la lectura del disco se haría de la siguiente forma:

1. El controlador lee en serie el bloque de la unidad, bit por bit, hasta que todo el bloque se encuentra en el buffer interno del controlador.
2. Se calcula la suma de verificación para corroborar que no existen errores de lectura.

3. El controlador provoca una interrupción.
4. Cuando el sistema operativo empieza su ejecución, puede leer el bloque del disco por medio del buffer del controlador, un byte o una palabra a la vez, en un ciclo, en el que durante cada iteración se lee un byte o una palabra del registro del controlador y se almacena en memoria.

Un ciclo programado en el CPU para la lectura de bytes desde el controlador (uno a la vez) desperdicia tiempo de CPU. Al utilizar DMA, el CPU le proporciona al controlador, dos elementos de la información, además de la dirección del bloque en el disco, la dirección en memoria donde debe ir el bloque y el número de bytes por transferir. Después de que el controlador ha leído todo el bloque del dispositivo a su buffer y ha corroborado la suma de verificación, copia el primer byte o palabra a la memoria principal, en la dirección especificada por medio de la dirección de memoria DMA. Entonces incrementa la dirección DMA y decrementa el contador DMA en el número de bytes que acaba de transferir. Este proceso se repite hasta que el contador se anula, momento en el cual el controlador provoca una interrupción. Al iniciar su ejecución el sistema operativo, no tiene que copiar el bloque en la memoria, ya está ahí.

Interfaz uniforme para los manejadores de dispositivos
Nombres de los dispositivos
Protección del dispositivo
Proporcionar un tamaño de bloque independiente del dispositivo
Uso de buffers
Asignación de espacio en los dispositivos por bloques
Asignación y liberación de los dispositivos de uso exclusivo
Informe de errores

Funciones del software de E/S independiente del dispositivo

DISCOS

El uso de discos para almacenar información tiene tres ventajas con respecto al uso de la memoria principal como almacenamiento:

1. La capacidad de espacio de almacenamiento disponible es mucho más grande.
2. El precio por bit es más barato.
3. La información no se pierde al apagar la computadora.

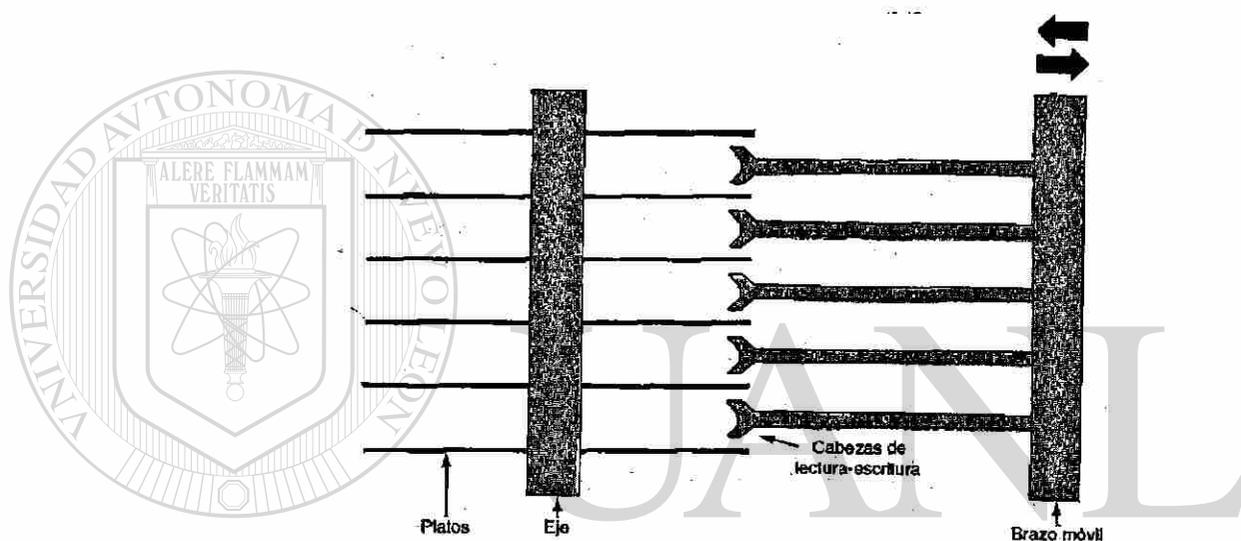


Fig. 6.1. Vista lateral de un disco de cabeza móvil

Operación de un Almacenamiento de Disco de Cabeza Móvil. ®

Los datos se graban sobre una serie de discos magnéticos o platos. Estos discos están conectados por un eje común que gira a una velocidad muy alta (algunos ejes alcanzan a girar a 3600 revoluciones por minuto).

El acceso a los datos es mediante una serie de cabezas de lectura-escritura. Una cabeza de lectura-escritura sólo puede tener acceso a los datos que estén adyacentes a ella. De este modo, antes de que pueda obtenerse acceso a los datos, la porción de la superficie del disco de la que se leerán los datos (ó se escribirán) debe girar hasta que se encuentre justo abajo (ó arriba) de la cabeza de lectura-escritura.

El tiempo que le toma a los datos girar desde la posición en que se encuentran hasta una adyacente a la cabeza de lectura-escritura se le llama: Tiempo de Latencia.

Cada una de las diferentes cabezas de lectura-escritura, mientras están fijas en una posición, determinan una pista circular de datos sobre la superficie de un disco.

Todas éstas cabezas están sujetas a una sola unidad de brazo móvil la cual puede moverse hacia adentro ó hacia afuera. Cuando el brazo móvil desplaza las cabezas hacia una nueva posición puede obtenerse acceso a otro conjunto de pistas.

Para una posición dada del brazo móvil, el conjunto de pistas definido por todas las cabezas forma un cilindro vertical.

El proceso de desplazar el brazo móvil hacia un nuevo cilindro se conoce como operación de búsqueda (tiempo de búsqueda).

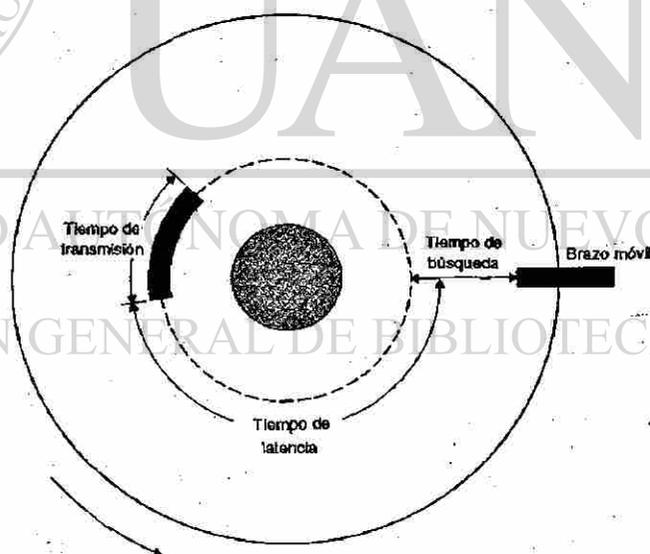


Fig. 6.2. Disco visto desde arriba

Para obtener acceso a un registro de datos en particular en un disco de cabeza móvil se hacen el siguiente proceso:

1. El brazo móvil debe desplazarse hacia el cilindro apropiado.

2. La porción del disco en donde se encuentran los datos debe girar hasta quedar inmediatamente abajo (ó arriba) de la cabeza de lectura-escritura (Tiempo de Latencia).
3. El registro, cuyo tamaño es arbitrario (el máximo tamaño es una pista) debe girar para pasar por la cabeza (Tiempo de transmisión).

En síntesis, un acceso a disco (lectura o escritura) implicará tres acciones importantes:

1. Una búsqueda
2. Un retraso rotacional (latencia)
3. Una transmisión de registros

Cabe señalar que cada operación implica movimientos mecánicos, teniendo como resultado un tiempo total de acceso de una fracción de segundo (0.01 a 0.1 segundos). Estas velocidades pudieran parecer lentas si las comparamos con las velocidades de procesamiento.

En los sistemas multiprogramados, muchos procesos pueden estar haciendo solicitudes de lectura y escritura de registros en discos. En ocasiones estos procesos realizan peticiones más rápido de lo que pueden ser atendidas por los discos de cabeza móvil, se forman colas de espera para cada dispositivo. Por

lo general algunos sistemas se limitan a atender las peticiones según el esquema FCFS (First-Come-First-Served), primero que llega, primero que se atiende, que es un método justo para asignar servicio, lamentablemente, cuando aumenta la carga, puede dar lugar a tiempos de espera muy largos. Además, exhibe un patrón de búsqueda aleatorio, provocando búsquedas de los cilindros más internos a los más externos, lo que hace que se consuma mucho tiempo.

La ineficiencia se debe al uso inapropiado de recursos de almacenamiento rotacional como discos y tambores.

- ◆ **Planificación de disco.**- El acomodo de solicitudes pendientes de acceso a disco para reducir las búsquedas se le conoce como planificación de disco e

implica un examen cuidadoso de las peticiones pendientes para determinar la forma más eficiente de atenderlas.

Un planificador de disco examina las relaciones de posición entre las peticiones en espera. La cola de espera se reordena para que las peticiones puedan atenderse con un mínimo de movimiento mecánico.

Tipos de planificación de Disco

- ◆ Optimización de la Búsqueda
- ◆ Optimización Rotacional (o de latencia)

Debido a que los tiempos de búsqueda tienen un orden de magnitud más grande que los tiempos de latencia, la mayor parte de los algoritmos de planificación se concentran en reducir al mínimo los tiempos de búsqueda para un conjunto de peticiones, la reducción de los tiempos de latencia por lo general tiene poco efecto sobre el rendimiento global del sistema, excepto claro, bajo cargas pesadas.

En la búsqueda de optimizar el rendimiento del disco, algunas veces es preciso retardar el procesamiento del disco. Al instalar un disco duro se pueden obtener tasas de transferencia de disco mayores que las que puede manejar un computador personal, así que los datos excedentes se deben almacenar temporalmente en el controlador de disco. La alternación puede ser útil en estos casos. Para retardar la tasa efectiva de transferencia, los registros consecutivos de un archivo secuencial están separados por $n-1$ bloques de disco a fin de dar oportunidad al procesador de alcanzar el disco. Esto da como resultado lo que se conoce como alternación de disco de n vías. Pero las velocidades de transferencia de disco son una cuestión independiente de la reducción de número de búsquedas mínimas; estos últimos aspectos siguen dominando las consideraciones de optimización del acceso a disco.

Características Deseables de las Políticas de Planificación de Disco

Algunos criterios que se siguen para clasificar las políticas de planificación son:

- ◆ **La productividad.**- El mayor número posible de peticiones atendidas por unidad de tiempo.
- ◆ **El tiempo promedio de respuesta.**- Tratar de reducir el tiempo promedio de respuesta, es decir, el tiempo promedio de espera más el tiempo promedio de servicio. Como, la planificación reduce el tiempo de espera por búsquedas, debe ser ciertamente capaz de mejorar el tiempo promedio de respuesta de FCFS.

La planificación mejora a menudo el rendimiento global pero reduce el nivel de atención para ciertas peticiones.

- ◆ **La varianza de los tiempos de respuesta (predecibilidad).**- La varianza es una medida estadística de cuanto se desvían elementos individuales del promedio de los elementos. Utilizamos la varianza para indicar la predecibilidad, esto es, a menor varianza mayor predecibilidad.

Si una política de planificación sólo trata de aumentar la producción sin reducir al mínimo la varianza, podría procesar nada más las peticiones fáciles y hacer caso omiso de las difíciles.

Optimización de la Búsqueda

- ◆ **SSTF (Primero en el menor tiempo de búsqueda).**- El brazo del disco se traslada enseguida (en cualquier dirección) a la petición que requiere un movimiento mínimo. Esto es, atiende las solicitudes de acuerdo a su proximidad con la última solicitud atendida. La siguiente solicitud en ser atendida es la más cercana a la última sin tener en cuenta la dirección en que se deba desplazar el brazo móvil. Una desventaja importante es que aumenta la varianza de los tiempos de respuesta debido a la discriminación contra las pistas exteriores e interiores. En un caso extremo, podría causar aplazamiento indefinido de las solicitudes alejadas de las cabezas de lectura-escritura. Sin embargo, resulta útil en sistemas de procesamiento por lotes, donde la productividad es la consideración más importante.

Aunque la elevada varianza de los tiempos de respuesta (impredecibilidad) lo hace inaceptable en sistemas interactivos.

- ◆ **SCAN** .- El brazo del disco se mueve hacia adentro y hacia afuera, atendiendo todas las peticiones que encuentra a su paso, cambia su dirección sólo cuando no hay más peticiones que atender en la dirección actual. Es decir, funciona como SSTF excepto que el brazo se sigue moviendo en una dirección preferida hasta que se atienden todas las solicitudes en esa dirección; después se invierte el proceso. A causa del movimiento oscilante de las cabezas de lectura-escritura las pistas más exteriores se visitan con menos frecuencia que las de la parte media, pero esto no es tan grave como la discriminación de SSTF.

- ◆ **C-SCAN (Scan circular)**.- El brazo se mueve en una sola dirección sobre la superficie del disco hacia la pista más interior, cuando no hay más peticiones en esa dirección regresa para atender la petición más cercana a la pista exterior y de nuevo se mueve hacia adentro. Es decir, elimina la discriminación del SCAN contra las pistas de la parte media realizando el barrido en una sola dirección; al terminar un barrido el brazo móvil salta al extremo opuesto del disco y continúa el barrido en la misma dirección.

Algunos resultados de simulaciones en la literatura indican que la mejor política de planificación de disco podría operar en dos etapas. Cuando la

carga es ligera, la política SCAN es la mejor. Cuando la carga es mediana o pesada, C-SCAN produce los mejores resultados. C-SCAN con optimización rotacional maneja en forma efectiva las situaciones de carga pesada.

- ◆ **SCAN de N pasos**.- El brazo del disco se mueve igual que en Scan, pero las peticiones llegan durante el barrido en una dirección se almacenan y reordenan para darles un servicio óptimo durante el barrido de retorno. Esto es, evita los retrasos y posiblemente un aplazamiento indefinido al obligar a las solicitudes que llegan a que esperen el barrido en la dirección opuesta para ser atendidas. El Scan de n pasos elimina la posibilidad de que ocurra un aplazamiento indefinido si llega un gran número de peticiones para el

cilindro actual. Este guarda dichas peticiones para atenderlas en el barrido de regreso.

- ♦ **Esquema de Eschenbach.**- En cada cilindro se atiende toda una pista de información sin importar si existen o no peticiones para ese cilindro. Las peticiones dentro de un cilindro se reordenan para atenderlas aprovechando su posición rotacional, pero si existen dos peticiones traslapadas dentro de un cilindro se atenderá sólo a una en ese barrido del brazo del disco. Fue diseñado para cargas en extremo pesadas, aunque C-SCAN con optimización rotacional ha demostrado ser más efectivo bajo todas las cargas. Fue uno de los primeros en tratar de optimizar no solo el tiempo de búsqueda, sino también el retraso rotacional.

Optimización Rotacional

En condiciones de carga pesada aumenta la probabilidad de múltiples referencias a un cilindro específico por lo que resulta útil tener en cuenta la optimización rotacional además de la optimización de la búsqueda. Ha sido utilizada por años en dispositivos de cabeza fija como los tambores.

- ♦ **SLTF (shortest-latency-time-first).**-La estrategia de primero el tiempo de latencia mas corto ó para la optimización rotacional es análoga de la estrategia SSTF para la optimización de la búsqueda. Una vez que el brazo del disco llega a un cilindro determinado, puede haber muchas solicitudes pendientes para las diversas pistas de dicho cilindro. Examina todas estas solicitudes y atiende primero la que tiene el retraso rotacional más corto.

La optimización rotacional se denomina a veces puesta en cola de sectores; los sectores se colocan en una cola según su posición alrededor del disco y se atiende primero los sectores más cercanos.

Consideraciones en los sistemas

Cuando el almacenamiento de disco resulta un cuello de botella, se tiende a agregar más discos al sistema. Hacerlo no siempre resuelve el problema

porque el cuello de botella puede deberse a una carga pesada de solicitudes sobre un número relativamente pequeño de discos. Si se detecta esta situación, la planificación de disco puede servir para mejorar la eficiencia y eliminar el cuello de botella.

- ◆ Memoria caché de disco.- Es un área de almacenamiento primario en la que se conservan los registros de acceso frecuente para ayudar a evitar la necesidad de búsquedas largas de disco.

Cuando se realiza una operación de escritura, podría pensarse que el registro se graba de inmediato en el disco. Así sucede en algunos sistemas, pero en otros con memoria caché de disco la escritura sólo hace que el registro se almacene en un buffer en almacenamiento primario; el registro permanece ahí hasta que el sistema se queda sin espacio de buffer para escrituras subsecuentes, y en ese momento se graba el registro en el disco.

Si fuera necesario leer un registro escrito recientemente, se puede obtener del buffer de memoria caché de disco en almacenamiento primario mucho más rápido que si tuviera que leer del disco.

La clave para aprovechar la memoria caché de disco es mantener los registros de acceso frecuente en el buffer de memoria caché de disco de almacenamiento primario. Desde luego esta técnica sólo funciona bien cuando es posible identificar los registros de acceso frecuente. Utilizando la heurística de localidad, un registro al que se hizo referencia en el pasado reciente tal vez será requerido en el futuro cercano.

Otras técnicas para mejorar el desempeño

La optimización del desempeño de los dispositivos rotacionales de almacenamiento se ha tratado de lograr empleando métodos de hardware, de sistemas operativos y de sistemas de aplicación. Por ello podríamos decir que otras técnicas para mejorar el desempeño son:

- La reorganización del disco para reducir la fragmentación;

- La partición del disco en la que se confinan los archivos a zonas pequeñas a fin de reducir la fragmentación;
- Colocar varias copias de datos estables de referencia frecuente en muchas partes del disco para reducir las distancias de búsqueda;
- La duplicación de datos estables de referencia frecuente en unidades de disco de acceso individual para lograr una mayor concurrencia;
- La agrupación de registros en bloques para reducir el número de búsquedas;
- El mantenimiento de datos de acceso frecuente en posiciones de acceso más rápido dentro de la jerarquía de almacenamiento;
- La colocación de datos en las pistas de la zona media en dispositivos que utilizan una planificación tipo SCAN;
- La lectura de una pista completa a la vez para aprovechar la localidad espacial
- La compresión de datos para reducir el espacio requerido y por consiguiente, los tiempos de acceso

En general se ha evitado el aumento en las velocidades de rotación de los discos por causa de limitaciones físicas.

-
- ◆ **Discos de RAM.-** Es un dispositivo de disco simulado en memoria convencional de acceso aleatorio. Elimina por completo los retrasos que se dan en los discos convencionales a causa de los movimientos mecánicos inherentes a las búsquedas y a la rotación del disco (los discos en RAM no implican movimientos mecánicos). Son útiles sobre todo en las aplicaciones de alto rendimiento. Están separados de la memoria principal, de modo que no utilizan espacio requerido por el sistema operativo o las aplicaciones. Los tiempos de referencia a datos individuales son uniformes, sin la amplia variabilidad de los discos convencionales. Los discos en RAM son caros. Son volátiles, es decir pierden su contenido cuando se apaga la computadora o se interrumpe el suministro de energía, podríamos

solucionar esto con el uso de baterías de respaldo (UPS), pero si el corte de energía se prolonga estas se pueden agotar por lo que siempre es más recomendable hacer respaldos frecuentes en discos convencionales.

- ◆ **Discos ópticos.-** Pueden contener enormes cantidades de datos bajo un régimen de grabación-única-lectura-múltiple o WORM write-once-read-many (los primeros discos láser). Los discos láser regrabables podrían desplazar los discos magnéticos de cabeza móvil. Resultan apropiados para aplicaciones de archivo, pero no son muy útiles para aplicaciones que requieren actualización regular. Las capacidades enormes de estos discos (varios gigabytes, quizá 100 veces mayor que los discos duros empleados normalmente en las PC's) han hecho que resulten realmente efectivos en algunas aplicaciones

RELOJES

También llamados cronómetros son esenciales para la operación de cualquier sistema de tiempo compartido. El software de reloj toma la forma de un manejador de dispositivo, aunque un reloj toma la forma de un manejador de dispositivo, no es un dispositivo de bloque, como un disco, ni un dispositivo de carácter, como una terminal.

Cronómetro de intervalos o reloj de interrupciones

Los relojes no tienen direcciones por medio de bloques, tampoco generan o aceptan flujos de caracteres, lo único que hacen es provocar interrupciones a intervalos bien definidos.

Un proceso que tiene asignado el CPU esta en ejecución. Si el proceso pertenece al sistema operativo, se dice que el sistema operativo esta en ejecución y puede tomar decisiones que afectan la operación del sistema. Para evitar que los usuarios monopolicen el sistema (accidental o deliberadamente), el sistema operativo tiene mecanismos para arrebatarse el CPU al usuario.

El sistema operativo mantiene un reloj de interrupciones o cronómetro de intervalos para generar interrupciones en algún momento futuro específico (o

después de cierto tiempo). El reloj de interrupciones ayuda a garantizar tiempos de respuesta aceptables para los usuarios interactivos, evita que el sistema quede bloqueado en un ciclo infinito de algún usuario y permite que los procesos respondan a eventos dependientes de tiempo. Los procesos que deben ejecutarse periódicamente dependen del reloj de interrupciones.

Por lo general las labores de un reloj son:

1. Mantener la hora del día
2. Evitar que los procesos se ejecuten más tiempo del permitido.
3. Mantener un registro del uso del CPU
4. Controlar la llamada al sistema ALARM por parte de los procesos del usuario.
5. Proporcionar cronómetros guardianes de partes del propio sistema.
6. Realizar resúmenes, monitoreo y recolección de estadísticas

TERMINALES

Cada computadora tiene una o más terminales que se utilizan para comunicarse con ella. Las terminales tienen un gran número de formas distintas. El manejador de terminal se encarga de ocultar todas estas diferencias, de forma que la parte independiente del dispositivo en el sistema operativo y los programas de usuario no tienen que volverse a escribir para cada tipo de terminal.

Los editores de pantalla y muchos otros programas sofisticados deben poder actualizar la pantalla de maneras complejas que le simple recorrido del texto en la parte inferior de la misma. Para lograr esto, muchos manejadores de terminales soportan varias secuencias de escape. A continuación se listan algunas:

1. Mover el cursor hacia arriba, abajo, a la izquierda o a la derecha una posición.
2. Mover el cursor a x , y
3. Insertar un caracter o una línea n el cursor

4. Eliminar un caracter o una línea del cursor
5. Recorrer la pantalla hacia arriba o hacia abajo n líneas
6. Limpiar la pantalla desde el cursor hasta el final de la línea o hasta el final de la pantalla.
7. Trabajar en modo vídeo inverso, subrayado, parpadeo o normal
8. Crear, destruir, mover o controlar ventanas.

Cuando el manejador ve el caracter que inicia la secuencia de escape, activa una bandera y espera a que llegue el resto de la secuencia. Cuando toda la secuencia ha llegado, el manejador debe llevarla a cabo en software. La inserción y eliminación de texto requiere el movimiento de bloques de caracteres en el vídeo RAM. El hardware no ayuda más que en recorrer descendente o ascendente texto y en exhibir el cursor.

DESEMPEÑO, COPROCESADORES, RISC Y FLUJO DE DATOS

Tres objetivos comunes de la evaluación del rendimiento:

- 1) **Evaluación para la Selección.**- El evaluador del desempeño debe decidir si conviene adquirir un sistema de cómputo de un proveedor específico.
- 2) **Proyección del Rendimiento.**- El objetivo del evaluador en este caso es estimar el desempeño de un sistema inexistente. Puede tratarse de un sistema de cómputo totalmente nuevo ó de un nuevo componente de hardware o software.
- 3) **Supervisión del Desempeño.**- El evaluador acumula datos de un sistema ó componente ya existente para asegurarse de que cumple con sus objetivos de desempeño, para ayudar a estimar el impacto de las modificaciones y para ofrecer a los administradores la información que necesitan para tomar decisiones estratégicas tales como modificar o no un sistema existente de prioridades de trabajo.

Evaluación y Predicción del Desempeño.- Se necesitan desde los primeros momentos de la concepción de un nuevo sistema, en la operación cotidiana del sistema después de la instalación y cuando se estudia la modificación o posible sustitución de un mejor sistema.

Desempeño.- Es la eficiencia con la que un sistema cumple sus objetivos.

Medidas de desempeño

- ◆ **Tiempo de retorno.-** En un sistema de procesamiento por lotes se define como el tiempo transcurrido desde la entrega de un trabajo hasta la devolución al usuario.
 - ◆ **Tiempos de respuesta.-** Es el tiempo de retorno en un sistema interactivo y a menudo se le define como el tiempo transcurrido desde que el usuario presiona el "enter" ó "botón del mouse" hasta que el sistema comienza a imprimir o exhibir una respuesta.
 - ◆ **Tiempo de reacción del sistema.-** Es el tiempo transcurrido desde que el usuario presiona "enter" ó el "botón del mouse" hasta que se otorga la primera tajada de tiempo de servicio a solicitud del usuario.
-
- ◆ **Varianza en los tiempos de respuesta.-** Es una medida de predecibilidad.
 - ◆ **Producción.-** Es la medida del rendimiento de trabajo por unidad de tiempo.
 - ◆ **Carga de trabajo.-** Es la medida de la cantidad de trabajo que se ha introducido al sistema, cantidad que el sistema debe procesar en condiciones normales para que su funcionamiento se considere aceptable.
 - ◆ **Capacidad.-** Es la medida de producción máxima que puede tener un sistema.
 - ◆ **Utilización.-** Es la fracción de tiempo en que está en uso un recurso (es una medida engañosa porque al parecer lo mejor es tener un porcentaje de utilización alto y esto podría ser señal de un aprovechamiento ineficiente).
- ### **Técnicas para Evaluar el Desempeño**
- ◆ **Tiempos.-** Son útiles para realizar comparaciones rápidas entre equipos.

- ◆ **Mezcla de instrucciones.**- Emplean un promedio ponderado de diversos tiempos de instrucciones más adecuadas para una aplicación específica.
- ◆ **Programa núcleo.**- Es un programa representativo que podría ejecutarse en una instalación. Se cronometra para una máquina dada, empleando las estimaciones de tiempo de instrucciones del fabricante y así se pueden hacer comparaciones entre máquinas distintas de acuerdo con la velocidad de ejecución esperada del programa núcleo.
- ◆ **Modelos analíticos.**- Son representaciones matemáticas de sistemas de cómputo o de sus componentes. Existe un volumen considerable de resultados matemáticos que puede aplicar el evaluador para ayudar a estimar el desempeño de un sistema de cómputo dado (Ejemplo: Teoría de colas y modelo de Markov).
- ◆ **Bancos de prueba.**- Es un programa real que el evaluador ejecuta en un sistema de cómputo en evaluación. El evaluador conoce las características del rendimiento del banco de prueba en un equipo ya existente de tal modo que cuando se ejecuta en un equipo nuevo el evaluador puede sacar conclusiones significativas.
- ◆ **Programas sintéticos.**- Son programas reales diseñados a la medida para ejercitar funciones específicas de un sistema de cómputo. Son útiles sobre todo cuando no existen bancos de prueba que realicen dichas funciones.
- ◆ **Simulación.**- Es una técnica con la cual el evaluador desarrolla un modelo computarizado del sistema de evaluación. Luego el modelo se ejecuta en un sistema de cómputo con lo que se refleja el comportamiento del sistema en evaluación.
 - ⇒ Hay simuladores que se manejan por eventos que se producen en el simulador de acuerdo con distribuciones de probabilidades.
 - ⇒ Los simuladores manejados por libreto son controlados por datos derivados empíricamente y manipulados con cuidado de manera que reflejen el comportamiento esperado del sistema simulado.

- ◆ **Supervisión del desempeño.-** Es la obtención y el análisis de información acerca del rendimiento de sistemas ya existentes.

Cuellos de Botella y Saturación

- ◆ **Cuello de botella.-** Un recurso se convierte en cuello de botella y limita el desempeño total del sistema cuando no es capaz de manejar el trabajo que se le envía.
- ◆ **Saturación.-** Los recursos que operan cerca de su capacidad máxima tienden a saturarse, es decir, los procesos que compiten por la atención del recurso comienzan a interferir unos con otros (Ejemplo: Hiperpaginación)
- ◆ **Ciclo de retroalimentación.-** Es un caso en el que cierta información acerca del estado actual del sistema puede afectar las solicitudes que llegan. Si la retroalimentación indica que dichas solicitudes pueden tener problemas para ser atendidas, quizás se les pueda enviar por otro conducto.
- ◆ **Retroalimentación negativa.-** La tasa de llegada de retroalimentación negativa contribuye a la estabilidad en sistemas manejados por colas (La tasa de llegada de solicitudes nuevas puede disminuir como resultado de la información retroalimentada).
- ◆ **Retroalimentación positiva.-** En estos sistemas, la información retroalimentada origina un aumento en algún parámetro, puede provocar inestabilidad en sistemas manejados por colas.
- ◆ **Coprocesadores.-** Son procesadores de aplicación especial que se agregan a los sistemas de cómputo para realizar operaciones que no están incluidas en el procesador o procesadores originales.
- ◆ **RISC (Computación con un conjunto reducido de instrucciones).-** Las arquitecturas de un RISC casi siempre tiene conjuntos de instrucciones en lenguaje de máquina poco abundantes relativamente sencillas de carga y almacenamiento para transferir datos entre la memoria y los registros conductos profundos y memoria caché. Aunque los programas RISC suelen

ser más tardados que sus equivalentes en CISC, casi siempre se ejecutan en forma rápida.

COMPUTACIÓN DISTRIBUIDA: LA PERSPECTIVA DE LA COMPUTACIÓN EN PARALELO.

♦ **Canalización.**- Es una técnica para mejorar el rendimiento permitiendo que varias instrucciones en lenguaje de máquina estén en diferentes etapas de ejecución al mismo tiempo. Cada etapa de la canalización de instrucciones realiza una etapa diferente sobre la instrucción y después la instrucción avanza a la siguiente etapa. La canalización hace posible que muchas instrucciones, cada una en diferente fase de ejecución, progresen al mismo tiempo.

♦ **Multiprocesadores.**-Uno de los atractivos de un sistema con múltiples procesadores o multiprocesamiento es que si falla un procesador por lo regular se puede seguir trabajando con los procesadores restantes. El sistema operativo debe darse cuenta de que un procesador específico ha fallado y ya no está disponible para ser asignado.

Hay dos técnicas comunes para detectar el paralelismo:

1. Distribución de ciclos: Considere la siguiente instrucción que suma los elementos de los arreglos "b" y "c" y coloca las sumas en el arreglo "a". Este ciclo hace que un procesado secuencial realice las 4 interacciones del ciclo una tras otra.

```
for i = 1 to 4 do
  a(i) = b(i) + c(i)
cobegin
  a(1) = b(1) + c(1)
  a(2) = b(2) + c(2)
  a(3) = b(3) + c(3)
  a(4) = b(4) + c(4)
coend
```

Todas las interacciones se pueden realizar independientemente una de otra separándose en unidades ejecutables de manera concurrente (al mismo tiempo).

2. Reducción de Altura de Árboles.- Aplica las propiedades de conmutatividad, asociatividad y propiedad distributiva de las operaciones aritméticas para reacomodar las expresiones algebraicas y hacerlas más adecuadas para la ejecución concurrente.

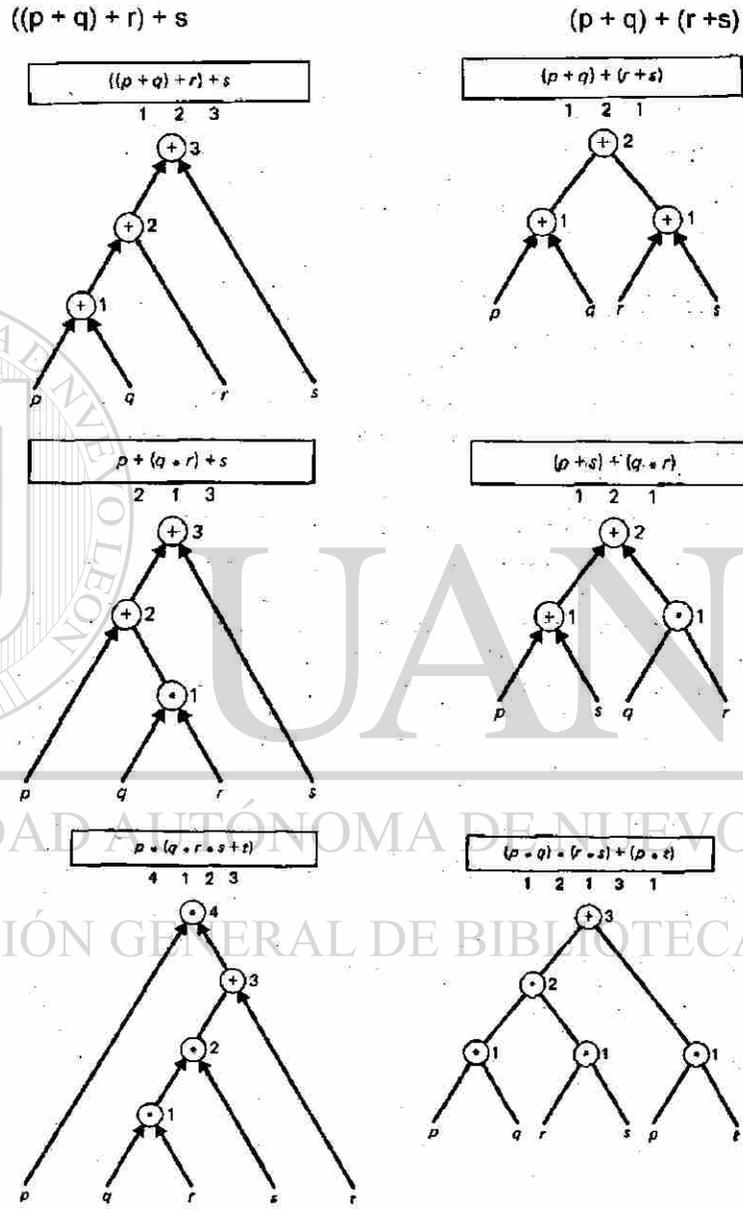


Fig.6.3. Aplicaciones de las propiedades asociativa, conmutativa y distributiva en la reducción de la altura de los árboles

Regla de Nunca Esperar.- Establece que es mejor darle al procesador una tarea que puede ser utilizada o no posteriormente que dejar al procesador sin trabajo.

ESQUEMAS DE INTERCONEXION DE LOS PROCESADORES

Los problemas fundamentales en el diseño de sistemas de multiprocesamiento son determinar la forma de conectar los múltiples procesadores y los procesadores de E/S con las unidades de almacenamiento.

Esquema de interconexión “Ducto Compartido”.- utiliza una sola trayectoria de comunicación con todos los procesadores, las unidades de almacenamiento y los procesadores de E/S. Esencialmente es una unidad pasiva.

Ejemplo: Esquema de red de área local: Ethernet.

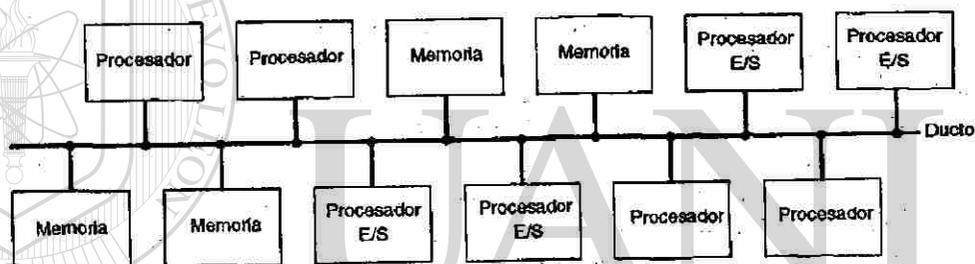


Fig. 6.4. Bus o ducto compartido

Ventajas:

- Se le pueden fácilmente adicionar nuevas unidades, basta conectarlas directamente al ducto para que ocurra la comunicación, cada unidad debe saber cuales otras unidades están conectadas al ducto, esto se maneja por software.

Desventajas:

- La principal se debe a la trayectoria única que tiene el ducto compartido.
- Sólo se puede manejar una transmisión a la vez.
- La velocidad de transmisión esta en función de la velocidad de transmisión del ducto.
- Una de las peores desventajas es la contención por el uso del ducto (se degrada el rendimiento del sistema).

Esquema de Interconexión “Matriz de Conmutadores Cruzados”.- puede manejar transmisiones simultáneas a todas las unidades de almacenamiento, por lo cual el conmutador cruzado es un esquema de conexión de mayor rendimiento. Hace posible realizar transmisiones simultáneas en ambos sentidos en todos los pares posibles de unidades de almacenamiento. La mayor desventaja es el grado de complejidad del conmutador cruzado.

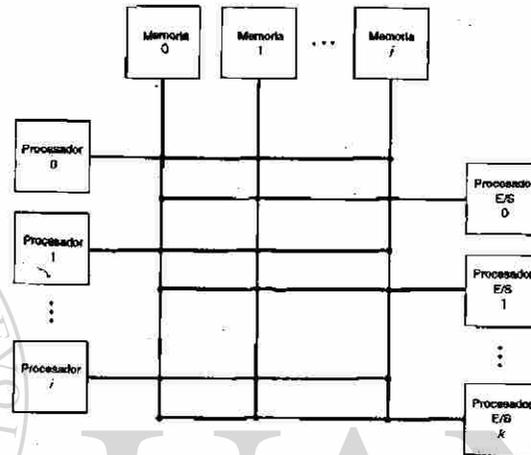


Fig. 6.5. Matriz de conmutadores cruzados

Esquema de Interconexión “Hiper cubo”.- la red de interconexión hiper cubo hace posible conectar un gran número de procesadores en forma relativamente económica. Utiliza un esquema de conexión muy económica y un hiper cubo de 16 dimensiones puede conectar 65,536 procesadores.

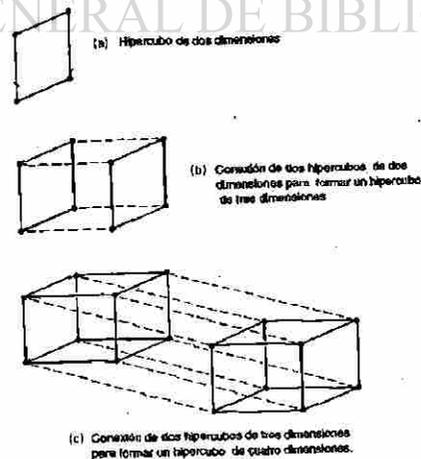


Fig. 6.6. Hiper cubo

Esquema de Interconexión de Redes de Múltiples Etapas.- es un esquema de concesión, permite a cualquier unidad conectarse con cualquier otra y reduce mucho la complejidad del esquema de interconexión para que sea factible conectar un gran número de procesadores. El término múltiples etapas se debe a que cada mensaje debe pasar por un gran número de elementos de conmutación antes de alcanzar su destino. Los retrasos que se presentan a medida que un mensaje pasa por los diversos tramos hace que éste esquema sea más lento que el de conmutadores cruzados pero el número de elementos de conmutación es mucho menor sobre todo cuando se conecta un gran número de procesadores, éste esquema ofrece una trayectoria única a través de la red pero puede ocurrir contención de los diferentes elementos de conmutación implicados en las diferentes trayectorias. Algunos de éste tipo de redes almacenan temporalmente los mensajes que chocan en los elementos de conmutación hasta que se despejan los elementos pueden proseguir los mensajes, esto aumenta el costo de la red de computación. Otros sistemas aceptan y procesan solo un mensaje en el conmutador y devuelven los demás mensajes que chocan para ser retransmitidos posteriormente con un retraso aleatorio.

Sistemas Fuertemente y Débilmente Acoplados

Multiprocesamiento débilmente acoplado

Implica conectar dos o más sistemas de cómputo independientes mediante un enlace de comunicación. Cada sistema tiene su propio (procesador) sistema operativo y su almacenamiento. Los sistemas pueden funcionar en forma independiente y se pueden comunicar entre sí si es necesario.

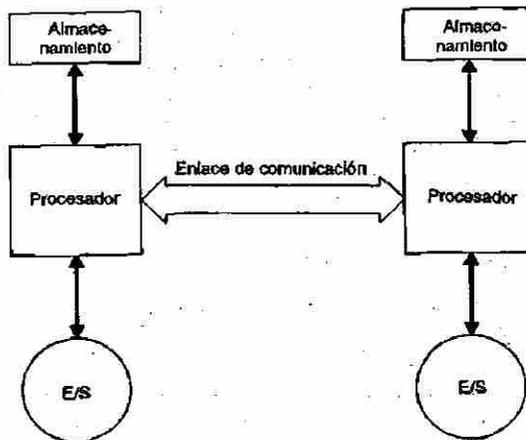


Fig. 6.7 Multiprocesamiento débilmente acoplado

Multiprocesamiento fuertemente acoplado

Utiliza un almacenamiento único compartido por los diferentes procesadores y un solo sistema operativo que controla todos los procesadores y el hardware del sistema. La comunicación se realiza con la memoria compartida.

Un aspecto clave de los sistemas fuertemente acoplados es la contención por la memoria compartida. Algunos estudios indican que si se equilibra la arquitectura y se distribuye la carga de trabajo a lo largo del sistema la contención es mínima.

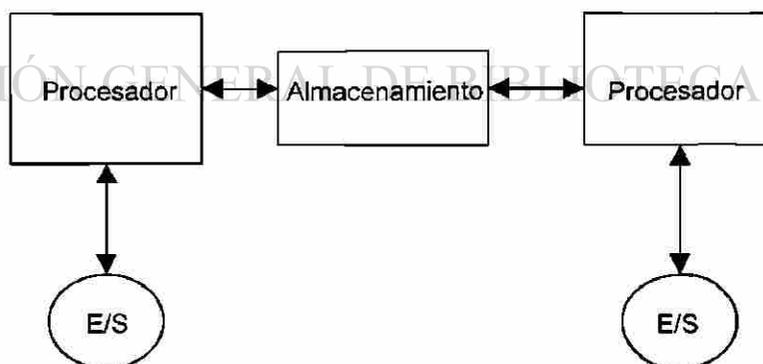


Fig. 6.8 Multiprocesamiento fuertemente acoplado

Organizaciones básicas de los sistemas operativos para multiprocesadores

La capacidad para aprovechar el paralelismo en el hardware y en los programas es fundamental. La adición de más procesadores así como las complejas conexiones con la memoria y los procesadores de E/S aumenta considerablemente el costo del hardware.

Amo-Esclavo.- es la más fácil de llevar a la práctica, solo el Amo puede ejecutar el sistema operativo el cual no tiene que ser reentrante y se simplifica mucho la exclusión mutua en el acceso a los datos del sistema. Si el Amo falla el sistema ya no podrá efectuar operaciones de E/S.

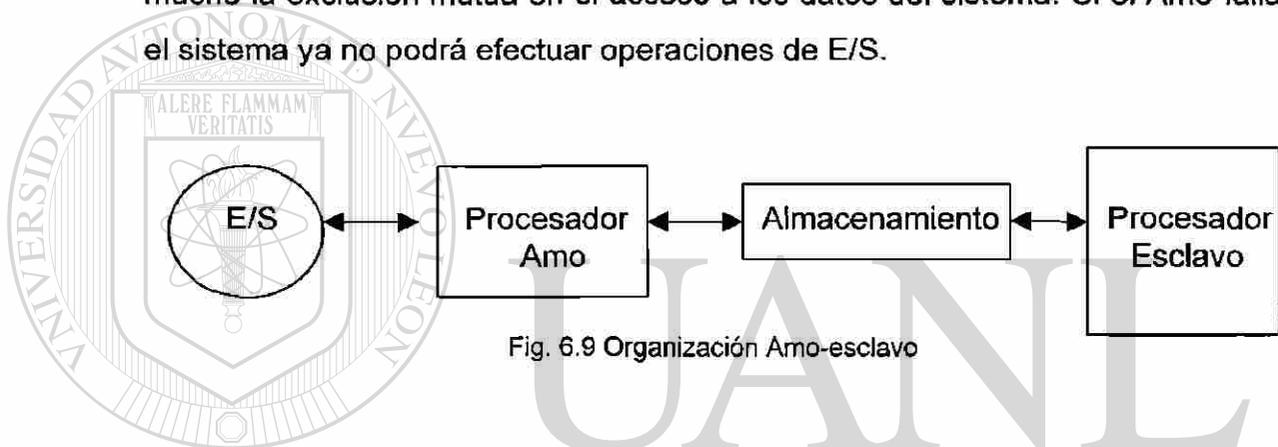


Fig. 6.9 Organización Amo-esclavo

Ejecutivos Individuales.- en su realización cada procesador tiene su propio sistema operativo y controla sus propios recursos. Cada proceso se ejecuta hasta completarse en el procesador al que fue asignado.

Organización Simétrica.- el multiprocesador simétrico es la organización más compleja de llevar a la práctica pero también es la más poderosa. El sistema operativo administra un conjunto de procesadores. La falla de algún procesador hace que el sistema operativo lo retire del conjunto de procesadores pero sigue funcionando. Un proceso dado se puede ejecutar en distintas ocasiones en procesadores diferentes.

La contención podría convertirse en un problema grave debido a que varios procesadores pueden estar en estado supervisor al mismo tiempo.

Anexo 2

Técnicas de Dinámica de Grupo

Juego de verano.

Este juego esta orientado hacia el grupo y hacia la tarea. Es un medio de evaluar diagnóstico, formativa y sumativamente ya que permite ver el estado inicial del alumno (en cuanto a conocimientos, actitudes, habilidades, destrezas, etc.), el avance del diseño día con día y en la competencia que se lleva a cabo el último día antes del segundo examen parcial se ve reflejado los conocimientos, habilidades, destrezas, etc.

En esta ocasión fue implementado como trabajo final, pero puede ser usado indistintamente con las correcciones a que dé lugar.

Objetivo explícito.-

Analizar, estudiar y comprender el material del curso.

Objetivo implícito.-

- Desarrollar la habilidad para trabajar en equipo
- Incrementar el sentido de responsabilidad y solidaridad con sus compañeros.
- Fomentar la participación.
- Fomentar la creatividad
- Transmitir y fomentar la Investigación.
- Proponer y poner en práctica nuevos conocimientos

Mecánica:

1. Se le pide al grupo que contribuya dando ideas para formar las reglas del juego. Se someterá a votación para determinar su permanencia en el reglamento eliminándose así las reglas excedentes. Esto será un acuerdo de grupo.
2. Se les solicita que diseñen el material con el cual se ha de jugar (manta,

lona, pirinola, dados, tarjetas rojas, amarillas, envases de refresco, etc.). Esto será un acuerdo de grupo.

3. Se arman siete equipos horizontales de 4 a 6 personas.
4. Se le asignan a cada equipo una unidad del libro de apuntes, se les pide que lean el material y que busquen en Internet o en algún libro o revista de literatura acerca del tema que les fue asignado.
Todo esto se hace en la primer sesión (en verano se manejan 3 horas de 50 minutos seguidas) después del encuadre.
5. Se les pide que generen un cuestionario de 20 o 30 preguntas con sus respectivas respuestas. De aquí se seleccionarán las más acordes con el material que debe verse en clase.
Los avances se verifican día con día.
6. Se organizan competencias sobre la base de lo que todos deben estudiar (la unidad que les fue asignada).
7. Se arman equipos verticales.

Instrucciones para el maestro:

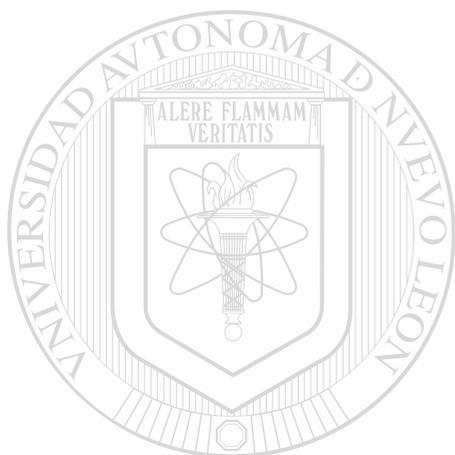
1. Todos participaran, persona que sea sorprendida sin hacer aportaciones se les sacará tarjeta amarilla para marcarlo y tarjeta roja para expulsarlo, este alumno deberá hacer méritos en otro equipo o en el mismo para poder ser admitido como miembro, esto implica un trabajo extra y será su amonestación por su pasividad.
2. Al dar inicio a la competencia se usará el material que ellos mismos diseñaron.
3. En este juego el maestro será moderador y se podrá evaluar el aprendizaje.

A continuación se presenta el reglamento diseñado por alumnos, cabe señalar que de aquí se pueden sacar conclusiones significativas:

1. Se trabajará en equipo, de forma equitativa, con la participación de todos los miembros del equipo.
2. No burlarse de nadie.
3. Tener la seriedad conveniente.
4. Se dará un minuto para contestar.
5. Se usarán dados (de hielo seco o de hule espuma) para escoger el rol de preguntas.
6. Ganará el equipo que llegue primero a la meta o el que acumule mayor número de puntos.
7. Pregunta que no se conteste se repetirá.
8. Se contestará en forma completa o con la idea principal.
9. Designar un representante de equipo.
10. El maestro o maestra hará las preguntas.
11. Se desalojará el salón con el fin de ubicar la manta diseñada por el grupo.
12. La pregunta se escogerá al azar.
13. El equipo en turno al contestar correctamente avanzará una casilla o se le dará un punto. Si no se sabe la respuesta o esta es incorrecta se dará oportunidad a otro equipo.
14. Todos los equipos tendrán el mismo número de participaciones. ®
15. Todos los equipos tendrán las preguntas y las respuestas elaboradas por los demás equipos.

Anexo 3

Con el fin de hacer la clase más amena y sobre todo para tratar los temas difíciles se contempla el uso de una microcomputadora PC MMX con 64 o 128 M en RAM, Tarjeta Sound Blaster, unidad para CD de 24X , hard disk con un mínimo de 2 gigas (puede ser más), tarjeta de vídeo True Color SVGA con salida TV Coder, lo anterior para correr las “animaciones” de los temas no-tangibles, con sonido y alta resolución que puedan conectarse a una televisión en el caso de que no este disponible el datashow.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

Anexo 4

1. **Primer Plan de Estudios de la Carrera de Ingeniero Administrador de Sistemas**
2. **Programa de Clase de Introducción a los Sistemas Operativos 1985.**
3. **Programa de Clase de Sistemas Operativos I 1988**
4. **Programa de Clase de Sistemas Operativos II 1988**
5. **Programa de Clase de Sistemas Operativos I 1995**



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



DIRECCIÓN GENERAL DE BIBLIOTECAS

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
 FACULTAD DE INGENIERIA MECANICA Y ELECTRICA.
 PROYECTO DE PLAN DE ESTUDIOS 1975

INGENIERO ADMINISTRADOR DE SISTEMAS

MATERIAS

FRECUENCIAS

PRIMER SEMESTRE

Matemáticas I	5
Física I y Lab.	5
Dibujo	3
Tecnología de Materiales y Lab.	5
Algebra	5
Química	5
	<u>28</u>

SEGUNDO SEMESTRE

Matemáticas II	5
Física II y Lab.	5
Mecánica Vectorial	5
Física III y Lab.	5
Contabilidad General	3
Balance de Energía	5
	<u>28</u> (R)

DIRECCIÓN GENERAL DE BIBLIOTECAS

TERCER SEMESTRE

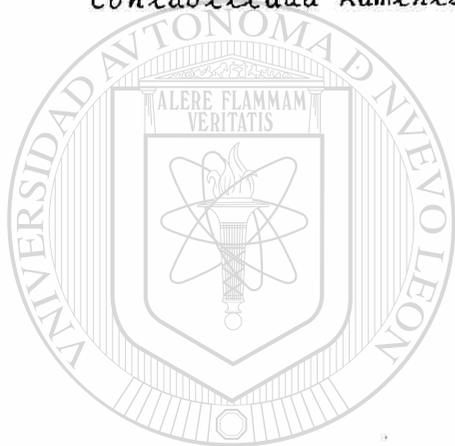
Matemáticas III	5
Análisis Numérico y Lab.	5
Estadística I	5
Circuitos Eléctricos I	5
Economía	5
Técnicas Legales y Lab.	5
	<u>30</u>

MATERIAS	FRECUENCIAS
<u>CUARTO SEMESTRE</u>	
Ecuaciones Diferenciales	5
Algoritmos Computacionales	5
Estadística II	5
Introducción a los Sistemas Computacionales	5
Programación Lineal	5
Teoría Administrativa y Lab.	5
	<u>30</u>
<u>QUINTO SEMESTRE</u>	
Transformadas de Laplace y Series de Fourier	5
Programación Digital y Lab.	5
Diseño de Experimentos	5
Introducción a los Sistemas Operativos	5
Investigación de Operaciones I	5
Ingeniería de Métodos y Lab.	5
	<u>30</u>
<u>SEXTO SEMESTRE</u>	
Teoría de Control	5
Programación Digital Avanzada	5
Ingeniería de Sistemas I	5
Teleproceso	5
Control de Producción I	5
Costos Industriales	5
	<u>30</u>
<u>SEPTIMO SEMESTRE</u>	
Computación Analógica y Lab.	5
Control de Calidad y Lab.	5
Ingeniería de Sistemas II	5
Investigación de Operaciones II	5
Control de Producción II	5
Ingeniería Económica y Lab.	5
	<u>30</u>

MATERIAS**FRECUENCIAS**

OCTAVO SEMESTRE

Relaciones Industriales y Lab.	5
Psicología Industrial	5
Diseño de Sistemas	5
Informática	5
Diseño de Sistemas de Producción	5
Contabilidad Administrativa	5
	<hr/>
	30
	<hr/>

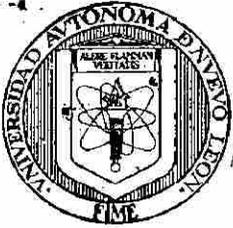


UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS





UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
 Facultad de Ingeniería Mecánica y Eléctrica
 Ciudad Universitaria
 Monterrey, N. L., México

PROGRAMA DE CLASE

MATERIA:- INTRODUCCION A LOS SISTEMAS OPERATIVOS
 FRECUENCIA:- 5 HORAS A LA SEMANA
 DURACION:- 80 SESIONES
 AREA:- SISTEMAS
 GRADO:- QUINTO SEMESTRE
 NIVEL:- LICENCIATURA
 PRE-REQUISITO:- INTRODUCCION A LOS SISTEMAS COMPUTACIONALES

I.- FUNDAMENTOS:-

El sistema operativo es el programa de control bajo el cual un computador digital funciona, por lo que esta clase es fundamental para poder utilizar un computador y entender su funcionamiento.

II.- OBJETIVOS GENERALES:-

En la clase de introducción a los sistemas operativos el alumno deberá aprender a reafirmar sus conocimientos de Hardware, aprender el Software básico de un computador y la forma en que este interactúa con el Hardware.

III.- OBJETIVOS PARTICULARES:-

El alumno deberá repasar la estructura básica del hardware y aprender las funciones y facilidades de un sistema operativo, el lenguaje máquina, lenguaje ensamblador; cargadores, estructuras de datos, características de lenguajes de alto nivel, los compiladores de los lenguajes de alto nivel, y las funciones de administración del equipo.

IV.- CONTENIDO TEMÁTICO Y OBJETIVOS ESPECÍFICOS.

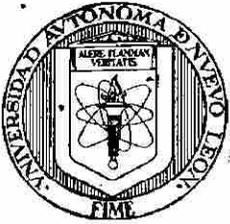
TEMA No. 1: INTRODUCCION

No. de SESIONES
15

- I.1.- Estructura de un Computador
 - 1.1.1.- CPU (UNIDAD CENTRAL DE PROCESO)
 - 1.1.2.- Memoria Principal
 - 1.1.3.- Canales de I/O (Entrada/Salida)
 - 1.1.4.- Dispositivos de I/O (Entrada/Salida)
 - 1.1.5.- Teleproceso
- I.2.- Tipos de un Sistema Operativo
 - 1.2.1.- De un solo usuario
 - 1.2.2.- Sistema operativo multiprogramable
 - 1.2.3.- Sistema operativo de tiempo real

OBJETIVOS ESPECÍFICOS.

- 1.- El alumno deberá aprender o reafirmar la estructura de un Computa



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
Facultad de Ingeniería Mecánica y Eléctrica
Ciudad Universitaria
Monterrey, N. L., México

- 2 -

- dor digital, Entender su funcionamiento y su relación con el sistema
2.- El alumno deberá aprender los diferentes tipos de sistemas operativos existentes, sus filosofías de diseño y sus diferencias - - principales.

TEMA No. 2.- FUNCIONES DE UN SISTEMA No. DE SESIONES
3

- 2.1.- Introducción
2.2.- Explicación general de las funciones

OBJETIVOS ESPECIFICOS.

- 1.- El alumno deberá aprender y entender las funciones principales de un sistema operativo general, las principales funciones se verán a -- d fondo en el punto X de programa.

TEMA No. 3.- FACILIDADES DE UN SISTEMA OPERATIVO 20

- 3.1.- Ensambladores
3.2.- Compiladores
3.3.- Editores
3.4.- Cargadores
3.5.- Utilerías
3.6.- Facilidades de Debugging
3.7.- Paquetes de aplicación

OBJETIVOS ESPECIFICOS.

- 1.- El alumno deberá conocer el objeto y funcionamiento de todas las facilidades del sistema operativo.
2.- Además el alumno deberá aprender como el sistema operativo le permite utilizar todas sus facilidades.

TEMA No. 4.- LENGUAJE MAQUINA No. DE SESIONES
4

- 4.1.- Introducción
4.2.- Formato de instrucción

OBJETIVOS ESPECIFICOS.

- 1.- El alumno deberá conocer las características del Lenguaje máquina y su funcionamiento.
2.- El alumno deberá entender los formatos de instrucción del lenguaje máquina.

TEMA No. 5.- ENSAMBLADORES No. DE SESIONES
6

- 5.1.- Funciones
5.2.- Estatutos
5.3.- Aplicaciones



- 3 -

OBJETIVOS ESPECIFICOS.

- 1.- El alumno deberá aprender el funcionamiento de un ensamblador y sus funciones principales estatutós.
- 2.- Además el alumno deberá reconocer en que situaciones se convierte la aplicación del ensamblador.

TEMA No. 6.- MACRO-ENSAMBLADORES

No. DE SESIONES

4

- 6.1.- Objetivo y funciones
- 6.2.- Macro-instrucciones

OBJETIVOS ESPECIFICOS.

- 1.- El alumno deberá entender la diferencia entre un ensamblador y un macro-ensamblador
- 2.- Además deberá aprender cuales son sus intrucciones y sus posi-
bles aplicaciones.

TEMA No. 7.- CARGADORES

No. DE SESIONES

4

- 7.1.- Generalidades
- 7.2.- Tipos de cargadores

OBJETIVOS ESPECIFICOS.

- 1.- El alumno deberá entender el funcionamiento de un cargador y su aplicación según su tipo

TEMA No. 8.- LENGUAJES DE PROGRAMACION

No. DE SESIONES

6

- 8.1.- Características y usos
- 8.2.- Estructuras y datos

OBJETIVOS ESPECIFICOS.

- 1.- El alumno deberá conocer las características y usos de los lenguajes de programación de alto nivel y para cada uno de ellos las estructuras de datos que soportan

TEMA No. 9.- LENGUAJES DE PROGRAMACION

No. DE SESIONES

8

- 9.1.- Funciones
- 9.2.- Facilidades

OBJETIVOS ESPECIFICOS.

- 1.- El alumno deberá aprender como utilizar el compilador de cada lenguaje de alto nivel para transformar un programa fuente de ese lenguaje a un programa ejecutable por la maquina.



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
 Facultad de Ingeniería Mecánica y Eléctrica
 Ciudad Universitaria
 Monterrey, N. L., México

- 4 -

TEMA No. 10. FUNCIONES PRINCIPALES DE UN SISTEMA OPERATIVO No. DE SESIONES
 10

- 10.1.- Procesamiento de I/O
- 10.2.- Administración y manejo de memoria
- 10.3.- Administración y manejo del procesador
- 10.4.- Administración y manejo de periféricos
- 10.5.- Administración y manejo de la información

OBJETIVOS ESPECÍFICOS

- 1.- El alumno deberá entender que el sistema operativo tiene como función principal manejar 2 factores que son el tiempo y el espacio
- 2.- El tiempo del procesador y de uso de los periféricos y el espacio de la memoria y dispositivos secundarios de almacenamiento.

V.- METODOLOGIA

El curso de introducción a los Sistemas Operativos es llevado de la siguiente manera; el Maestro por lo general explica los temas de programa y para cada tema utiliza como ejemplos los diferentes tipos de sistemas operativos actuales o encarga las exposiciones de algún punto de los alumnos, inclusive aunque el maestro ya lo haya expuesto.

VI.- CRITERIO DE EVALUACION

El criterio de evaluación del curso de Introducción a los Sistemas Operativos es el siguiente:

- 1.- 2 Exámenes Parciales
- 2.- 1 Proyecto final
- 3.- Tareas o ejercicios

Los porcentajes se dejan al criterio de el maestro, como también las tareas o ejercicios o el proyecto final.



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
Facultad de Ingeniería Mecánica y Eléctrica
Ciudad Universitaria
Monterrey, N. L., México

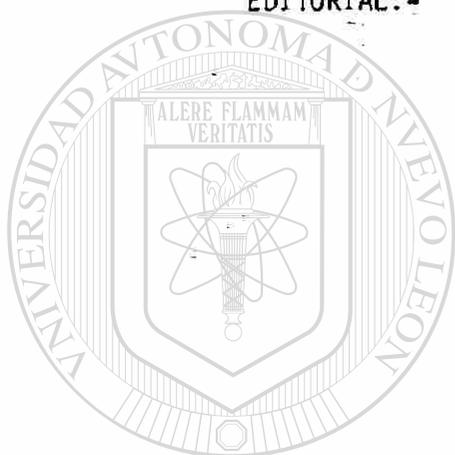
- 5 -

VII.- BIBLIOGRAFIA BASICA

LIBRO:- SISTEMAS OPERATIVOS
AUTOR:- STUART E MADNICK Y JOHN J DONOVAN
EDITORIAL:- DIANA

VIII.- BIBLIOGRAFIA COMPLEMENTARIA

LIBRO:- OPERATING SYSTEMS
AUTOR:- HARRY KATZAN Jr.
EDITORIAL:- VAN NOSTRAND REIHOLD CO

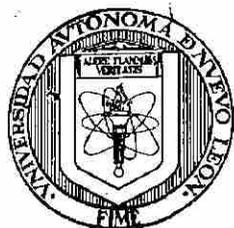


UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS





UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
Facultad de Ingeniería Mecánica y Eléctrica
Ciudad Universitaria
Monterrey, N. L., México

- 6 -

FORMULARON

ING. DAVID GARZA GARZA

David Garza Garza
FECHA 11 / ENE / 1985

REVISARON

ING. DAVID GARZA GARZA
JEFE DE ACADEMIA

David Garza Garza
FECHA 11 / ENE / 1985

ING. PEDRO MONTEMAYOR QUIROGA
JEFE DEL DEPARTAMENTO DE SISTEMAS

Pedro Montemayor Quiroga

AUTORIZO:-

ING. ELIAS TORRES GOMEZ
COORDINADOR DE CIENCIAS
DE LA ADMINISTRACION

Elias Torres Gomez

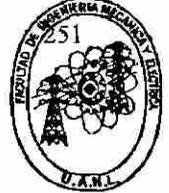
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS





UNIVERSIDAD AUTONOMA DE NUEVO LEON
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA



COORDINACION DE ADMINISTRACION Y SISTEMAS
DEPARTAMENTO DE SISTEMAS

NOMBRE DE LA MATERIA: SISTEMAS OPERATIVOS I
NOMBRE DEL PROGRAMA: SISTEMAS OPERATIVOS I
UBICACION: 6o. SEMESTRE PARA LA CARRERA DE IAS
REQUISITOS: ESTRUCTURA DE DATOS
ORGANIZACION COMPUTACIONAL
SESIONES TOTALES: 66 HRS-CLASE
FRECUENCIA: 5 HRS-CLASE/SEMANA

UNDAMENTO DE LA MATERIA:

La materia de sistemas operativos I tiene una importancia vital para los conocimientos de un ingeniero en sistemas, puesto que el sistema operativo es el programa necesario e indispensable para poner disponible el recurso computadora al usuario en cualquier tipo de medio ambiente, brindando las facilidades para administrar los recursos de la manera más adecuada.

DESCRIPCION DE LA MATERIA:

Esta clase maneja los conceptos acerca de los componentes principales de un sistema operativo.

OBJETIVOS GENERALES DE LA MATERIA:

El alumno deberá comprender y aplicar los conceptos principales de un sistema operativo tales como administración del procesador, administración de memoria etc.

TEMARIO:

- I.- INTRODUCCION Y CLASIFICACION DE SISTEMAS OPERATIVOS
- II.- COMPONENTES PRINCIPALES DE UN SISTEMA OPERATIVO
- III.- ADMINISTRACION DE PROCESOS
- IV.- ADMINISTRACION DEL PROCESADOR
- V.- ADMINISTRACION DEL ALMACENAMIENTO PRINCIPAL
- VI.- ADMINISTRACION DEL ALMACENAMIENTO

OBJETIVOS DE APRENDIZAJE

I.- INTRODUCCION Y CLASIFICACION DE SISTEMAS OPERATIVOS

OBJETIVO PARTICULAR.- Durante el desarrollo de esta unidad, se conocerán las bases, la evolución y los diferentes tipos de sistemas operativos para poder comprender sus funciones.

Para lograr el objetivo de la presente unidad, el alumno deberá comprender las funciones de los diferentes sistemas operativos.

Tiempo estimado de estudio para esta unidad, 8 horas-clase.

II.- COMPONENTES PRINCIPALES DE UN SISTEMA OPERATIVO

OBJETIVO PARTICULAR.- Durante el desarrollo de esta unidad, se tendrá un panorama donde se defina cada componente, su función y su interrelación con cada uno de los demás componentes.

Para lograr el objetivo de la presente unidad, el alumno deberá comprender los componentes tales como el despachador de tarífas y el monitor.

Tiempo estimado de estudio para esta unidad 10 horas-clase.

III.- ADMINISTRACION DE PROCESOS

OBJETIVO PARTICULAR.- Durante el desarrollo de esta unidad, se analizarán las principales técnicas para la administración del CPU y de los procesos en su secuencia de ejecución.

Para lograr el objetivo de la presente unidad, el alumno deberá comprender:

1. Que es un proceso, cuales son las transiciones de estados de un proceso y el procesamiento de interrupciones para procesos.
2. Los conceptos del manejo de varios procesos trabajando al mismo tiempo sobre los mismos recursos.

Tiempo estimado de estudio para esta unidad 12 horas-clase.

IV.- ADMINISTRACION DEL PROCESADOR

OBJETIVO PARTICULAR.- Durante el desarrollo de esta unidad, se analizarán las técnicas más comunes en la asignación y ejecución de tareas del procesador.

Para lograr el objetivo de la presente unidad, el alumno deberá comprender:

1. Los niveles, objetivos y criterios del Scheduler (programador de tareas), así como los algoritmos de Scheduling más usados.
2. Las metas de un sistema de multiprocesamiento, el paralelismo de proceso, la organización de hardware para multiprocesamientos y los sistemas operativos de multiprocesamiento.

Tiempo estimado de estudio para esta unidad 12 horas-clase.

V.- ADMINISTRACION DEL ALMACENAMIENTO PRINCIPAL

OBJETIVO PARTICULAR.- Durante el desarrollo de esta unidad, se analizarán las técnicas más comunes en las asignaciones y relocalizaciones de programas y datos en memoria principal.

Para lograr el objetivo de la presente unidad, el alumno deberá:

1. Entender las técnicas para el manejo de memoria real y su aplicación para sistemas operativos de un solo usuario o para sistemas de memoria virtual.
2. Comprender los conceptos de paginación, segmentación y memoria asociativa para entender el funcionamiento de la memoria virtual en un sistema operativo.

Tiempo estimado de estudio para esta unidad 12 horas-clase.

VI.- ADMINISTRACION DEL ALMACENAMIENTO

OBJETIVO PARTICULAR.- Durante el desarrollo de esta unidad, se analizarán las características físicas de las unidades de disco y las técnicas para administrar este recurso de almacenamiento.

Para lograr el objetivo de la presente unidad, el alumno deberá:

1. Comprender los motivos para generar políticas de Scheduling (programación de tareas) para los discos, las características más deseables de estas políticas y su optimización.

2. Entender el sistema de archivos, sus características, funciones y su aplicación, para utilizarlo como soporte de un sistema de base de datos.

Tiempo estimado de estudio para esta unidad 12 horas-clase.

METODOLOGIA

Cada semestre, antes de iniciar el curso, se reúne la academia a la que pertenecen todos los maestros que imparten la materia, y a su criterio se establece la forma de llevar el curso para la mejor obtención de los objetivos trazados en el programa.

CRITERIO DE EVALUACION:

Siguiendo los lineamientos, leyes y reglas de la U.A.N.L. y la F.I.M.E., se establece en las juntas de academia la mejor forma de evaluar el logro de los objetivos del programa.

BIBLIOGRAFIA

1.- INTRODUCCION A LOS SISTEMAS OPERATIVOS

HARVEY M. DIETEL

ADDISON-WESLEY

2.- SISTEMAS OPERATIVOS

TENENBAUM

PRETINCE-HALL

3.- LA POTENCIA DE PC/DOS

SIECHERT/WOOD

MC. GRAW HILL

Fecha de elaboración : NOVIEMBRE 1988

Responsable del programa: ING. VICTORIANO ALATORRE G.

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

Firma

DIRECCIÓN GENERAL DE BIBLIOTECAS



UNIVERSIDAD AUTONOMA DE NUEVO LEON
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA



COORDINACION DE ADMINISTRACION Y SISTEMAS
DEPARTAMENTO DE SISTEMAS

NOMBRE DE LA MATERIA: SISTEMAS OPERATIVOS II
NOMBRE DEL PROGRAMA: SISTEMAS OPERATIVOS II
UBICACION: 7o. SEMESTRE PARA LA CARRERA DE IAS
REQUISITOS: ESTRUCTURA DE DATOS
SISTEMAS OPERATIVOS I
SESIONES TOTALES: 40 HRS-CLASE
FRECUENCIA: 3 HRS-CLASE/SEMANA

FUNDAMENTO DE LA MATERIA:

La materia de sistemas operativos es muy extensa y por consiguiente se requiere de dos cursos, en este curso se cubrirán temas que no están incluidos en el programa anterior tales como: sistemas operativos para redes y casos prácticos.

DESCRIPCION DE LA MATERIA:

En esta clase se manejarán conceptos de sistemas operativos para una red, para P.C. y para mainframe.

OBJETIVOS GENERALES DE LA MATERIA:

El alumno aprenderá los conceptos principales de los sistemas operativos para una red local, así como casos prácticos de mainframe y P.C.

TEMARIO:

- I.- ASOCIACION DEL SISTEMA OPERATIVO CON OTROS SUBSISTEMAS
- II.- SISTEMAS OPERATIVOS DE REDES
- III.- GENERACION Y CONFIGURACION DEL SISTEMA OPERATIVO
- IV.- CASOS DE ESTUDIO
- V.- EVALUACION Y RENDIMIENTO DEL SISTEMA OPERATIVO
- VI.- COMPILADORES E INTERPRETADORES

OBJETIVOS DE APRENDIZAJE

I.- ASOCIACION DEL SISTEMA OPERATIVO CON OTROS SUBSISTEMAS

OBJETIVO PARTICULAR.- Durante el desarrollo de la presente unidad, se comprenderán las interfaces del sistema operativo con software adicional para sistemas, tales como sistemas de base de datos, sistemas de manejo de pantalla.

Para lograr el objetivo de esta unidad, el alumno deberá:

1. Comprender las utilerías necesarias para la creación, cooperación y mantenimiento de un sistema de base de datos.

Tiempo estimado de estudio para esta unidad, 7 horas-clase.

II.- SISTEMAS OPERATIVOS DE REDES

OBJETIVO PARTICULAR.- Durante el desarrollo de la presente unidad, se comprenderán las diferencias de un sistema operativo de una sola máquina a un sistema operativo de red.

Para lograr el objetivo de esta unidad, el alumno deberá:

1. Comprender los conceptos de una red local de computadoras personales.
2. Comprender los conceptos de una red de proceso distribuido.

Tiempo estimado de estudio para esta unidad, 7 horas-clase.

III.- GENERACION Y CONFIGURACION DEL SISTEMA OPERATIVO

OBJETIVO PATRICULAR.- Durante el desarrollo de la presente unidad, se comprenderá como ajustar los parámetros de un sistema operativo para una configuración de hardware especial.

Para lograr el objetivo de esta unidad, el alumno deberá:

1. Aprender como calcular el número de páginas o segmentos en la memoria principal.
2. Comprender como definir la configuración de almacenamiento secundario (área del sistema, área de swap y área de los usuarios).
3. Comprender como crear el sistema de cuentas de usuario y el sistema de seguridad.

Tiempo estimado de estudio para esta unidad, 7 horas-clase.

IV.- CASOS DE ESTUDIO

OBJETIVO PARTICULAR.- Durante el desarrollo de la presente unidad, se mostrarán medios diferentes de trabajo en sistema operativo para brindar una mejor perspectiva.

Para lograr el objetivo de esta unidad, el alumno deberá comprender los diferentes componentes y funciones para P.C. (MS-DOS), para red (Novell), Unix y VMS.

Tiempo estimado de estudio para esta unidad, 7 horas-clase.

V.- EVALUACION Y RENDIMIENTO DEL SISTEMA OPERATIVO

OBJETIVO PARTICULAR.- Durante el desarrollo de la presente unidad, se analizará y evaluará el rendimiento de los sistemas operativos para poder ajustar los parámetros necesarios para mantener el mejor rendimiento.

Para lograr el objetivo de esta unidad, el alumno deberá:

1. Aprender a evaluar el tiempo de respuesta de un sistema operativo.
2. Analizar y evaluar todas las estadísticas de un sistema operativo.
3. Entender la teoría de colas para manejar colas de procesos por lotes.

Tiempo estimado de estudio para esta unidad, 6 horas-clase.

VI.- COMPILADORES E INTERPRETADORES

OBJETIVO PARTICULAR.- Durante el desarrollo de la presente unidad, se comprenderá la estructura y funciones de un compilador ó interpretador.

Para lograr el objetivo de esta unidad, el alumno deberá:

256

1. Comprender los diferentes tipos de compiladores ó interpretadores.
2. Entender los subprocesos de una compilación (análisis de léxico, sintáxis y semántica).
3. Comprender la estructura y funciones de lenguajes de quinta generación.

Tiempo estimado de estudio para esta unidad, 6 horas-clase.

METODOLOGIA

- Cada semestre, antes de iniciar el curso, se reúne la academia a la que pertenecen todos los maestros que imparten la materia, y a su criterio se establece la forma de llevar el curso para la mejor obtención de los objetivos trazados en el programa.

CRITERIO DE EVALUACION:

Siguiendo los lineamientos, leyes y reglas de la U.A.N.L. y la F.I.M.E., se establece en las juntas de academia la mejor forma de evaluar el logro de los objetivos del programa.

BIBLIOGRAFIA

1.- INTRODUCCION A LOS SISTEMAS OPERATIVOS

HARVEY M. DIETEL

ADDISON-WESLEY

2.- SISTEMAS OPERATIVOS

TENENBAUM

PRENTICE-HALL

3.- LA POTENCIA DE PC/DOS

SIECHERT/WOOD

MC. GRAW-HILL

Fecha de elaboración : NOVIEMBRE DE 1988

Responsable del programa: ING. VICTORIANO ALATORRE GONZALEZ

Firma



DIRECCIÓN GENERAL DE BIBLIOTECAS



**UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERIA MECANICA Y ELECTRICA**



**COORDINACION DE ADMINISTRACION Y SISTEMAS
DEPARTAMENTO DE SISTEMAS**

NOMBRE DE LA MATERIA: SISTEMAS OPERATIVOS I
NOMBRE DEL PROGRAMA: SISTEMAS OPERATIVOS I
UBICACION: 6º. SEMESTRE DE LA CARRERA DE IAS
REQUISITOS: ESTRUCTURA DE DATOS
 ORGANIZACION COMPUTACIONAL
SESIONES TOTALES: 80 UNIDADES
FRECUENCIA: 5 UNIDADES/SEMANA

FUNDAMENTO DE LA MATERIA:

Tener conocimientos del funcionamiento, características principales y la evolución de los sistemas operativos es fundamental para el Ingeniero Administrador de Sistemas, debido a que estos proporcionan el medio sobre el cual se desarrollan y dá mantenimiento a sistemas computacionales.

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
DESCRIPCION DE LA MATERIA:

Debido a lo extenso e importancia de los sistemas operativos, se ha dividido el contenido de este tema en dos cursos. El primero, Sistemas Operativos I analiza la clasificación y características de los sistemas operativos, la administración de procesos, la administración de la memoria, los sistemas de archivo, entrada/salida y bloqueos. Concluyendo con casos de sistemas operativos de mayor aplicación en la actualidad.

OBJETIVOS GENERALES DE LA MATERIA:

Al finalizar el curso el alumno podrá reconocer los componentes y la administración de sistemas operativos centralizados o tradicionales.

TEMARIO:

- I. INTRODUCCION
- II. PROCESOS
- III. ADMINISTRACION DE LA MEMORIA
- IV. SISTEMAS DE ARCHIVOS
- V. ENTRADA/SALIDA
- VI. BLOQUEOS
- VII. CASOS DE ESTUDIO

OBJETTIVOS DE APRENDIZAJE**I.- INTRODUCCION**

OBJETIVO PARTICULAR.- Durante el desarrollo de esta unidad, se conocerán las bases, la evolución y los diferentes tipos de sistemas operativos, para así comprender las funciones, estructuras y componentes.

Para lograr el objetivo de la presente unidad, el alumno deberá comprender las funciones de los diferentes sistemas operativos y sus componentes.

Tiempo estimado de estudio, 7 unidades.

II.- PROCESOS

OBJETIVO PARTICULAR.- Durante el desarrollo de la presente unidad, se analizarán las principales técnicas para la administración del CPU y de los procesos en su secuencia de ejecución.

Para lograr el objetivo de la presente unidad, el alumno deberá comprender :

- 1.- Qué es un proceso, cuales son las transiciones de estado de un proceso, como se logra la comunicación entre estos y los problemas que enfrenta el S.O. para lograr la comunicación.
- 2.- Como decide el S.O. que procesos debe ejecutar cuando dos o más procesos son ejecutables lógicamente.

Tiempo estimado de estudio, 11 unidades.

III.- ADMINISTRACION DE LA MEMORIA.

OBJETIVO PARTICULAR.- Durante el desarrollo de esta unidad, se analizarán las técnicas más comunes en las asignaciones y relocalizaciones de programas y datos en memoria principal.

Para lograr el objetivo de la presente unidad, el alumno deberá :

1. Entender las técnicas para el manejo de memoria real y su aplicación para S.O. de un solo usuario o para sistemas de memoria virtual.
2. Comprender los conceptos de paginación, segmentación y memoria asociativa, para entender el funcionamiento de la memoria virtual en un S.O.

Tiempo estimado de estudio, 11 unidades.

IV.- SISTEMA DE ARCHIVO

OBJETIVO PARTICULAR.- Durante el desarrollo de la presente unidad, se analizará como almacenar una gran cantidad de información, que no se pierda hasta cuando el proceso que la generó termina su ejecución y que dos o más procesos tengan acceso concurrente a la información.

Para lograr el objetivo de la presente unidad, el alumno deberá comprender:

1. La forma de uso de los archivos y las propiedades que estos poseen.
2. La utilización de los directorios, su organización, sus propiedades y las operaciones que se pueden llevar a cabo con ellos.
3. La implantación de un sistema de archivo, la seguridad de la información y mecanismos de protección debe de tener un sistema de archivos.

Tiempo estimado de estudio, 11 unidades.

V.- ENTRADA/SALIDA.

OBJETIVO PARTICULAR.- Durante el desarrollo de la presente unidad, se analizará la forma de controlar los dispositivos de entrada/salida de la computadora.

Para lograr el objetivo de la presente unidad, el alumno deberá comprender:

1. Los principios del Hardware de E/S.
2. Los principios del Software de E/S.

Tiempo estimado de estudio , 11 unidades.

VI.-BLOQUEOS.

OBJETIVO PARTICULAR.- Durante el desarrollo de la presente unidad, se analizarán los bloqueos de los procesos.

Para lograr el objetivo de la presente unidad, el alumno deberá comprender:

1. ¿Que es bloqueo?
2. Detección del bloqueo.
3. Evación del bloqueo.
4. Prevención del bloqueo.

Tiempo estimado de estudio, 11 unidades.

VII.- CASOS DEL ESTUDIO.

OBJETIVO PARTICULAR.- Durante el desarrollo de la presente unidad, se analizarán las características de los S.O. de mayor aplicación actualmente.

Para lograr el objetivo de la presente unidad el alumno deberá comprender:

- | | |
|--|-------------------|
| 1. Las características principales del S.O. Unix. | (t.e. 5 unidades) |
| 2. Las características principales del S.O. Windows, NT. | (t.e. 5 unidades) |
| 3. Las características principales del S.O. D.O.S. | (t.e. 3 unidades) |
| 4. Las características principales del S.O. OS/2 | (t.e. 5 unidades) |

Tiempo estimado de estudio, 18 unidades

METODOLOGIA

Cada semestre, antes de iniciar el curso, se reúne la academia a la que pertenecen todos los maestros que imparten la materia, y a su criterio se establece la forma de llevar el curso para la mejor obtención de los objetivos trazados en el programa.

CRITERIO DE EVALUACION:

Siguiendo los lineamientos, leyes y reglas de la U.A.N.L. y la F.I.M.E., se establece en las juntas de academia la mejor forma de evaluar el logro de los objetivos del programa.

BIBLIOGRAFIA

1.- SISTEMAS OPERATIVOS MODERNOS

ANDREW TANENBAUM
PRENTICE HALL

2.- INTRODUCCION A LOS SISTEMAS OPERATIVOS

HARVEY M. DIETEL
ADDUSIB-WESLEY

Fecha de actualización: FEBRERO DE 1995

Responsable de actualización: ING. FERNANDO TREVIÑO ELIZONDO.

ING. ROSA ALICIA ELIZONDO CALLEJAS

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS

ING. FERNANDO TREVIÑO ELIZONDO

Jefe de la Academia de Organización Computacional.

Anexo 5

Evaluaciones y Productos de Aprendizaje

Evaluación de Diagnóstico

Esta evaluación se aplica con el fin de tener un panorama de inicio de actividades. Cada grupo es diferente y bajo esta premisa debemos evaluar el estado en el que vienen nuestros alumnos

Objetivo explícito.- Verificar el estado inicial del alumno, es decir con los conceptos que viene de las materias antecedentes.

Objetivo implícito.-

- Planear las estrategias a seguir durante el curso.
- Detectar carencias.
- Detectar confusiones, errores de buena fe, lagunas, etc.

El diagnóstico se puede lograr con una sola pregunta: ¿qué es un sistema operativo? Los alumnos manejan el término, lo han escuchado pero tienen diferente concepción de él.

Se anotan en el pizarrón las palabras más significativas o ideas principales y se comienzan a enlazar conceptos hasta llegar a una definición. Con ello no solo me entero de su estado inicial también les doy el primer concepto de la clase.

Esta misma técnica la utilizo para evaluación formativa en algunos temas que se prestan para esto mismo. Por supuesto que también se puede implementar un examen sencillo de cinco preguntas con el consecuente comentario del valor de este instrumento.

Evaluación Formativa

Objetivo explícito.- Verificar el grado de avance del alumno con respecto a un tema o unidad.

Objetivo implícito.-

- Planear las estrategias a seguir según se vea que haya habido aprendizajes significativos, el dominio de un concepto etc.
- Detectar carencias, avances y retrocesos.
- Detectar confusiones, errores de buena fe, lagunas, etc.
- Revirar en caso de error por parte del maestro.

Esto se logra con exámenes “sorpresa” al final de la unidad o bien cuando se ha pasado por un tema “escabroso” o difícil. Se precisan parámetros que esta medición puede dar, en un principio cuando se les dice: “Saquen una hoja y pónganle su nombre” suele causar sorpresa o confusión pero cuando se les dice que es solo para medir si están aprendiendo o no y que carece de valor para su calificación, se relajan y puede encontrarse la medición buscada. La pregunta principal: “Qué he entendido hasta ahora de este tema”.

Evaluación Sumativa

En un primer término podemos hacer uso de los exámenes, los exámenes que se utilizan son opción múltiple, falso o verdadero, subrayar la respuesta correcta; leer un párrafo, encontrar y señalar el o los conceptos implícitos en él, la redacción de una acción en la que se vea involucrado uno o varios conceptos de los que se les proporciona una lista. Lo anterior con el fin de conceptualicen, redacten, cuiden su ortografía, etc; en un plano tal vez más elevado quisiera poder hacer que se apropiaran del conocimiento mediante el aprendizaje significativo.

Se ha hablado mucho de los productos de aprendizaje como una alternativa para la evaluación, la acreditación y la calificación, ahora tenemos la fundamentación para implementarlos o incorporarlos dentro de nuestra práctica docente. Como productos de aprendizaje tenemos:

Trabajos de Investigación por Equipo

Siguiendo la tónica del trabajo por equipo y con el fin de fomentar la integración al medio en el cual se desenvolverán se les encarga una investigación por equipo. Invariablemente se les solicita información sobre redes de computadora y todo lo que esto involucra. Este trabajo se les solicita con el fin de que tengan la información actualizada y se les pide que la profundidad del tema la van a desarrollar tanto como quieran estar preparados para los siguientes semestres en las que se convierte en el tema de la clase.

Se les pide un reporte del trabajo por equipo y uno para el maestro y en él se puede observar limpieza, presentación, orden etc.

Ellos mismos eligen su organización, preparación de materiales, visitas a distribuidores, entrevistas a usuarios y rol de participación. En este tipo de trabajos se pueden observar su sentido de responsabilidad, su puntualidad, cooperatividad, desarrollo de interés, desarrollo de temas, iniciativas, habilidades de análisis, de creatividad y de pensamiento.

Trabajos Finales

Los trabajos finales que se les encarga utilizan más de un concepto de los que se ven en clase y de esta manera se puede saber si hubo aprendizaje o no. Se batalla un poco con las "copias", pero finalmente al individualizar el trabajo final se ha podido eliminar ese inconveniente. Cada semestre se encarga un trabajo diferente, si son 40 alumnos son 40 proyectos diferentes, distinto enfoque o distinto tema.

Este tipo de trabajos pueden reflejar si se maneja el concepto, la creatividad del alumno, sus capacidades que como individuo tiene, las que puede desarrollar y algunos otros factores.

Resumen Autobiográfico

Se desea obtener el grado de Maestro en Ciencias de la Administración con Especialidad en Relaciones Industriales debido a que una de las aplicaciones “fuertes” de la computación es precisamente la administración y el principal campo de acción es precisamente en la industria por lo que se consideró adecuado el giro que se dio al estudiar esta maestría. Un maestro es un administrador del conocimiento y los alumnos a quienes se les brinda atención al final de sus carreras trabajaran para la industria precisamente.

Se ha pertenecido a la “National Council of Teachers of Mathematics” en lo Estados Unidos de América la cual se dedica a la investigación e implementación de métodos de enseñanza de las matemáticas y sus aplicaciones desde el pre-kinder hasta el grado doce que corresponde al termino de la “High School”, entre los métodos está también el desarrollo de software para mejorar el proceso enseñanza-aprendizaje. Se mantuvo una comunicación con el Instituto de Educación de la Universidad de Londres, para intercambio de ideas y soluciones a problemas específicos, esto sucedió por 1988.

Ariadne Beatriz Sánchez Ruiz, nació para el mundo de la computación en 1978 al principiar sus estudios en la carrera de Licenciado en Ciencias Computacionales en la Facultad de Ciencias Físico - Matemáticas de la Universidad Autónoma de Nuevo León y pasó al ejercicio de la profesión el 1982 cuando comenzó a laborar en el Departamento de Informática de la Universidad Autónoma de Nuevo León como practicante, lo que le fue tomado en cuenta para proporcionarle su planta como trabajador universitario, en un principio como programador colaborando en el proyecto Geigy, la base de datos que estaba conectada a Palo Alto California y posteriormente promovida a analizar, traducir manuales de los sistemas y de equipos con el fin de hacerlos entendibles para los demás usuarios. En 1984 participó en el análisis del “Sistema Sínodo” en la Facultad de Odontología, regresando nueve meses después a la Administración Central, participando activamente en la auditoría

de sistemas hasta 1987; posteriormente realizó el análisis correspondiente del sistema que se había implantado en la Facultad de Ingeniería Civil por espacio de 30 días. En 1988 se unió al grupo de trabajo de la Facultad de Arquitectura en la que estuvo encargada del Laboratorio de Computación, compra de equipo, asistente del jefe del Departamento de Informática, supervisora y maestra de los cursos de verano infantil. En 1990 prestó sus servicios en el Post-grado de Ortodoncia de la Facultad de Odontología en la que también fungió como encargada del área computacional dando clases a los alumnos del post-grado y apoyando en la elaboración de material didáctico asistido por computadora; asistió en los proyectos "Génesis", cefalometría por medio de computadora y algunos proyectos menores. En 1991 se unió a la FIME en donde comenzó como operador del sistema Escolar de la misma, combinando esto con la impartición de cátedra a nivel licenciatura participando en el análisis para la implantación de un sistema escolar para post-grado, elaboración de la página de Internet del mismo y después como encargada de las salas de cómputo del edificio 9.

En 1997 publicó su primer manual de apuntes para Sistemas Operativos I y en 1998 la revisión del mismo. Actualmente estudia material para elaborar material didáctico por medio de la realidad virtual.

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

DIRECCIÓN GENERAL DE BIBLIOTECAS



