

Temas Generales para la preparación de la Oposición al Cuerpo Superior de Sistemas y Tecnologías de la Información de la Administración del Estado.

**Cuerpo Superior de Estadísticos del Estado
Especialidad de Estadística-Ciencia de Datos.**

Almacenamiento y modelos de datos

Tema 3. Sistemas operativos

AUTOR: Alexandra Vlad

Creación: Junio 2021

ÍNDICE

1	INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS	3
2	QUÉ HACEN LOS SISTEMAS OPERATIVOS	4
3	ORGANIZACIÓN DE SISTEMAS DE COMPUTACIÓN.....	6
4	ARQUITECTURA DE SISTEMAS DE COMPUTACIÓN	8
5	ESTRUCTURA DE SISTEMAS OPERATIVOS	11
6	OPERACIONES EN SISTEMAS OPERATIVOS.....	13
7	GESTIÓN DE PROCESOS	14
8	GESTIÓN DE MEMORIA.....	18
9	GESTIÓN DE ALMACENAMIENTO (SISTEMA DE ARCHIVOS)	26
10	PROTECCIÓN Y SEGURIDAD.....	30
11	SISTEMAS DISTRIBUIDOS. ENTORNOS DE COMPUTACIÓN	34
12	RESUMEN ESQUEMÁTICO	36
13	BIBLIOGRAFÍA BÁSICA.....	38

1 Introducción a los sistemas operativos

El hardware de un ordenador es, en potencia, una máquina capaz de llevar a cabo multitud de tareas útiles, pero por sí sola no resulta de gran ayuda, si no se cuenta con algún mecanismo que permita indicarle cómo llevar a cabo las acciones que deseamos realice. Es, por tanto, el software del ordenador el que convierte al hardware en algo realmente útil. El software más importante es el *sistema operativo*, cuya misión es la de controlar todo el ordenador. El control de todos los aspectos hardware de un ordenador resulta muy complicado, piénsese en la escritura de un dato en un disco, y la enorme complejidad que conlleva una operación tan simple. Si el programador tuviera que tener en cuenta todos los aspectos relacionados con el control del hardware, es muy probable que su trabajo resultara imposible. Por ello, la primera función del sistema operativo es la de crear una máquina virtual, de manera que el programador tenga una visión simplificada del computador, en la cual las particularidades hardware le sean transparentes.

Cualquier sistema informático se organiza en capas, de forma tal que en la capa más profunda se sitúa el hardware y en la capa superior el usuario. Esta estructura se puede observar en la siguiente figura.

Juegos	Hoja de cálculo	Base de datos
Compiladores		Intérprete de comandos
SISTEMA OPERATIVO		
Lenguaje máquina		
Microprogramación		
Dispositivos físicos		

Sobre la máquina física se sitúa la microprogramación. Un microprograma es un software muy primitivo que constituye un intérprete entre el hardware y el lenguaje máquina. Así, el microprograma es el encargado de ejecutar instrucciones de código máquina, tales como ADD (suma de dos operandos), o JUMP (salto en el flujo secuencial del programa). Por ejemplo, si se ha de llevar a cabo una orden MULT (multiplicación de dos operandos), el microprograma determina donde están almacenados los factores, programa la Unidad Aritmético Lógica (ALU) para que realice la operación y guarda el resultado. El conjunto de instrucciones que puede interpretar el microprograma constituye el código máquina del procesador.

El lenguaje máquina está constituido por algunos cientos de instrucciones, que realizan operaciones elementales aritméticas, lógicas, de movimiento de la información y de control de flujo de ejecución. En este nivel se puede empezar a hablar de programación de los dispositivos físicos, mediante la carga de valores en registros de dispositivos especiales. Por ejemplo, para ordenar a un disco que realice una lectura hay que almacenar en sus registros los parámetros necesarios para llevar a cabo la operación. Estas tres primeras capas constituyen el hardware del ordenador. La primera misión del sistema operativo es la de ocultar toda esta complejidad al usuario.

Sobre el sistema operativo se sitúan los programas típicos del sistema, como son compiladores, editores, intérpretes de órdenes, etc. Estas utilidades son programas del sistema, pero no constituyen parte del sistema operativo. La diferencia fundamental entre estos programas y el sistema operativo, es que este último se ejecuta en modo central, modo supervisor o modo núcleo, y está protegido mediante hardware contra la alteración por parte de los usuarios. Esta diferencia es sutil, pero muy importante, un usuario puede escribir su propio intérprete de comandos, pero no puede modificar el controlador de disco.

En la capa más superior se sitúan los programas de aplicación que dan servicio al usuario no programador.

2 Qué hacen los sistemas operativos

Como hemos visto anteriormente, un sistema operativo es un programa que actúa como intermediario entre el usuario y el hardware de un computador y su propósito es proporcionar un entorno en el cual el usuario pueda ejecutar programas. El objetivo principal de un sistema operativo es lograr que el sistema de computación se use de manera cómoda, y el objetivo secundario es que el hardware del computador se emplee de manera eficiente.

Un sistema Operativo (SO) es en sí mismo un programa de computadora. Sin embargo, es un programa muy especial, quizá el más complejo e importante en una computadora. El SO despierta a la computadora y hace que reconozca a la CPU, la memoria, el teclado, el sistema de vídeo y las unidades de disco. Además, proporciona la facilidad para que los usuarios se comuniquen con la computadora y sirve de plataforma a partir de la cual se corran programas de aplicación.

Cuando enciendes una computadora, lo primero que ésta hace es llevar a cabo un autodiagnóstico llamado auto prueba de encendido (Power On Self Test, POST). Durante la POST, la computadora identifica su memoria, sus discos, su teclado, su sistema de vídeo y cualquier otro dispositivo conectado a ella. Lo siguiente que la computadora hace es buscar un SO para arrancar (boot).

Una vez que la computadora ha puesto en marcha su SO, mantiene al menos parte de éste en su memoria en todo momento. Mientras la computadora esté encendida, el SO tiene 4 tareas principales:

- Proporcionar ya sea una interfaz de línea de comando o una interfaz gráfica al usuario, para que este último se pueda comunicar con la computadora. Interfaz de línea de comando: tú introduces palabras y símbolos desde el teclado de la computadora, ejemplo, el MS-DOS. Interfaz gráfica del Usuario (GUI), seleccionas las acciones mediante el uso de un Mouse para pulsar sobre figuras llamadas iconos o seleccionar opciones de los menús.
- Administrar los dispositivos de hardware en la computadora. Cuando corren los programas, necesitan utilizar la memoria, el monitor, las unidades de disco, los puertos de Entrada/Salida (impresoras, módems, etc.). El SO sirve de intermediario entre los programas y el hardware.
- Administrar y mantener los sistemas de archivo de disco. Los SO agrupan la información dentro de compartimientos lógicos para almacenarlos en el disco. Estos grupos de información son llamados archivos. Los archivos pueden contener instrucciones de programas o información creada por el usuario. El SO mantiene una lista de los archivos en un disco, y nos proporciona las herramientas necesarias para organizar y manipular estos archivos.
- Apoyar a otros programas. Otra de las funciones importantes del SO es proporcionar servicios a otros programas. Estos servicios son similares a aquellos que el SO proporciona directamente a los usuarios. Por ejemplo, listar los archivos, grabarlos a disco, eliminar archivos, revisar espacio disponible, etc. Cuando los programadores escriben programas de computadora, incluyen en sus programas instrucciones que solicitan los servicios del SO. Estas instrucciones son conocidas como "llamadas del sistema"

Un sistema operativo es el programa que oculta la verdad del hardware al programador y presenta una vista simple y agradable de los archivos nominados que pueden leerse y escribirse. El sistema operativo resguarda al programador del hardware del disco y presenta una interfaz simple orientada al archivo, también disimula mucho del trabajo concerniente a interrupciones, relojes o cronómetros, manejo de memoria y otras características de bajo nivel.

La función del sistema operativo es la de presentar al usuario con el equivalente de una máquina ampliada o máquina virtual que sea más fácil de programar que el hardware implícito.

Un sistema operativo es una parte importante de casi cualquier sistema de computación. Un sistema de computación puede dividirse en cuatro componentes: el hardware, el sistema operativo, los programas de aplicación y los usuarios.

El hardware (unidad central de procesamiento (UCP), memoria y dispositivos de entrada y salida (E/S)) proporciona los recursos de computación básicos. Los programas de aplicación (compiladores,

sistemas de bases de datos, juegos de video y programas para negocios) definen la forma en que estos recursos se emplean para resolver los problemas de computación de los usuarios. Puede haber distintos usuarios (personas, máquinas, otros computadores) que intentan resolver problemas diferentes; por lo tanto, es posible que haya diferentes programas de aplicación. El sistema operativo controla y coordina el uso del hardware entre los diversos programas de aplicación de los distintos usuarios.

Podemos ver al sistema operativo como un asignador de recursos. Un sistema de computación tiene muchos recursos (hardware y software) que pueden requerirse para resolver un problema: tiempo de la UCP, espacio de memoria, espacio de almacenamiento de archivos, dispositivos de E/S, etc. El sistema operativo actúa como el administrador de estos recursos y los asigna a usuarios y programas concretos según los necesiten las tareas de los usuarios.

Puesto que pueden surgir conflictos en las solicitudes de recursos, el sistema operativo debe decidir a qué solicitudes se les asignaran para que el sistema de computación pueda funcionar de manera eficiente y justa.

En términos generales no hay una definición de sistema operativo completamente adecuada. Los sistemas operativos existen porque son una manera razonable de solucionar el problema de crear un sistema de computación utilizable.

El objetivo fundamental de los sistemas de computación es ejecutar los programas de los usuarios y facilitar la resolución de sus problemas. El hardware se construye con este fin, pero como este no es fácil de utilizar, se desarrollan programas de aplicación que requieren ciertas operaciones comunes, como el control de dispositivos de E/S. las funciones comunes de control y de asignación de recursos se integran para formar un solo fragmento de software: el sistema operativo.

3 Organización de sistemas de computación

Aunque la organización de los sistemas operativos varía mucho entre los diferentes sistemas existen una serie de componentes, más o menos estándar, que son comunes.

La magnitud del sistema operativo hace que la forma más usual de diseñarlo sea mediante la creación de módulos. Estos módulos son independientes entre sí, en la medida de lo posible, y se encargan de gestionar cada una de las funciones del sistema operativo. A continuación, vamos a describir esta estructura de módulos.

Administrador de procesos

El elemento constitutivo fundamental de cualquier sistema informático es el proceso. Un proceso es fundamentalmente un programa en ejecución. Un sistema se puede considerar como un conjunto de procesos ejecutando de forma concurrente, algunos de estos procesos son procesos de usuario y otros son procesos del mismo sistema operativo. La gestión de la ejecución concurrente es quizás el problema más importante al que se enfrenta el sistema operativo.

Las actividades que ha de desarrollar el administrador de procesos incluyen todas aquellas necesarias para el control de todos los programas en ejecución dentro del sistema. Las más importantes son:

- Crear y eliminar los procesos de usuario y del sistema.
- Mantener las estructuras necesarias para la caracterización de los procesos. En especial, el mantenimiento de la tabla de procesos, donde se almacena la información referente a todos los procesos que hay en el sistema.
- Suspender, apropiar y reanudar los procesos.
- Proporcionar los mecanismos necesarios para la sincronización de procesos y definir la política de reparto del tiempo de ejecución.
- Proporcionar los mecanismos para la comunicación entre los procesos.
- Proporcionar los mecanismos para evitar, o manejar, los bloqueos mutuos.

Administrador de la memoria principal

La ejecución concurrente de procesos origina la necesidad de repartir la memoria entre ellos. La forma más usual es la de crear un espacio de direcciones virtuales en el cual la memoria es mucho mayor que la capacidad física real. Este espacio de direcciones virtuales lo ha de controlar el sistema operativo teniendo en cuenta el tamaño real de la memoria. La administración ha de incluir también una política de reparto. En cualquier caso, la gestión de la memoria principal siempre necesitará apoyo del hardware para llevarse a cabo.

Las actividades de control de almacenamiento más importantes que hay que realizar son las siguientes:

- Controlar la relación entre el espacio de direcciones virtuales y el almacenamiento real.
- Controlar que zonas de memoria se están usando y que proceso las usa, así como proteger las zonas de memoria de cada proceso de posibles accesos de otros procesos.
- Decidir que procesos serán cargados en memoria cuando quede espacio disponible.
- Asignar espacio para satisfacer los requerimientos de las aplicaciones y recuperar el espacio que quede libre tras la terminación de un proceso o mediante la apropiación.

Administrador del almacenamiento secundario

El enorme flujo de información que ha de gestionar un sistema operativo moderno hace que la gestión del almacenamiento secundario sea fundamental para el rendimiento del sistema. La mayoría de las aplicaciones y datos han de permanecer en almacenamiento secundario hasta que su presencia sea requerida en el almacenamiento principal. Este módulo se encarga de la administración física del almacenamiento, dejando la mayoría de los aspectos de administración lógica al administrador de archivos. Las tareas fundamentales que se han de cumplir son:

- Administración del espacio libre, y de la asignación de espacio que será requerida por los procesos.
- Administración de las operaciones sobre disco, como la secuenciación de las peticiones de acceso, para optimizar el tiempo de respuesta.

Administrador del sistema de E/S

El administrador de E/S hace referencia a la creación de una máquina extendida. La misión de este módulo es la de ocultar todo lo relacionado con las particularidades hardware de la E/S. Sus funciones fundamentales son:

Gestión de un sistema de memoria temporal intermedia, que cree un sistema de cache con los dispositivos de E/S.

- Creación de un interfaz general con los controladores de dispositivos, para un acceso uniforme.
- Creación de los controladores específicos para cada periférico.

La idea fundamental es la de que únicamente cada manejador de dispositivo conozca las particularidades del dispositivo que controla.

Administrador de archivos

Este es uno de los componentes más visibles del sistema operativo. Este módulo se ocupa de la gestión de almacenamiento a nivel lógico. El sistema operativo proporciona una visión lógica uniforme de almacenamiento, el archivo, que ignora todos los aspectos físicos del almacenamiento. El usuario ignora cómo y dónde se almacena un archivo que para él es una secuencia de bytes que mantienen información relacionada. Además, el sistema operativo presenta una estructura de directorios en forma de árbol para facilitar la organización del almacenamiento al usuario. La forma en que se organiza lógicamente el almacenamiento es muy importante, de ella dependen dos aspectos fundamentales: el aprovechamiento satisfactorio del espacio disponible y la rapidez en el acceso a la información.

Las funciones principales que ha de realizar el administrador de archivos son:

- Creación y eliminación de archivos
- Creación y eliminación de directorios.
- Proporcionar las operaciones primitivas necesarias para la manipulación de archivos y directorios.
- Relación entre la organización lógica y la organización física de los dispositivos de almacenamiento. Gestión del espacio libre.
- Gestión de la seguridad del sistema de archivos, y de la protección tanto frente a otros usuarios como a fallos del sistema.

Sistema de protección

La protección de todos los recursos de un proceso (memoria, archivos, cpu, etc.), hace referencia a dos aspectos. Primero ha de proteger los recursos asignados a un proceso, evitando accesos a sus segmentos de memoria, o escrituras en zonas del disco que él tenga en uso. Además ha de evitar que los errores en un subsistema que está funcionando de forma incorrecta afecten al resto del sistema.

Administrador de redes

Los equipos modernos rara vez realizan su trabajo en solitario. La mayoría funcionan dentro de redes locales, que a su vez se comunican con redes globales de información. Es necesario que el sistema operativo proporcione los mecanismos necesarios para que los procesos puedan comunicarse entre sí. En el caso límite, el sistema distribuido, el propio sistema operativo se encuentra dentro de la red. La misión fundamental de este módulo es el control de flujo de información dentro de la red en lo que afecta al propio sistema.

4 Arquitectura de sistemas de computación

Actualmente, los sistemas operativos se organizan en capas.

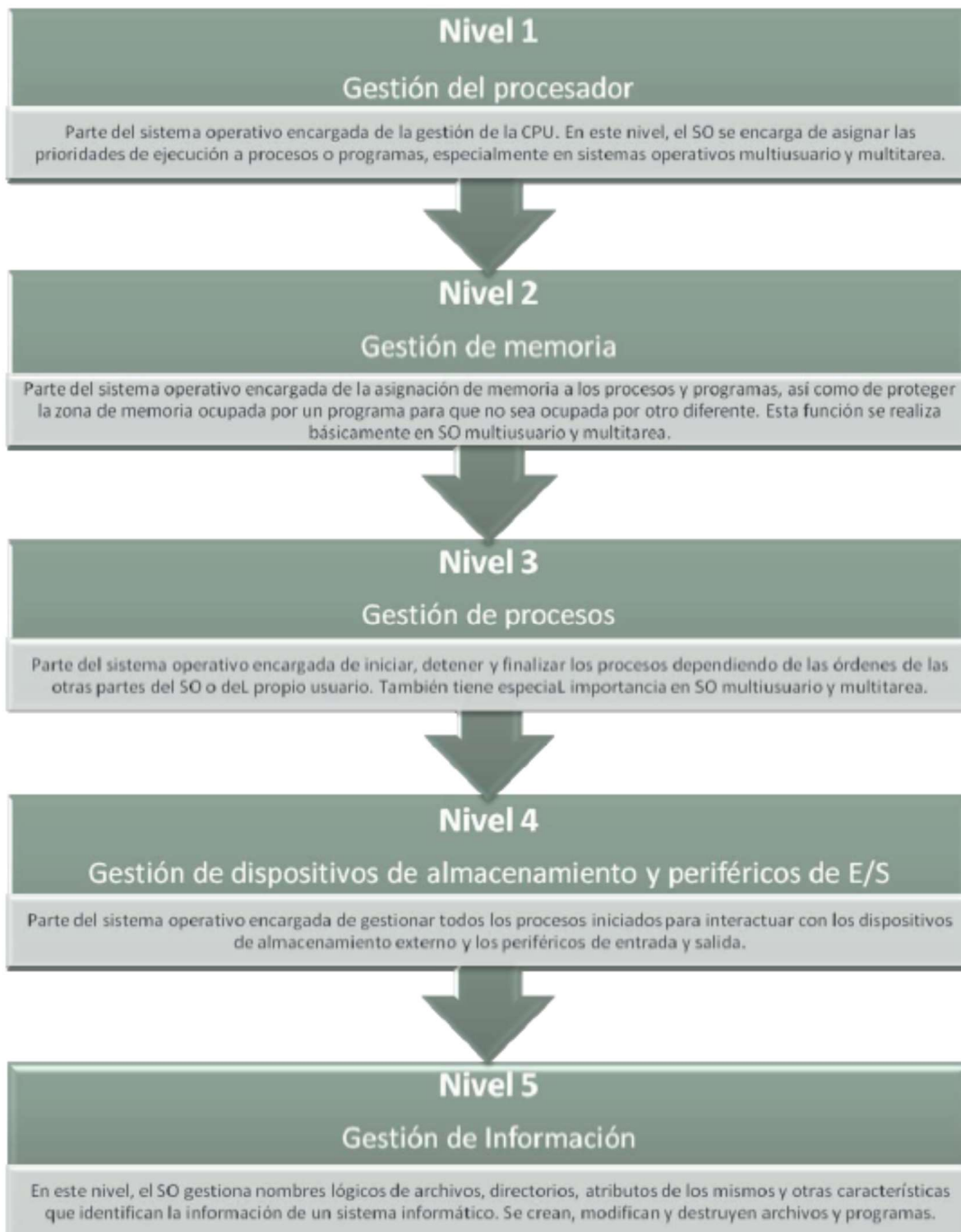
Si se tiene en cuenta las funciones que un sistema operativo puede realizar tenemos lo siguiente:

- Control y ejecución de programas.
- Control, gestión y administración de periféricos.
- Control, gestión y administración de usuarios.
- Control de procesos.
- Control de errores de sistema y de aplicaciones.
- Control y gestión de seguridad frente a intrusos o virus.
- Etcétera.

Un sistema operativo se estructura, básicamente, en cinco niveles o capas. Cada uno de estos niveles se comunica con el nivel inmediatamente inferior y superior de tal forma que todos ellos están coordinados y consiguen el objetivo del sistema operativo.

El sistema operativo está compuesto por un conjunto de paquetes de software que pueden utilizarse para gestionar las interacciones con el hardware. Estos elementos se incluyen por lo general en este conjunto de software:

- **El núcleo**, que representa las funciones básicas del sistema operativo, como, por ejemplo, la gestión de la memoria, de los procesos, de los archivos, de las entradas/salidas principales y de las funciones de comunicación.
- **El intérprete de comandos**, que posibilita la comunicación con el sistema operativo a través de un lenguaje de control, permitiendo al usuario controlar los periféricos sin conocer las características del hardware utilizado, la gestión de las direcciones físicas, etcétera.
- **El sistema de archivos**, que permite que los archivos se registren en una estructura arbórea.



Otra posible clasificación de los niveles del SO es la siguiente:

Nivel	Descripción
3	Usuario. Se encarga de controlar los procesos a nivel usuario, es decir, muestra al usuario el proceso que está ejecutándose o el que desea ejecutar.
2	Supervisor. Se encarga de realizar la comunicación de cada proceso entre el sistema y el usuario, controla y coordina la gestión de entrada y salida de los diferentes procesos hacia los periféricos y los dispositivos de almacenamiento externo.

1	Ejecutivo. Se encarga de administrar la memoria, almacenar los procesos en páginas (segmentación, paginación, almacenamiento en búfer, etc.), tanto en memoria principal como en disco.
0	Núcleo. Se encarga de controlar la CPU, de gestionar que procesos llegan a la CPU para ser ejecutados (planificación de procesos), de realizar tareas básicas de sistema, de la comunicación con el hardware, etc.

En la actualidad la mayoría de los sistemas operativos son multiusuario y multitarea, es decir, pueden ejecutar varios procesos a la vez. Para ello, el núcleo de estos sistemas operativos gestiona las denominadas máquinas virtuales.

Una máquina virtual es un archivo, o conjunto de archivos que es una copia exacta del hardware real que gestiona el propio sistema operativo. Dispone de los niveles de núcleo, ejecutivo, supervisor y usuario, como si fueran independientes, para cada usuario que utilice el sistema.

Este tipo de gestión, que realizan la mayoría de los sistemas operativos multiusuario y multitarea, confiere gran estabilidad al sistema, ya que ante eventuales errores de la aplicación, el hardware no se bloquea; la aplicación del usuario y su correspondiente copia del hardware, es la que deja de ejecutarse. Por tanto, el resto de usuarios podrán seguir ejecutando sus programas sin más problemas. Es decir, el bloqueo de un proceso que puede estar ejecutando un usuario no afecta al resto de procesos, ya que cada uno de ellos, incluso para el mismo usuario, se ejecuta en una máquina virtual diferente, por lo que el hardware de la máquina no se ve afectado.

Una vez que el usuario termina de ejecutar su aplicación, se produce la interacción real con el hardware. El sistema operativo termina de ejecutar correctamente la aplicación, el resto de usuarios siguen trabajando con sus máquinas virtuales y, evidentemente, el hardware funciona sin problemas.

Se descarga el contenido de los archivos que forman la máquina virtual al hardware, momento en que se produce la operación real de entrada y salida sobre el periférico o dispositivo de almacenamiento.

5 Estructura de sistemas operativos

A continuación, se describen algunas de las posibles estructuras de los SO. Hay que señalar que algunas de éstas son producto de desarrollo de los SO a lo largo de la historia y que, si bien no deberían seguir usándose debido a sus muchas carencias, se siguen manteniendo por compatibilidad con sistemas ya existentes.

Una posible clasificación de los sistemas operativos podría ser por estructura. De esta forma tendríamos sistemas monolíticos, sistema en capas, sistema operativo como máquina virtual y modelo cliente-servidor (o sistema operativo tipo microkernel).

Sistemas monolíticos

Este tipo de organización, aunque es la peor, es la más extendida. La estructura consiste en la total ausencia de estructura. El código se organiza como una serie de procedimientos que resuelven tareas específicas. Los procedimientos se llaman unos a otros, dando lugar a lo que se conoce como *código espagueti*. No existe ocultación de la información, cada procedimiento es visible para todos los que forman el sistema operativo. Las desventajas de este tipo de diseño son obvias: dificultad para la depuración y el mantenimiento, gran cantidad de errores, dificultad de actualización y todas las relacionadas con cualquier tipo de programa no estructurado. Aunque no existe una estructuración global definida, el sistema operativo monolítico se suele organizar en tres niveles, más o menos diferenciados:

- Un programa principal que atiende las llamadas de los programas de usuario e invoca el procedimiento de servicio del SO solicitado.
- Un conjunto de procedimientos de servicio que llevan a cabo las llamadas al sistema. Existe un procedimiento de servicio para cada llamada al sistema.
- Un conjunto de procedimientos de más bajo nivel, que ayudan a la ejecución de los procedimientos de servicio. Estos procedimientos son invocados por diferentes rutinas de servicio.

Sistemas en capas (jerárquica)

El primer SO construido con esta filosofía fue THE (Technische Hogeschool Eindhoven) que constaba de 6 capas:

- Capa 0. Se ocupa de la asignación de procesos y del intercambio de procesos cada vez que los temporizadores lo indican. Esta capa contiene todo lo referente a la multiprogramación, de forma tal que el resto de capas no tienen que ocuparse de todo lo referente al hecho de que hay varios procesos ejecutándose a la vez.
- Capa1. Gestiona todo lo referente a la asignación de memoria. Es la encargada de proporcionar la memoria necesaria a cada proceso y de realizar los intercambios que sean necesarios con la memoria secundaria. Las capas superiores no deben preocuparse de donde se encuentra su memoria, ni de gestionar la asignación.
- Capa 2. Comunica cada proceso con la consola del operador. Por encima de esta capa cada proceso tiene su propia consola de operador.
- Capa 3. Controla los dispositivos de entrada/salida (E/S) y se encarga de todas las peculiaridades de su programación, así como mantener el flujo de información con ellos, mediante áreas de memoria temporal de intercambio. Las capas superiores ven a los dispositivos de E/S como entes abstractos, aislándose de sus problemas físicos específicos.
- Capa 4. Programas de usuario. Estos se ven aislados de los problemas de multiprogramación, gestión de memoria, de la consola y de la E/S.
- Capa 5. En la parte superior de la pirámide se sitúa el proceso operador del sistema.

Máquina Virtual

El sistema de máquina virtual proporciona la multiprogramación, pero eliminando la creación de una máquina extendida. Para ello, crea un monitor de la máquina virtual que se ejecuta directamente sobre

el HW. Este monitor posibilita la multiprogramación mediante la creación de máquinas virtuales que reproducen el HW del ordenador. Sobre cada una de estas máquinas virtuales cada usuario puede ejecutar su propio SO. Cuando un usuario realiza una llamada al SO, ésta es capturada por su propio SO. El monitor de la máquina virtual proporciona a ese SO las instrucciones normales del manejo del sistema y, finalmente, es el monitor el que lleva a cabo las acciones sobre la máquina real. De esta forma, cada usuario gana en flexibilidad.

Modelo cliente-servidor (o microkernel)

La tendencia en los SO actuales es eliminar todo el código del SO que sea posible trasladándolo al usuario, manteniendo un núcleo mínimo (microkernel). Así la mayoría de las funciones se trasladan a procesos en modo usuario, ejecutándose el SO solo cuando es necesario. La estructura es la siguiente: cuando un proceso necesita un servicio (proceso cliente) envía la solicitud a un proceso servidor que realiza el trabajo por él y devuelve la respuesta. En este esquema el SO solo se encarga de la comunicación entre el cliente y el servidor. Como se han aislado las diferentes facetas del SO, el fallo en uno de los módulos servidores no implica fallo total del sistema. Sin embargo, existen acciones que no pueden llevarse a cabo por un proceso en modo usuario, por ello es necesario arbitrar algún mecanismo para que los servidores puedan llevar a cabo estas acciones. Para ello existen varias políticas:

- Una idea es la de hacer que aquellos servidores críticos, por ejemplo, acceso a los dispositivos de E/S, se ejecuten en modo núcleo.
- Otra solución es la de crear una serie de mecanismos dentro del núcleo que lleven a cabo las tareas más críticas.

6 Operaciones en sistemas operativos

A continuación, se enumeran las funciones de los sistemas operativos:

- Aceptar todos los trabajos y conservarlos hasta su finalización.
- Interpretación de comandos: Interpreta los comandos que permiten al usuario comunicarse con el ordenador.
- Control de recursos: Coordina y manipula el hardware de la computadora, como la memoria, las impresoras, las unidades de disco, el teclado o el Mouse.
- Manejo de dispositivos de E/S: Organiza los archivos en diversos dispositivos de almacenamiento, como discos flexibles, discos duros, discos compactos o cintas magnéticas.
- Manejo de errores: Gestiona los errores de hardware y la pérdida de datos.
- Secuencia de tareas: El sistema operativo debe administrar la manera en que se reparten los procesos. Definir el orden. (Quien va primero y quien después).
- Protección: Evitar que las acciones de un usuario afecten el trabajo que está realizando otro usuario.
- Multiacceso: Un usuario se puede conectar a otra máquina sin tener que estar cerca de ella.
- Contabilidad de recursos: establece el costo que se le cobra a un usuario por utilizar determinados recursos.

Características de los sistemas operativos.

En general, se puede decir que un Sistema Operativo tiene las siguientes características:

- Conveniencia. Un Sistema Operativo hace más conveniente el uso de una computadora.
- Eficiencia. Un Sistema Operativo permite que los recursos de la computadora se usen de la manera más eficiente posible.
- Habilidad para evolucionar. Un Sistema Operativo deberá construirse de manera que permita el desarrollo, prueba o introducción efectiva de nuevas funciones del sistema sin interferir con el servicio.
- Encargado de administrar el hardware. El Sistema Operativo se encarga de manejar de una mejor manera los recursos de la computadora en cuanto a hardware se refiere, esto es, asignar a cada proceso una parte del procesador para poder compartir los recursos.
- Relacionar dispositivos (gestionar a través del kernel). El Sistema Operativo se debe encargar de comunicar a los dispositivos periféricos, cuando el usuario así lo requiera.
- Organizar datos para acceso rápido y seguro.
- Manejar las comunicaciones en red. El Sistema Operativo permite al usuario manejar con alta facilidad todo lo referente a la instalación y uso de las redes de computadoras.
- Procesamiento por bytes de flujo a través del bus de datos.
- Facilitar las entradas y salidas. Un Sistema Operativo debe hacerle fácil al usuario el acceso y manejo de los dispositivos de Entrada/Salida de la computadora.

7 Gestión de procesos

Uno de los conceptos más importantes que gira entorno a un sistema operativo es el de proceso. Un proceso es un programa en ejecución junto con el entorno asociado (registros, variables, etc.).

El corazón de un sistema operativo es el núcleo, un programa de control que reacciona ante cualquier interrupción de eventos externos y que da servicio a los procesos, creándolos, terminándolos y respondiendo a cualquier petición de servicio por parte de los mismos.

Planificación del Procesador (Scheduler)

La planificación del procesador se refiere a la manera o técnicas que se usan para decidir cuánto tiempo de ejecución y cuando se le asignan a cada proceso del sistema. Obviamente, si el sistema es monousuario y monotarea no hay mucho que decidir, pero en el resto de los sistemas esto es crucial para el buen funcionamiento del sistema.

Características a considerar de los Procesos

No todos los equipos de cómputo procesan el mismo tipo de trabajos, y un algoritmo de planificación que en un sistema funciona excelente puede dar un rendimiento pésimo en otro cuyos procesos tienen características diferentes. Estas características pueden ser:

- Cantidad de Entrada/Salida: Existen procesos que realizan una gran cantidad de operaciones de entrada y salida (aplicaciones de bases de datos, por ejemplo).
- Cantidad de Uso de CPU: Existen procesos que no realizan muchas operaciones de entrada y salida, sino que usan intensivamente la unidad central de procesamiento. Por ejemplo, operaciones con matrices.
- Procesos de Lote o Interactivos: Un proceso de lote es más eficiente en cuanto a la lectura de datos, ya que generalmente lo hace de archivos, mientras que un programa interactivo espera mucho tiempo (no es lo mismo el tiempo de lectura de un archivo que la velocidad en que una persona teclea datos) por las respuestas de los usuarios.
- Procesos en Tiempo Real: Si los procesos deben dar respuesta en tiempo real se requiere que tengan prioridad para los turnos de ejecución.
- Longevidad de los Procesos: Existen procesos que típicamente requerirán varias horas para finalizar su labor, mientras que existen otros que solo necesitan algunos segundos.

Un proceso es una actividad que se apoya en datos, recursos, un estado en cada momento y un programa.

El Bloque de Control de Procesos (PCB).

Un proceso se representa desde el punto de vista del sistema operativo, por un conjunto de datos donde se incluyen el estado en cada momento, recursos utilizados, registros, etc., denominado Bloque de Control de Procesos (PCB).

Los objetivos del bloque de control de procesos son los siguientes:

- Localización de la información sobre el proceso por parte del sistema operativo.
- Mantener registrados los datos del proceso en caso de tener que suspender temporalmente su ejecución o reanudarla.

La información contenida en el bloque de control es la siguiente:

- Estado del proceso. Información relativa al contenido del controlador del programa (Program Counter, PC), estado de procesador en cuanto a prioridad del proceso, modo de ejecución, etc., y por último el estado de los registros internos de la computadora.
- Estadísticas de tiempo y ocupación de recursos para la gestión de la planificación del procesador.
- Ocupación de memoria interna y externa para el intercambio (swapping).
- Recursos en uso (normalmente unidades de entrada/salida).
- Archivos en uso.
- Privilegios.

Estas informaciones se encuentran en memoria principal en disco y se accede a ellas en los momentos en que se hace necesaria su actualización o consulta. Los datos relativos al estado del proceso siempre se encuentran en memoria principal.

Existe un Bloque de Control de Sistema (SCB) con objetivos similares al anterior y entre los que se encuentra el enlazado de los bloques de control de procesos existentes en el sistema.

El cambio de contexto se producirá en caso de ejecución de una instrucción privilegiada, una llamada al sistema operativo o una interrupción, es decir, siempre que se requiera la atención de algún servicio del sistema operativo.

Estado de los Procesos.

Los bloques de control de los procesos se almacenan en colas, cada una de las cuales representa un estado particular de los procesos, existiendo en cada bloque, entre otras informaciones. Los estados de los procesos son internos del sistema operativo y transparentes al usuario.

Los estados de los procesos se pueden dividir en dos tipos: activos e inactivos.

- Estados activos: Son aquellos que compiten con el procesador o están en condiciones de hacerlo. Se dividen en:
 - Ejecución. Estado en el que se encuentra un proceso cuando tiene el control del procesador. En un sistema monoprocesador este estado sólo lo puede tener un proceso.
 - Preparado. Aquellos procesos que están dispuestos para ser ejecutados, pero no están en ejecución por alguna causa (Interrupción, haber entrado en cola estando otro proceso en ejecución, etc.).
 - Bloqueado. Son los procesos que no pueden ejecutarse de momento por necesitar algún recurso no disponible (generalmente recursos de entrada/salida).
- Estados inactivos: Son aquellos que no pueden competir por el procesador, pero que pueden volver a hacerlo por medio de ciertas operaciones. En estos estados se mantiene el bloque de control de proceso aparcado hasta que vuelva a ser activado. Se trata de procesos que no han terminado su trabajo que lo han impedido y que pueden volver a activarse desde el punto en que se quedaron sin que tengan que volver a ejecutarse desde el principio. Son de dos tipos:
 - Suspendido bloqueado. Es el proceso que fue suspendido en espera de un evento, sin que hayan desaparecido las causas de su bloqueo.
 - Suspendido programado. Es el proceso que han sido suspendido, pero no tiene causa para estar bloqueado.

Operaciones sobre procesos.

Los sistemas operativos actuales poseen una serie de funciones cuyo objetivo es el de la manipulación de los procesos. Las operaciones que se pueden hacer sobre un proceso son las siguientes:

- Crear el proceso. Se produce con la orden de ejecución del programa y suele necesitar varios argumentos, como el nombre y la prioridad del proceso. Aparece en este momento el PCB, que será insertado en la cola de procesos preparados. La creación de un proceso puede ser de dos tipos:
 - Jerárquica. En ella, cada proceso que se crea es hijo del proceso creador y hereda el entorno de ejecución de su padre. El primer proceso que ejecuta un usuario será hijo del intérprete de comandos con el que interactúa.
 - No jerárquica. Cada proceso creado por otro proceso se ejecuta independientemente de su creador con un entorno diferente. Es un tipo de creación que no suele darse en los sistemas operativos actuales.
- Destruir un proceso. Se trata de la orden de eliminación del proceso con la cual el sistema operativo destruye su PCB.
- Suspendir un proceso. Es un proceso de alta prioridad que paraliza un proceso que puede ser reanudado posteriormente. Suele utilizarse en ocasiones de mal funcionamiento o sobrecarga del sistema.
- Reanudar un proceso. Trata de activar un proceso que a sido previamente suspendido.
- Cambiar la prioridad de un proceso.

- Temporizar la ejecución de un proceso. Hace que un determinado proceso se ejecute cada cierto tiempo (segundos, minutos, horas...) por etapas de una sola vez, pero transcurrido un periodo de tiempo fijo.
- Despertar un proceso. Es una forma de desbloquear un proceso que habrá sido bloqueado previamente por temporización o cualquier otra causa.

Prioridades

Todo proceso por sus características e importancia lleva aparejadas unas determinadas necesidades de ejecución en cuanto a urgencia y asignación de recursos.

Las prioridades según los sistemas operativos se pueden clasificar del siguiente modo:

- Asignadas por el sistema operativo. Se trata de prioridades que son asignadas a un proceso en el momento de comenzar su ejecución y dependen fundamentalmente de los privilegios de su propietario y del modo de ejecución.
- Asignadas por el propietario.
- Estáticas.
- Dinámicas.

El Núcleo del Sistema Operativo

Todas las operaciones en las que participan procesos son controladas por la parte del sistema operativo denominada núcleo (nucleus, core o kernel, en inglés). El núcleo normalmente representa sólo una pequeña parte de lo que por lo general se piensa que es todo el sistema operativo, pero es tal vez el código que más se utiliza. Por esta razón, el núcleo reside por lo regular en la memoria principal, mientras que otras partes del sistema operativo son cargadas en la memoria principal sólo cuando se necesitan.

Los núcleos se diseñan 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 interrupciones.

El núcleo de un sistema operativo normalmente contiene el código necesario para realizar las siguientes funciones:

- Manejo de interrupciones.
- Creación y destrucción de procesos.
- Cambio de estado de los procesos.
- Despacho.
- Suspensión y reanudación de procesos.
- Sincronización de procesos.
- Comunicación entre procesos.
- Manipulación de los bloques de control de procesos.
- Apoyo para las actividades de entrada/salida.
- Apoyo para asignación y liberación de memoria.
- Apoyo para el sistema de archivos.
- Apoyo para el mecanismo de llamada y retorno de un procedimiento.
- Apoyo para ciertas funciones de contabilidad del sistema.

Núcleo o Kernel y niveles de un Sistema Operativo

El Kernel consiste en la parte principal del código del sistema operativo, el cual se encargan de controlar y administrar los servicios y peticiones de recursos y de hardware con respecto a uno o varios procesos, este se divide en 5 capas:

- Nivel 1. Gestión de Memoria: que proporciona las facilidades de bajo nivel para la gestión de memoria secundaria necesaria para la ejecución de procesos.

- Nivel 2. Procesador: Se encarga de activar los cuantums de tiempo para cada uno de los procesos, creando interrupciones de hardware cuando no son respetadas.
- Nivel 3. Entrada/Salida: Proporciona las facilidades para poder utilizar los dispositivos de E/S requeridos por procesos.
- Nivel 4. Información o Aplicación o Interprete de Lenguajes: Facilita la comunicación con los lenguajes y el sistema operativo para aceptar las ordenes en cada una de las aplicaciones. Cuando se solicitan ejecutando un programa el software de este nivel crea el ambiente de trabajo e invoca a los procesos correspondientes.
- Nivel 5. Control de Archivos: Proporciona la facilidad para el almacenamiento a largo plazo y manipulación de archivos con nombre, va asignando espacio y acceso de datos en memoria.

El núcleo y los procesos

El núcleo (Kernel) de un sistema operativo es un conjunto de rutinas cuya misión es la de gestionar el procesador, la memoria, la entrada/salida y el resto de procesos disponibles en la instalación. Toda esta gestión la realiza para atender al funcionamiento y peticiones de los trabajos que se ejecutan en el sistema.

8 Gestión de memoria

La memoria es uno de los principales recursos de la computadora, la cual debe de administrarse con mucho cuidado. Aunque actualmente la mayoría de los sistemas de cómputo cuentan con una alta capacidad de memoria, de igual manera las aplicaciones actuales tienen también altos requerimientos de memoria, lo que sigue generando escasez de memoria en los sistemas multitarea y/o multiusuario.

La parte del sistema operativo que administra la memoria se llama administrador de memoria y su labor consiste en llevar un registro de las partes de memoria que se estén utilizando y aquellas que no, con el fin de asignar espacio en memoria a los procesos cuando éstos la necesiten y liberándola cuando terminen, así como administrar el intercambio entre la memoria principal y el disco en los casos en los que la memoria principal no le pueda dar capacidad a todos los procesos que tienen necesidad de ella.

Los sistemas de administración de memoria se pueden clasificar en dos tipos: los que desplazan los procesos de la memoria principal al disco y viceversa durante la ejecución y los que no.

El propósito principal de una computadora es el de ejecutar programas, estos programas, junto con la información que accedan deben de estar en la memoria principal (al menos parcialmente) durante la ejecución.

Para optimizar el uso del CPU y de la memoria, el sistema operativo debe de tener varios procesos a la vez en la memoria principal, para lo cual dispone de varias opciones de administración tanto del procesador como de la memoria. La selección de uno de ellos depende principalmente del diseño del hardware para el sistema. A continuación, se observarán los puntos correspondientes a la administración de la memoria.

Memoria real

La memoria real o principal es en donde son ejecutados los programas y procesos de una computadora y es el espacio real que existe en memoria para que se ejecuten los procesos. Por lo general esta memoria es de mayor costo que la memoria secundaria, pero el acceso a la información contenida en ella es de más rápido acceso. Solo la memoria cache es más rápida que la principal, pero su costo es a su vez mayor.

Memoria virtual

El termino memoria virtual se asocia a dos conceptos que normalmente a parecen unidos:

1. El uso de almacenamiento secundario para ofrecer al conjunto de las aplicaciones la ilusión de tener más memoria RAM de la que realmente hay en el sistema. Esta ilusión existe tanto a nivel del sistema, es decir, teniendo en ejecución más aplicaciones de las que realmente caben en la memoria principal, sin que por ello cada aplicación individual pueda usar más memoria de la que realmente hay o incluso de forma más general, ofreciendo a cada aplicación más memoria de la que existe físicamente en la máquina.
2. Ofrecer a las aplicaciones la ilusión de que están solas en el sistema, y que, por lo tanto, pueden usar el espacio de direcciones completo. Esta técnica facilita enormemente la generación de código, puesto que el compilador no tiene por qué preocuparse sobre dónde residirá la aplicación cuando se ejecute.

Espacio De Direcciones

Los espacios de direcciones involucrados en el manejo de la memoria son de tres tipos:

- Direcciones físicas: son aquellas que referencian alguna posición en la memoria física.
- Direcciones lógicas: son las direcciones utilizadas por los procesos. Sufren una serie de transformaciones, realizadas por el procesador (la MMU), antes de convertirse en direcciones físicas.
- Direcciones lineales: direcciones lineales se obtienen a partir de direcciones lógicas tras haber aplicado una transformación dependiente de la arquitectura.

Los programas de usuario siempre tratan con direcciones virtuales, nunca ven las direcciones físicas reales.

Unidad de Manejo de Memoria

La unidad de manejo de memoria (MMU) es parte del procesador. Sus funciones son:

- Convertir las direcciones lógicas emitidas por los procesos en direcciones físicas.
- Comprobar que la conversión se puede realizar. La dirección lógica podría no tener una dirección física asociada. Por ejemplo, la página correspondiente a una dirección se puede haber trasladado a una zona de almacenamiento secundario temporalmente.
- Comprobar que el proceso que intenta acceder a una cierta dirección de memoria tiene permisos para ello.
- La MMU se inicializa para cada proceso del sistema. Esto permite que cada proceso pueda usar el rango completo de direcciones lógicas (memoria virtual), ya que las conversiones de estas direcciones serán distintas para cada proceso.
- En todos los procesos se configura la MMU para que la zona del núcleo solo se pueda acceder en modo privilegiado del procesador.
- La configuración correspondiente al espacio de memoria del núcleo es idéntica en todos los procesos.

Intercambio

El objetivo del intercambio es dar cabida a la ejecución de más aplicaciones de las que pueden residir simultáneamente en la memoria del sistema:

Consiste en trasladar el código y los datos de un proceso completo de memoria al sistema de almacenamiento secundario, para cargar otro previamente almacenado, no permite a un proceso utilizar más memoria RAM de la que realmente existe en el sistema. Esta técnica puede ser ineficiente ya que se tiene que hacer el intercambio completo del proceso, aunque éste solo vaya a ejecutar una pequeña porción del código.

Durante el intercambio un proceso puede ser sacado temporalmente de memoria y llevado a un lugar especial del disco y posteriormente vuelto a memoria y continuada su ejecución.

El lugar de almacenamiento temporal suele ser un espacio suficientemente grande como para acomodar copias de las imágenes de memoria de todos los usuarios.

Asignación Contigua

La memoria principal normalmente se divide en dos particiones:

- Sistema operativo residente, normalmente en la parte baja de memoria con los vectores de interrupción.
- Procesos de usuario en la parte alta.

Asignación de partición simple:

Puede utilizarse un esquema de registro de relocalización y límite para proteger un proceso de usuario de otro y de cambios del código y datos del sistema operativo.

El registro de relocalización contiene la dirección física más pequeña; el registro límite contiene el rango de las direcciones lógicas cada dirección lógica debe ser menor al registro límite

Asignación de particiones múltiples:

Bloques de distintos tamaños están distribuidos en memoria, cuando llega un proceso se le asigna un hueco suficientemente grande para acomodarle.

El sistema operativo debe tener información sobre:

- Particiones asignadas
- Particiones libres (huecos)

Asignación de partición dinámica

El proceso de compactación es una instancia particular del problema de asignación de memoria dinámica, el cual es el cómo satisfacer una necesidad de tamaño n con una lista de huecos libres.

Existen muchas soluciones para el problema. El conjunto de huecos es analizado para determinar cuál hueco es el más indicado para asignarse. Las estrategias más comunes para asignar algún hueco de la tabla son:

- Primer ajuste: Consiste en asignar el primer hueco con capacidad suficiente. La búsqueda puede iniciar ya sea al inicio o al final del conjunto de huecos o en donde terminó la última búsqueda. La búsqueda termina al encontrar un hueco lo suficientemente grande.
- Mejor ajuste: Busca asignar el espacio más pequeño de los espacios con capacidad suficiente. La búsqueda se debe de realizar en toda la tabla, a menos que la tabla esté ordenada por tamaño. Esta estrategia produce el menor desperdicio de memoria posible.
- Peor ajuste: Asigna el hueco más grande. Una vez más, se debe de buscar en toda la tabla de huecos a menos que esté organizada por tamaño. Esta estrategia produce los huecos de sobra más grandes, los cuales pudieran ser de más uso si llegan procesos de tamaño mediano que quepan en ellos.

Se ha demostrado mediante simulacros que tanto el primer y el mejor ajuste son mejores que el peor ajuste en cuanto a minimizar tanto el tiempo del almacenamiento. Ni el primer o el mejor ajuste es claramente el mejor en términos de uso de espacio, pero por lo general el primer ajuste es más rápido.

Problema: La fragmentación.

Fragmentación

La fragmentación es la memoria que queda desperdiciada al usar los métodos de gestión de memoria que se vieron en los métodos anteriores. Tanto el primer ajuste, como el mejor y el peor producen fragmentación externa.

La fragmentación es generada cuando durante el reemplazo de procesos quedan huecos entre dos o más procesos de manera no contigua y cada hueco no es capaz de soportar ningún proceso de la lista de espera.

La fragmentación puede ser:

- Fragmentación Externa: existe el espacio total de memoria para satisfacer un requerimiento, pero no es contigua.
- Fragmentación Interna: la memoria asignada puede ser ligeramente mayor que la requerida; esta referencia es interna a la partición, pero no se utiliza.

La fragmentación externa se puede reducir mediante la compactación para colocar toda la memoria libre en un solo gran bloque, pero esta solo es posible si la relocalización es dinámica y se hace en tiempo de ejecución.

Paginación

Es una técnica de manejo de memoria, en la cual el espacio de memoria se divide en secciones físicas de igual tamaño, denominadas marcos de página. Los programas se dividen en unidades lógicas, denominadas páginas, que tienen el mismo tamaño que los marcos de páginas. De esta forma, se puede cargar una página de información en cualquier marco de página.

Las páginas sirven como unidad de almacenamiento de información y de transferencia entre memoria principal y memoria auxiliar o secundaria. Cada marco se identifica por la dirección de marco, que está en la posición física de la primera palabra en el marco de página.

Las páginas de un programa necesitan estar contiguamente en memoria, aunque el programador lo observe de esta forma. Los mecanismos de paginación permiten la correspondencia correcta entre las direcciones virtuales (dadas por los programas) y las direcciones reales de la memoria que se reverencien.

Cada página consiste en z palabras contiguas; un espacio de direcciones N de un programa consiste de n páginas $(0, 1, 2, 3, \dots, n-1)$ ($n \cdot z$ direcciones virtuales) y el espacio de memoria consiste de m marcos

de páginas $(0, z, 2z, \dots, (m-1)z)$ ($m \cdot z$ posiciones). Una dirección virtual a es equivalente a una dirección dada como una dupla (p, d) , en la cual p es el número de la página y d el número de la palabra dentro de la página, de acuerdo con la relación:

$$\begin{aligned}a &= p \cdot z + d \quad (0 \leq d < z) \\ p &= (a/z) \quad (\text{parte entera de la división}) \\ d &= a \bmod z \quad (\text{resto de divisor } a/z)\end{aligned}$$

En las máquinas que usan aritmética binaria, el cálculo de (p, d) es trivial, si z es potencia de 2. Por ejemplo, si el campo de direcciones de la instrucción es de m bits ($m > 6$), los cuatro bits más significativos indican el número de la página y los $m-4$ bits restantes, el desplazamiento.

Para tener el control de las páginas, debe mantenerse una tabla en memoria que se denomina tabla de Mapas de Página (PMT) para cada uno de los procesos.

Hasta ahora, los métodos que hemos visto de la administración de la memoria principal, nos han dejado con un problema: fragmentación, (huecos en la memoria que no pueden usarse debido a lo pequeño de su espacio) lo que nos provoca un desperdicio de memoria principal.

Una posible solución para la fragmentación externa es permitir que espacio de direcciones lógicas lleve a cabo un proceso en direcciones no contiguas, así permitiendo al proceso ubicarse en cualquier espacio de memoria física que esté disponible, aunque esté dividida. Una forma de implementar esta solución es a través del uso de un esquema de paginación. La paginación evita el considerable problema de ajustar los pedazos de memoria de tamaños variables que han sufrido los esquemas de manejo de memoria anteriores. Dado a sus ventajas sobre los métodos previos, la paginación, en sus diversas formas, es usada en muchos sistemas operativos.

Al utilizar la memoria virtual, las direcciones no pasan en forma directa al bus de memoria, sino que van a una unidad administradora de la memoria (MMU –Memory Management Unit). Estas direcciones generadas por los programas se llaman direcciones virtuales y conforman el hueco de direcciones virtuales. Este hueco se divide en unidades llamadas páginas. Las unidades correspondientes en la memoria física se llaman marcos para página o frames. Las páginas y los frames tienen siempre el mismo tamaño.

Tablas de páginas

Cada página tiene un número que se utiliza como índice en la tabla de páginas, lo que da por resultado el número del marco correspondiente a esa página virtual. Si el bit presente / ausente es 0, se provoca un señalamiento (trap) hacia el sistema operativo. Si el bit es 1, el número de marco que aparece en la tabla de páginas se copia en los bits de mayor orden del registro de salida, junto con el ajuste (offset) de 12 bits, el cual se copia sin modificaciones de la dirección virtual de entrada. Juntos forman una dirección física de 15 bits. El registro de salida se coloca entonces en el bus de la memoria como la dirección en la memoria física.

En teoría, la asociación de las direcciones virtuales con las físicas se efectúa según lo descrito. El número de página virtual se divide en un número de página virtual (los bits superiores) y un ajuste (los bits inferiores). El número de página virtual se utiliza como un índice en la tabla de páginas para encontrar la entrada de esa página virtual. El número de marco (si existe) se determina a partir de la tabla de páginas. El número de marco se asocia al extremo superior del ajuste y reemplaza al número de página virtual para formar una dirección física que se puede enviar a la memoria.

La finalidad de la tabla de páginas es asociar las páginas virtuales con los marcos. En términos matemáticos, la tabla de páginas es una función, cuyo argumento es el número de página virtual y como resultado el número del marco físico. Mediante el resultado de esta función, se puede reemplazar el campo de la página virtual de una dirección virtual por un campo de marco, lo que produce una dirección en la memoria física. Sin embargo, hay que enfrentar dos aspectos fundamentales:

-
1. La tabla de páginas puede ser demasiado grande.
 2. La asociación debe ser rápida.

El primer punto proviene del hecho de que las computadoras modernas utilizan direcciones virtuales de al menos 32 bits. Por ejemplo, si el tamaño de página es de 4K, un hueco de direcciones de 32 bits tiene un millón de páginas; en el caso de un hueco de direcciones de 64 bits, se tendría más información de la que uno quisiera contemplar.

El segundo punto es consecuencia del hecho de que la asociación virtual – física debe hacerse en cada referencia a la memoria. Una instrucción común tiene una palabra de instrucción y también un operando de memoria. Entonces es necesario hacer una, dos o más referencias a la tabla de páginas por cada instrucción.

Características de la paginación:

- El espacio de direcciones lógico de un proceso puede ser no contiguo.
- Se divide la memoria física en bloques de tamaño fijo llamados marcos (frames).
- Se divide la memoria en bloques de tamaño llamados páginas.
- Se mantiene información en los marcos libres.
- Para correr un programa de n páginas de tamaño, se necesitan encontrara n marcos y cargar el programa.
- Se establece una tabla de páginas para trasladar las direcciones lógicas a físicas.
- Se produce fragmentación interna.

Ventajas de la paginación

- Es posible comenzar a ejecutar un programa, cargando solo una parte del mismo en memoria, y el resto se cargará bajo la solicitud.
- No es necesario que las paginas estén contiguas en memoria, por lo que no se necesitan procesos de compactación cuando existen marcos de páginas libres dispersos en la memoria.
- Es fácil controlar todas las páginas, ya que tienen el mismo tamaño.
- El mecanismo de traducción de direcciones (DAT) permite separar los conceptos de espacio de direcciones y espacios de memoria. Todo el mecanismo es transparente al usuario.
- Se libera al programador de la restricción de programar para un tamaño físico de memoria, con lo que se aumenta su productividad. Se puede programar en función de una memoria mucho más grande a la existente.
- Al no necesitarse cargar un programa completo en memoria para su ejecución, se puede aumentar el número de programas multiprogramándose.
- Se elimina el problema de fragmentación externa.

Desventajas de la paginación

- El costo de hardware y software se incrementa, por la nueva información que debe manejarse y el mecanismo de traducción de direcciones necesario. Se consume muchos más recursos de memoria, tiempo en el CPU para su implantación.
- Se deben reservar áreas de memoria para las PMT de los procesos. Al no ser fija el tamaño de estas, se crea un problema semejante al de los programas (como asignar un tamaño óptimo sin desperdicio de memoria, u "ovearhead" del procesador).
- Aparece el problema de fragmentación interna. Así, si se requieren 5K para un programa, pero las paginas son de 4K, deberán asignárseles 2 paginas (8k), con lo que quedan 3K sin utilizar. La suma de los espacios libres dejados de esta forma puede ser mayor que el de varias páginas, pero no podrá ser utilizados. Debe asignarse un tamaño promedio a las páginas, evitando que, si son muy pequeñas, se necesiten TABLAS BMT y PMT muy grandes, y si son muy grandes, se incremente el grado de fragmentación interna.

Traducción de Direcciones

La dirección generada por la CPU se divide en:

-
- Numero de página (p): utilizado como índice en la tabla de páginas que contiene la dirección base de cada página en la memoria física.
 - Offset de la página (d): combinado con la dirección base define la dirección física que será enviada a la unidad de memoria.

Segmentación

Es un esquema de manejo de memoria mediante el cual la estructura del programa refleja su división lógica; llevándose a cabo una agrupación lógica de la información en bloques de tamaño variable denominados segmentos. Cada uno de ellos tienen información lógica del programa: subrutina, arreglo, etc. Luego, cada espacio de direcciones de programa consiste de una colección de segmentos, que generalmente reflejan la división lógica del programa.

La segmentación permite alcanzar los siguientes objetivos:

- Modularidad de programas: cada rutina del programa puede ser un bloque sujeto a cambios y recompilaciones, sin afectar por ello al resto del programa.
- Estructuras de datos de largo variable: ejm. Stack, donde cada estructura tiene su propio tamaño y este puede variar.
- Protección: se puede proteger los módulos del segmento contra accesos no autorizados.
- Compartición: dos o más procesos pueden ser un mismo segmento, bajo reglas de protección; aunque no sean propietarios de los mismos.
- Enlace dinámico entre segmentos: puede evitarse realizar todo el proceso de enlace antes de comenzar a ejecutar un programa. Los enlaces se establecerán solo cuando sea necesario.

Ventajas de la segmentación

El esquema de segmentación ofrece las siguientes ventajas:

- El programador puede conocer las unidades lógicas de su programa, dándoles un tratamiento particular.
- Es posible compilar módulos separados como segmentos el enlace entre los segmentos puede suponer hasta tanto se haga una referencia entre segmentos.
- Debido a que es posible separar los módulos se hace más fácil la modificación de los mismos. Cambios dentro de un módulo no afecta al resto de los módulos.
- Es fácil el compartir segmentos.
- Es posible que los segmentos crezcan dinámicamente según las necesidades del programa en ejecución.
- Existe la posibilidad de definir segmentos que aún no existan. Así, no se asignará memoria, sino a partir del momento que sea necesario hacer usos del segmento. Un ejemplo de esto, serían los arreglos cuya dimensión no se conoce hasta tanto no se comienza a ejecutar el programa. En algunos casos, incluso podría retardar la asignación de memoria hasta el momento en el cual se referencia el arreglo u otra estructura de dato por primera vez.

Desventajas de la segmentación

- Hay un incremento en los costos de hardware y de software para llevar a cabo la implantación, así como un mayor consumo de recursos: memoria, tiempo de CPU, etc.
- Debido a que los segmentos tienen un tamaño variable se pueden presentar problemas de fragmentación externas, lo que puede ameritar un plan de reubicación de segmentos en memoria principal.
- Se complica el manejo de memoria virtual, ya que los discos almacenan la información en bloques de tamaños fijos, mientras los segmentos son de tamaño variable. Esto hace necesaria la existencia de mecanismos más costosos que los existentes para paginación.

-
- Al permitir que los segmentos varíen de tamaño, puede ser necesarios planes de reubicación a nivel de los discos, si los segmentos son devueltos a dicho dispositivo; lo que conlleva a nuevos costos.
 - No se puede garantizar que, al salir un segmento de la memoria, este pueda ser traído fácilmente de nuevo, ya que será necesario encontrar nuevamente un área de memoria libre ajustada a su tamaño.
 - La compartición de segmentos permite ahorrar memoria, pero requiere de mecanismos adicionales de hardware y software.

Estas desventajas tratan de ser minimizadas, bajo la técnica conocida como Segmentación paginada.

Segmentación Paginada

Paginación y segmentación son técnicas diferentes, cada una de las cuales busca brindar las ventajas enunciadas anteriormente.

Para la segmentación se necesita que estén cargadas en memoria, áreas de tamaños variables. Si se requiere cargar un segmento en memoria; que antes estuvo en ella y fue removido a memoria secundaria; se necesita encontrar una región de la memoria lo suficientemente grande para contenerlo, lo cual no es siempre factible; en cambio "recargar" una página implica solo encontrar un espacio de página disponible.

A nivel de paginación, si quiere referenciar en forma cíclica n páginas, estas deberán ser cargadas una a una generándose varias interrupciones por fallas de páginas; bajo segmentación, esta página podría conformar un solo segmento, ocurriendo una sola interrupción, por falla de segmento. No obstante, si bajo segmentación, se desea acceder un área muy pequeña dentro de un segmento muy grande, este deberá cargarse completamente en memoria, desperdiciándose memoria; bajo paginación solo se cargará la página que contiene los ítems referenciados.

Puede hacerse una combinación de segmentación y paginación para obtener las ventajas de ambas. En lugar de tratar un segmento como una unidad contigua, este puede dividirse en páginas. Cada segmento puede ser descrito por su propia tabla de páginas.

Los segmentos son usualmente múltiplos de páginas en tamaño, y no es necesario que todas las páginas se encuentren en memoria principal a la vez; además las páginas de un mismo segmento, aunque se encuentren contiguas en memoria virtual; no necesitan estarlo en memoria real.

Las direcciones tienen tres componentes: (s, p, d), donde la primera indica el número del segmento, la segunda el número de la página dentro del segmento y la tercera el desplazamiento dentro de la página. Se deberán usar varias tablas:

- SMT (tabla de mapas de segmentos): una para cada proceso. En cada entrada de la SMT se almacena la información descrita bajo segmentación pura, pero en el campo de dirección se indicará la dirección de la PMT (tabla de mapas de páginas) que describe a las diferentes páginas de cada segmento.
- PMT (tabla de mapas de páginas): una por segmento; cada entrada de la PMT describe una página de un segmento; en la forma que se presentó la página pura.
- TBM (tabla de bloques de memoria): para controlar asignación de páginas por parte del sistema operativo.
- JT (tabla de Jobs): que contiene las direcciones de comienzo de cada una de las SMT de los procesos que se ejecutan en memoria.

En el caso, de que un segmento sea de tamaño inferior o igual al de una página, no se necesita tener la correspondiente PMT, actuándose en igual forma que bajo segmentación pura; puede agregarse un bit adicional (S) a cada entrada de la SMT, que indicara si el segmento está paginado o no.

Ventajas de la segmentación paginada

El esquema de segmentación paginada tiene todas las ventajas de la segmentación y la paginación:

- Debido a que los espacios de memorias son segmentados, se garantiza la facilidad de implantar la compartición y enlace.
- Como los espacios de memoria son paginados, se simplifican las estrategias de almacenamiento.
- Se elimina el problema de la fragmentación externa y la necesidad de compactación.

Desventajas de la segmentación paginada

- Las tres componentes de la dirección y el proceso de formación de direcciones hacen que se incremente el costo de su implantación. El costo es mayor que en el caso de segmentación pura o paginación pura.
- Se hace necesario mantener un número mayor de tablas en memoria, lo que implica un mayor costo de almacenamiento.
- Sigue existiendo el problema de fragmentación interna de todas- o casi- todas las páginas finales de cada uno de los segmentos. Bajo paginación pura se desperdician solo la última página asignada, mientras que bajo segmentación – paginada el desperdicio puede ocurrir en todos los segmentos asignados.

Para concluir se entiende que:

- En la memoria principal son ejecutados los programas y procesos de una computadora y es el espacio real que existe en memoria para que se ejecuten los procesos.
- La memoria virtual es aquella que ofrece a las aplicaciones la ilusión de que están solas en el sistema y que pueden usar el espacio de direcciones completo.
- Las direcciones de memoria son de tres tipos: físicas, lógicas y lineales.
- El objetivo del intercambio es dar cabida a la ejecución de más aplicaciones de las que pueden residir simultáneamente en la memoria del sistema.
- La asignación consiste en determinar cuál espacio vacío en la memoria principal es el más indicado para ser asignado a un proceso.
- Las estrategias más comunes para asignar espacios vacíos (huecos) son: primer ajuste, mejor ajuste, peor ajuste.
- La fragmentación es la memoria que queda desperdiciada al usar los métodos de gestión de memoria tal como la asignación.
- La fragmentación puede ser interna o externa.
- La paginación es una técnica de gestión de memoria en la cual el espacio de memoria se divide en secciones físicas de igual tamaño llamadas marcos de página, las cuales sirven como unidad de almacenamiento de información.
- La segmentación es un esquema de manejo de memoria mediante el cual la estructura del programa refleja su división lógica; llevándose a cabo una agrupación lógica de la información en bloques de tamaño variable denominados segmentos.

9 Gestión de almacenamiento (sistema de archivos)

Un sistema de archivos (File System) es una estructura de directorios con algún tipo de organización el cual nos permite almacenar, crear y borrar archivos en diferentes formatos. En esta sección se revisarán conceptos importantes relacionados con los sistemas de archivos.

ALMACENAMIENTO FÍSICO DE DATOS.

En un sistema informático es evidente que existe la necesidad por parte de los usuarios y aplicaciones de almacenar datos en algún medio, a veces por periodos largos y a veces por instantes. Cada aplicación y cada usuario deben tener ciertos derechos con sus datos, como son el poder crearlos y borrarlos, o cambiarlos de lugar; así como tener privacidad frente a otros usuarios o aplicaciones. El sub- sistema de archivos del sistema operativo se debe encargar de estos detalles, además de establecer el formato físico en el cual almacenará los datos en discos duros, cintas o discos flexibles.

Es de todos conocido que, tradicionalmente, la información en los sistemas modernos se almacena en discos duros, discos flexibles y unidades de disco óptico, y en todos ellos se comparten algunos esquemas básicos para darles formato físico: las superficies de almacenamiento están divididas en círculos concéntricos llamados «pistas» y cada pista dividida en «sectores». A la unión lógica de varias pistas a través de varias superficies «paralelas» de almacenamiento se les llama «cilindros», los cuales son inspeccionados al momento de lectura o escritura de datos por las respectivas unidades físicas llamadas

«cabezas». Las superficies de almacenamiento reciben el nombre de «platos» y generalmente están en movimiento rotativo para que las cabezas accedan a las pistas que los componen. Los datos se escriben a través de los sectores en las pistas y cilindros modificando las superficies por medio de las cabezas.

El tiempo que una cabeza necesita para ir de una pista a otra se le llama tiempo de búsqueda y depender/a de la distancia entre la posición actual y la distancia a la pista buscada. El tiempo que tarda una cabeza en posicionarse, dentro de una pista, desde el sector actual al sector deseado se le denomina tiempo de latencia y depende de la distancia entre sectores y la velocidad de rotación del disco. El impacto que tiene las lecturas y escrituras sobre el sistema está determinado por la tecnología usada en los platos y cabezas y por la forma de resolver las peticiones de lectura y escritura, es decir, los algoritmos de planificación.

Algoritmos de planificación de peticiones

Los algoritmos de planificación de peticiones de lectura y escritura a discos se encargan de registrar dichas peticiones y de responderlas en un tiempo razonable. Los algoritmos más comunes para esta tarea son:

- Primero en llegar, primero en ser servido (FIFO): las peticiones son encoladas de acuerdo al orden en que llegaron y de esa misma forma se van leyendo o escribiendo las mismas. La ventaja de este algoritmo es su simplicidad que no causa sobrecarga, su desventaja principal es que no aprovecha ninguna característica de las peticiones, de manera que es muy factible que el brazo del disco se mueva muy ineficientemente, ya que las peticiones pueden tener direcciones en el disco unas muy alejadas de otras. Por ejemplo, si se están haciendo peticiones a los sectores 6, 10, 8, 21 y 4, las mismas serán resueltas en el mismo orden.
- Primero el más cercano a la posición actual: en este algoritmo las peticiones se ordenan de acuerdo a la posición actual de la cabeza lectora, sirviendo primero a aquellas peticiones más cercanas y reduciendo, así, el movimiento del brazo, lo cual constituye la ventaja principal de este algoritmo. Su desventaja consiste en que puede haber solicitudes que se queden esperando para siempre, en el infortunado caso de que existan peticiones muy alejadas y en todo momento estén entrando peticiones que estén más cercanas. Para las peticiones 6, 10, 8, 21 y 4, las mismas serán resueltas en el orden 4, 6, 8, 10 y 21.
- Por exploración (algoritmo del elevador): en este algoritmo el brazo se está moviendo en todo momento desde el perímetro del disco hacia su centro y viceversa, resolviendo las peticiones que

existan en la dirección que tenga en turno. En este caso las peticiones 6, 10, 8, 21 y 4 serán resueltas en el orden 6, 10, 21, 8 y 4; es decir, la posición actual es 6 y como va hacia los sectores de mayor numeración (hacia el centro, por ejemplo), en el camino sigue el sector 10, luego el 21, siendo el más céntrico. Ahora el brazo resolverá las peticiones en su camino hacia el borde del plato y la primera petición que se encuentra es la del sector 8 y luego la 4. La ventaja de este algoritmo es que el brazo se mueve mucho menos que en FIFO y evita la espera indefinida; su desventaja es que no es justo, ya que no sirve las peticiones en el orden en que llegan, además las peticiones en los extremos interior y exterior gozan de un tiempo de respuesta un poco mayor.

- Por exploración circular: es una variación del algoritmo anterior, con la única diferencia que, al llegar a la parte central, el brazo regresa al exterior sin resolver ninguna petición, lo cual proporciona un tiempo de respuesta más cercano al promedio para todas las peticiones, sin importar si están cercas del centro o del exterior.

Asignación del espacio de almacenamiento.

El subsistema de archivos se debe encargar de localizar espacio libre en los medios de almacenamiento para guardar archivos y para después borrarlos, renombrarlos o agrandarlos. Para ello se vale de ubicaciones especiales que contienen la lista de archivos creados y por cada archivo una serie de direcciones que apuntan a su contenido. Esas localidades especiales se llaman directorios. Para asignarle espacio a los archivos existen tres criterios generales que se describen a continuación.

- Asignación contigua: cada directorio contiene los nombres de archivos y la dirección del bloque inicial de cada archivo, así como el tamaño total de cada uno. Por ejemplo, si un archivo comienza en el sector 17 y mide 10 bloques, cuando el archivo sea accedido, el brazo se moverá inicialmente al bloque 17 y de ahí hasta el 27. Si el archivo es borrado y luego creado otro más pequeño, quedarían huecos inútiles entre archivos útiles, lo cual se denomina fragmentación externa.
- Asignación encadenada: con este criterio los directorios contienen los nombres de archivos y por cada uno de ellos la dirección del bloque inicial que compone al archivo. Cuando un archivo es leído, el brazo va a esa dirección inicial y encuentra los datos iniciales junto con la dirección del siguiente bloque y así sucesivamente.

Con este criterio no es necesario que los bloques estén contiguos y no existe la fragmentación externa, pero en cada «eslabón» de la cadena se desperdicia espacio con las direcciones mismas. En otras palabras, lo que se crea en el disco es una lista ligada.

- Asignación con índices (indexada): en este esquema se guarda en el directorio un bloque de índices para cada archivo, con apuntadores hacia todos sus bloques constituyentes, de manera que el acceso directo se agiliza notablemente, a cambio de sacrificar varios bloques para almacenar dichos apuntadores. Cuando se quiere leer un archivo o cualquiera de sus partes, se hacen dos accesos: uno al bloque de índices y otro a la dirección deseada. Éste es un esquema excelente para archivos grandes, pero no para pequeños, para los que, la relación entre bloques destinados a índices respecto de los destinados para datos supone un coste inasumible.

Métodos de acceso en los sistemas de archivos.

Los métodos de acceso se refieren a las capacidades que el subsistema de archivos proporciona para acceder a datos dentro de los directorios y medios de almacenamiento en general. Se ubican tres formas generales: acceso secuencial, acceso directo y acceso directo indexado.

- Acceso secuencial: es el método más lento y consiste en recorrer los componentes de un archivo uno a uno hasta llegar al registro deseado. Se necesita que el orden lógico de los registros sea igual al orden físico en el medio de almacenamiento. Este tipo de acceso se usa comúnmente en cintas y cartuchos.
- Acceso directo: permite acceder a cualquier sector o registro inmediatamente, por medio de llamadas al sistema como la de seek. Este tipo de acceso es rápido y se usa comúnmente en discos duros y discos o archivos manejados en memoria de acceso aleatorio.

-
- Acceso directo indexado: este tipo de acceso es útil para grandes volúmenes de información o datos. Consiste en que cada archivo tiene una tabla de apuntadores, donde cada apuntador va a la dirección de un bloque de índices, lo cual permite que el archivo se expanda a través de un espacio enorme. Consume una cantidad importante de recursos en las tablas de índices, pero es muy rápido.

Operaciones soportadas por el subsistema de archivos.

Independientemente de los algoritmos de asignación de espacio, de los métodos de acceso y de la forma de resolver las peticiones de lectura y escritura, el subsistema de archivos debe proveer un conjunto de llamadas al sistema para operar con los datos proporcionando mecanismos de protección y seguridad. Las operaciones básicas que la mayoría de los sistemas de archivos soportan son:

- Crear (create): permite crear un archivo sin datos, con el propósito de indicar que ese nombre ya está usado. Se deben crear las estructuras básicas para soportarlo.
- Borrar (delete): eliminar el archivo y liberar los bloques para su uso posterior.
- Abrir (open): antes de usar un archivo se debe abrir para que el sistema conozca sus atributos, tales como el propietario, la fecha de modificación, etc.
- Cerrar (close): después de realizar todas las operaciones deseadas, el archivo debe cerrarse para asegurar su integridad y para liberar recursos de su control en la memoria.
- Leer o Escribir (read, write): añadir información al archivo o leer el carácter o una cadena de caracteres a partir de la posición actual.
- Concatenar (append): es una forma restringida de la llamada «write», en la cual sólo se permite añadir información al final del archivo.
- Localizar (seek): para los archivos de acceso directo se permite posicionar el apuntador de lectura o escritura en un registro aleatorio, a veces a partir del inicio o final del archivo.
- Leer atributos: permite obtener una estructura con todos los atributos del archivo especificado, tales como permisos de escritura, de borrado, ejecución, etc.
- Asignar atributos: permite cambiar los atributos de un archivo, por ejemplo, en UNIX, donde todos los dispositivos se manejan como si fueran archivos, es posible cambiar el comportamiento de una terminal con una de estas llamadas.
- Renombrar (rename): permite cambiarle el nombre e incluso a veces la posición en la organización de directorios del archivo especificado.

Los subsistemas de archivos también proporcionan un conjunto de llamadas para operar sobre directorios, las más comunes son crear, borrar, abrir, cerrar, renombrar y leer. Sus funcionalidades son obvias, pero existen también otras dos operaciones no tan comunes que son la de «crear un enlace» y la de «destruir un enlace». La operación de crear un enlace sirve para que, desde diferentes puntos de la organización de directorios, se pueda acceder a un mismo archivo sin necesidad de copiarlo o duplicarlo. La llamada a «destruir un enlace» lo que hace es eliminar esas referencias, sin afectar al archivo original.

Algunas facilidades extras de los sistemas de archivos

Algunos sistemas de archivos proporcionan herramientas al administrador del sistema para facilitarle su labor. Las más notables son la utilidad de compartir archivos y los sistemas de «cuotas».

La utilidad de compartir archivos se refiere a la posibilidad de que los permisos de los archivos o directorios dejen que un grupo de usuarios puedan acceder para diferentes operaciones: leer, escribir, borrar, crear, etc. El dueño verdadero es quien decide qué permisos se aplicarán al grupo e, incluso, a otros usuarios que no formen parte de su grupo. La utilidad de «cuotas» se refiere a que el sistema de archivos es capaz de llevar un control para que cada usuario pueda usar un máximo de espacio de almacenamiento. Cuando el usuario excede ese límite, el sistema le avisa y le niega el permiso de seguir escribiendo, obligándole a borrar algunos archivos si es que quiere almacenar otros o que crezcan los existentes. La versión de UNIX SunOS contiene esa utilidad.

SISTEMAS DE ARCHIVOS AISLADOS

Los sistemas de archivos aislados son aquellos que residen en un solo ordenador y no existe la posibilidad de que, aun estando en una red, otros sistemas puedan usar sus directorios y archivos. Por ejemplo, los archivos en discos duros en el sistema MS-DOS clásico se puede englobar en esta categoría.

SISTEMAS DE ARCHIVOS COMPARTIDOS O DE RED

Estos sistemas de archivos permiten ser accedidos desde otros nodos en una red. Generalmente, existe un «servidor» que es el ordenador en donde reside el sistema de archivos físicamente, y por otro lado están los «clientes», que se valen del servidor para ver sus archivos y directorios como si fueran locales a la máquina.

Los sistemas de archivos compartidos en red más populares son los provistos por Netware, el Remote File Sharing (RFS en UNIX), Network File System (NFS de Sun Microsystems) y el Andrew File System (AFS). En general, los servidores proporcionan un medio para que los clientes, localmente, realicen peticiones de operaciones sobre archivos las cuales son «atrapadas» por un «driver» o un «módulo» en el núcleo del sistema operativo, el cual se comunica con el servidor a través de la red, ejecutándose la operación en el servidor. Existen servidores de tipo «stateless» y «no-stateless». Un servidor «stateless» no registra el estado de las operaciones sobre los archivos, de manera que el cliente se encarga de todo ese trabajo. La ventaja de este esquema es que, si el servidor falla, el cliente no perderá información ya que ésta se guarda en memoria localmente, de manera que cuando el servidor reanude su servicio el cliente proseguirá como si nada hubiese sucedido. Con un servidor «no-stateless», esto no es posible.

La protección sobre las operaciones se lleva a cabo tanto en los clientes como en el servidor: si el usuario quiere ejecutar una operación indebida sobre un archivo, recibirá un mensaje de error y posiblemente se envíe un registro al subsistema de «seguridad» para informar al administrador del sistema de dicho intento de violación.

En la práctica, el conjunto de permisos que cada usuario tiene sobre el total de archivos se almacena en estructuras llamadas «listas de acceso» (access lists).

10 Protección y seguridad

Los términos "seguridad" y "protección" se suelen considerar como sinónimos, pero se puede hacer una distinción entre ambos. La seguridad comprende toda la problemática relacionada con hacer que los ficheros no puedan ser leídos o escritos por ninguna persona no autorizada, e incluye aspectos técnicos, administrativos, legales y políticos. Por otro lado, la protección se podría definir como el conjunto de mecanismos específicos del sistema operativo que tienen como finalidad el proporcionar seguridad.

Seguridad

La seguridad se aplica tanto a pérdida de datos como a intrusos que intentan hacer lo que no deben. Algunas de las causas comunes de pérdida de información son las siguientes:

- Desastres tales como incendios, terremotos, riadas, etc.
- Errores hardware o software.
- Errores humanos.

La mayoría de estos problemas se pueden solucionar realizando copias de seguridad y almacenando las cintas en lugares físicamente lejanos a los datos originales.

Existen dos tipos de intrusos: intrusos pasivos que únicamente quieren leer información no autorizada e intrusos activos que quieren realizar modificaciones no permitidas. Los intrusos se pueden clasificar en las siguientes categorías:

- Usuarios no técnicos motivados por la curiosidad.
- Personal cualificado (estudiantes, programadores, técnicos, ...), que consideran un reto personal violar la seguridad de un sistema.
- Intento de obtener beneficios económicos.
- Espionaje comercial o militar.

Podemos definir tres requisitos necesarios para la seguridad dentro de un sistema informático:

- Privacidad: la información debe ser accesible para lectura únicamente por las partes autorizadas. Este tipo de acceso incluye la impresión, tanto por pantalla como impresora, e incluso la revelación de la existencia de un determinado objeto o suceso.
- Integridad: los elementos relacionados con la seguridad únicamente pueden ser modificados por las partes autorizadas. Modificación incluye escritura, creación, cambio de estado y borrado.
- Disponibilidad: los elementos relacionados con la seguridad deben estar disponibles únicamente a las partes autorizadas.

Tipos de Amenazas

Los tipos de amenazas a la seguridad de un sistema informático los podemos caracterizar teniendo en cuenta como esta información es suministrada por el sistema. En general, hay un flujo de información de una fuente a un destino. Teniendo esto en cuenta, podemos señalar cuatro categorías de amenazas:

- Interrupción: Un elemento del sistema es destruido o se hace inservible. Es una amenaza a la disponibilidad. Ejemplos son la destrucción de algún elemento hardware (discos, líneas de comunicación, etc.) y la desactivación del sistema de gestión de ficheros.
- Intercepción: Una parte no autorizada obtiene acceso a un elemento relacionado con la seguridad. Es una amenaza a la privacidad. La parte no autorizada puede ser una persona, un programa o un computador. Ejemplos son la copia ilícita de programas y la visualización de ficheros que han de permanecer ocultos.
- Modificación: Una parte no autorizada no sólo obtiene acceso, sino que puede modificar un elemento relacionado con la seguridad. Es una amenaza a la integridad. Ejemplos son la

alteración del contenido de un fichero y modificar un programa para que funcione de forma diferente.

- Fabricación: Una parte no autorizada inserta nuevos elementos en el sistema. Es una amenaza a la integridad. Ejemplos son adición de registros a un fichero y la inclusión de mensajes espúreos en una red.

La mayoría de los métodos de autenticación se basan en identificar algo que el usuario tiene o conoce. El mecanismo más común de autenticación consiste en que todo usuario ha de introducir una contraseña, que es solicitada por el programa de conexión cuando el usuario introduce su nombre. El inconveniente de este método es que las contraseñas pueden ser fácilmente averiguables si el usuario utiliza su nombre, dirección, o similar como contraseña. Otra forma de averiguar una contraseña consiste en probar todas las combinaciones de letras, números y símbolos de puntuación hasta adivinar la contraseña.

Existen variantes como que el sistema le añada a cada contraseña un número aleatorio, o asignarle a cada usuario un libro con una secuencia de contraseñas, de forma que cada vez que se conecta tiene que introducir la palabra de paso siguiente.

Otros tipos de mecanismos de autenticación se pueden basar en objetos que únicamente cada usuario puede tener, como tarjetas con banda magnética, al estilo de los cajeros automáticos. Otra posibilidad es medir o comprobar ciertas características que están indisolublemente unidas a cada persona, como la voz, escritura, huellas dactilares, etc.

En instalaciones en las que la seguridad es prioritaria, estas medidas se pueden complementar con restricciones de acceso a la habitación en la que se encuentran los terminales, asignar a cada usuario un terminal concreto, establecer un horario concreto de trabajo, etc.

Amenazas de Origen Software

Uno de los tipos más sofisticados de amenazas tiene su origen en programas que explotan las debilidades de los sistemas. Estos programas se dividen en dos grupos: aquellos que necesitan un programa anfitrión y aquellos que son independientes. Los primeros son trozos de programas que no pueden existir de forma autónoma, mientras que los segundos son programas completos que pueden ser planificados y ejecutados por el sistema operativo. También hay que distinguir entre aquellos programas que no se replican y los que lo hacen. Estos últimos son programas o trozos de programas que cuando se ejecutan pueden generar una o más copias de ellos mismos, que serán posteriormente activadas en la

Podemos distinguir seis tipos de amenazas de origen software:

- Bomba Lógica: Es un código incrustado en un programa que comprueba si ciertas condiciones se cumplen, en cuyo caso ejecuta alguna acción no autorizada. Estas condiciones pueden ser la existencia de ciertos ficheros, una fecha particular, la ejecución de una aplicación concreta, etc. Una vez que la bomba explota, puede alterar o eliminar datos, parar el sistema, etc. Un ejemplo de uso de bomba lógica es el caso de un programador que vende un programa a una empresa. Si transcurrido un cierto tiempo la empresa no ha pagado, el programador revela la existencia de la bomba lógica con el fin de obtener su dinero.
- Puerta Falsa (Trapdoor): Es un punto de entrada secreto en un programa, de forma que alguien que conozca la existencia de dicha puerta puede obtener permisos de acceso sin tener que pasar por los mecanismos normales de autenticación. La puerta falsa es un código que reconoce alguna secuencia de entrada especial o se dispara si es ejecutado por cierto usuario o por la ocurrencia de una secuencia determinada de sucesos.
- Caballo de Troya (Trojan Horse): Es una rutina oculta en un programa de utilidad. Cuando el programa se ejecuta, se ejecuta la rutina y ésta realiza acciones no autorizadas y perniciosas. Estos programas permiten realizar de forma indirecta acciones que no puede realizar de forma directa. Por ejemplo, un programa caballo de troya puede ser un editor que cuando es ejecutado modifica los permisos de los ficheros que edita de forma que éstos puedan ser

accedidos por cualquier usuario. El autor del programa suele inducir a su utilización colocándolo en un directorio común y dándole un nombre de forma que aparente ser un programa de utilidad.

- Virus: Es código introducido en un programa que puede infectar otros programas mediante la copia de sí mismo en dichos programas. Además de propagarse, un virus realiza alguna función no permitida.
- Bacteria: Programa que consume recursos del sistema replicándose, asimismo, pero no daña explícitamente ningún fichero. Se suele reproducir exponencialmente, por lo que puede acaparar recursos como CPU, memoria y disco.
- Gusano (Worm): Es un programa que usa las redes de computadores para pasar de unos sistemas a otros. Una vez que llega a un sistema, el gusano se puede comportar como un virus o una bacteria, puede implantar programas caballo de troya, o puede realizar acciones no autorizadas. Para replicarse, los gusanos emplean algunos programas que proporcionan servicios de red, como correo electrónico, ejecución remota de programas y conexión a sistemas remotos.

Protección

La aparición de la multiprogramación introdujo en los computadores la posibilidad de compartir recursos entre varios usuarios. Entre los recursos compartidos están la CPU, la memoria, los dispositivos de entrada/salida, los programas y los datos. El hecho de poder compartir recursos es lo que introdujo la necesidad de protección.

Existen varios motivos por lo que la protección es necesaria. El más obvio es la necesidad de prevenir intentos de violación de restricciones de acceso por parte de un usuario. El más importante es, sin embargo, la necesidad de asegurar que cada proceso use los recursos del sistema de forma consistente de acuerdo con las políticas establecidas para el uso de esos recursos. Este requisito es fundamental para que un sistema sea fiable.

El papel de la protección en un sistema informático es proporcionar un conjunto de mecanismos que permita llevar a cabo las políticas que gobiernan el uso de los recursos. Estas políticas pueden ser fijas, establecidas en el diseño del sistema operativo, pueden ser formuladas por el administrador del sistema, o incluso puede ser establecidas por los usuarios.

Un principio importante es la separación entre política y mecanismo. Los mecanismos determinan hacer cierta cosa, mientras que las políticas deciden es lo que hay que hacer. En el ámbito de la protección, la política define qué datos han de ser protegidos y el mecanismo la forma de llevar a cabo la política. Esta separación favorece la flexibilidad. Debido a que las políticas pueden cambiar con el tiempo o con el lugar, es deseable que el sistema operativo ofrezca un conjunto general de mecanismos por medio de los cuales se puedan establecer políticas diferentes.

Dominios de Protección

Un computador contiene un conjunto de objetos que han de ser protegidos. Estos objetos pueden ser tanto hardware (CPUs, segmentos de memoria, discos, impresoras, etc.) como software (procesos, ficheros, semáforos, etc.). Cada objeto tiene un identificador único en el sistema y un conjunto de operaciones que le pueden ser aplicadas. Las operaciones posibles dependen del objeto. Por ejemplo, los segmentos de memoria pueden ser leídos o escritos, mientras que un fichero puede ser creado, abierto, leído, escrito, cerrado y borrado. El mecanismo de protección ha de prohibir a los procesos el acceso a aquellos objetos

Para estudiar los diferentes mecanismos de protección se introduce el concepto de dominio de protección. Un dominio de protección es un conjunto de pares objeto-derechos, de forma que cada par especifica un objeto y algún subconjunto de operaciones que se pueden realizar sobre dicho objeto.

En cada instante de tiempo, cada proceso se ejecuta dentro de un dominio concreto. Este dominio puede cambiar durante la vida del proceso.

Un dominio puede ser varias cosas:

-
- Cada usuario puede ser un dominio, en cuyo caso el conjunto de objetos que puede ser accedido depende de la identidad del usuario. Los cambios de dominio ocurren cuando se cambia de usuario.
 - Cada proceso puede ser un dominio, en cuyo caso el conjunto de objetos que puede ser accedido depende de la identidad del proceso. Los cambios de dominio se producen cuando un proceso envía un mensaje a otro y espera la respuesta.
 - Cada procedimiento puede ser un dominio, en cuyo caso el conjunto de objetos que puede ser accedido se corresponde con las variables locales definidas dentro del procedimiento. Los cambios de dominio se producen cuando se realiza una llamada a procedimiento.

El sistema operativo debe de conocer en todo momento los objetos que pertenecen a cada dominio. Una forma de llevar este control es tener una matriz de acceso, en la que las filas son los dominios y las columnas los objetos. Cada elemento de la matriz muestra los derechos de un dominio sobre un objeto. Los cambios de dominio se pueden incluir fácilmente en la matriz de protección considerando que cada dominio es a su vez un objeto sobre el que existe una operación que permite entrar en él. En la práctica no es viable almacenar la matriz de protección porque es grande y tiene muchos huecos (sparse matrix). La mayoría de los dominios no tiene acceso a la mayoría de los objetos, por lo que almacenar una gran matriz casi vacía desperdicia mucho espacio en disco. Dos soluciones prácticas consisten en almacenar la matriz, bien por filas, bien por columnas, y almacenar únicamente los elementos no vacíos. En el primer caso se habla de capacidades, y en el segundo, de listas de control de acceso.

Listas de Control de Acceso

Cuando la matriz se almacena por columnas, a cada objeto se le asocia una lista ordenada que contiene todos los dominios que pueden acceder al objeto y la forma de hacerlo. Esta lista se denomina lista de control de acceso (control access list o ACL).

El propietario de un objeto puede modificar la lista de control de acceso del mismo. El único problema es que los cambios pueden no afectar a los usuarios que ya estén usando el objeto.

Capacidades

La otra forma de almacenar la matriz de protección es por filas. Con este método, cada proceso tiene una lista de los objetos a los que puede acceder, así como una indicación de las operaciones que están permitidas sobre cada uno de ellos. Esta lista se denomina lista de capacidades (listas_C o C_lists), y cada elemento de la lista se llama capacidad.

Cada objeto está representado por una capacidad. Cuando un proceso quiere ejecutar una operación sobre un objeto, ha de especificar la capacidad del objeto como parámetro. La simple posesión de la capacidad permite que se conceda el acceso (siempre que éste esté permitido sobre el objeto).

Cada capacidad tiene un campo que indica el tipo del objeto, un campo de derechos de acceso, y un campo que identifica al objeto.

Las listas de capacidades se asocian a un dominio, pero los procesos que se ejecutan en ese dominio no deben de acceder a la lista de forma directa. Por el contrario, cada capacidad es un objeto protegido que debe ser gestionado por el sistema operativo y únicamente puede ser accedido de forma indirecta por parte de los procesos. Generalmente, suelen ser referenciadas por su posición dentro de la lista de capacidades. Las listas de capacidades son también objetos, por lo que pueden ser apuntadas desde otras listas, permitiendo el compartir subdominios.

11 Sistemas distribuidos. Entornos de computación

En un sistema operativo distribuido los usuarios pueden acceder a recursos remotos de la misma manera en que lo hacen para los recursos locales. La migración de datos y procesos de una instalación a otra queda bajo el control del sistema operativo distribuido.

Permiten distribuir trabajos, tareas o procesos, entre un conjunto de procesadores. Puede ser que este conjunto de procesadores esté en un equipo o en diferentes, en este caso es transparente para el usuario. Existen dos esquemas básicos de éstos. Un sistema fuertemente acoplado es aquel que comparte la memoria y un reloj global, cuyos tiempos de acceso son similares para todos los procesadores. En un sistema débilmente acoplado los procesadores no comparten ni memoria ni reloj, ya que cada uno cuenta con su memoria local.

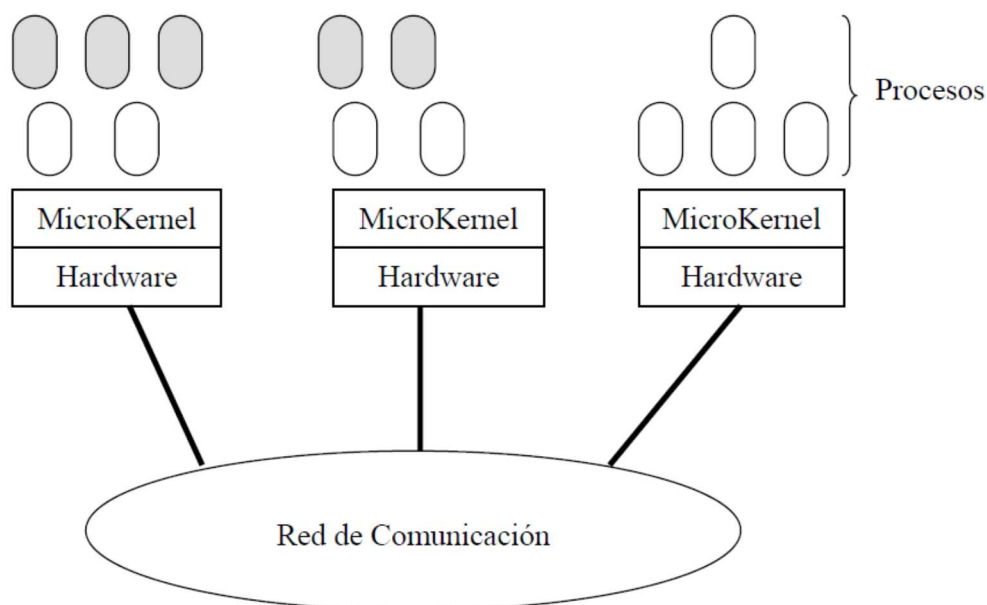
Los sistemas distribuidos deben de ser muy confiables, ya que si un componente del sistema se descompone otro componente debe de ser capaz de reemplazarlo.

Entre los diferentes Sistemas Operativos distribuidos que existen tenemos los siguientes: Sprite, Solaris-MC, Mach, Chorus, Spring, Amoeba, Taos, etc.

Características de los Sistemas Operativos distribuidos:

- Colección de sistemas autónomos capaces de comunicación y cooperación mediante interconexiones hardware y software.
- Proporciona abstracción de máquina virtual a los usuarios.
- Objetivo clave es la transparencia.
- Generalmente proporcionan medios para la compartición global de recursos.

Servicios añadidos: denominación global, sistemas de archivos distribuidos, facilidades para distribución de cálculos (a través de comunicación de procesos internodos, llamadas a procedimientos remotos, etc.).



Ventajas de los sistemas distribuidos:

- Economía: Buena relación rendimiento/coste
 - Avances en tecnología de microprocesadores y redes de área local.
- Alto rendimiento: Procesamiento paralelo.
- Soporte de aplicaciones inherentemente distribuidas.
 - Por ejemplo: empresa distribuida geográficamente
- Capacidad de crecimiento: Escalabilidad.
- Fiabilidad y disponibilidad: Tolerancia a fallos.
- Carácter abierto y heterogéneo:
 - Estándares de interoperabilidad.
- Compartir recursos y datos.

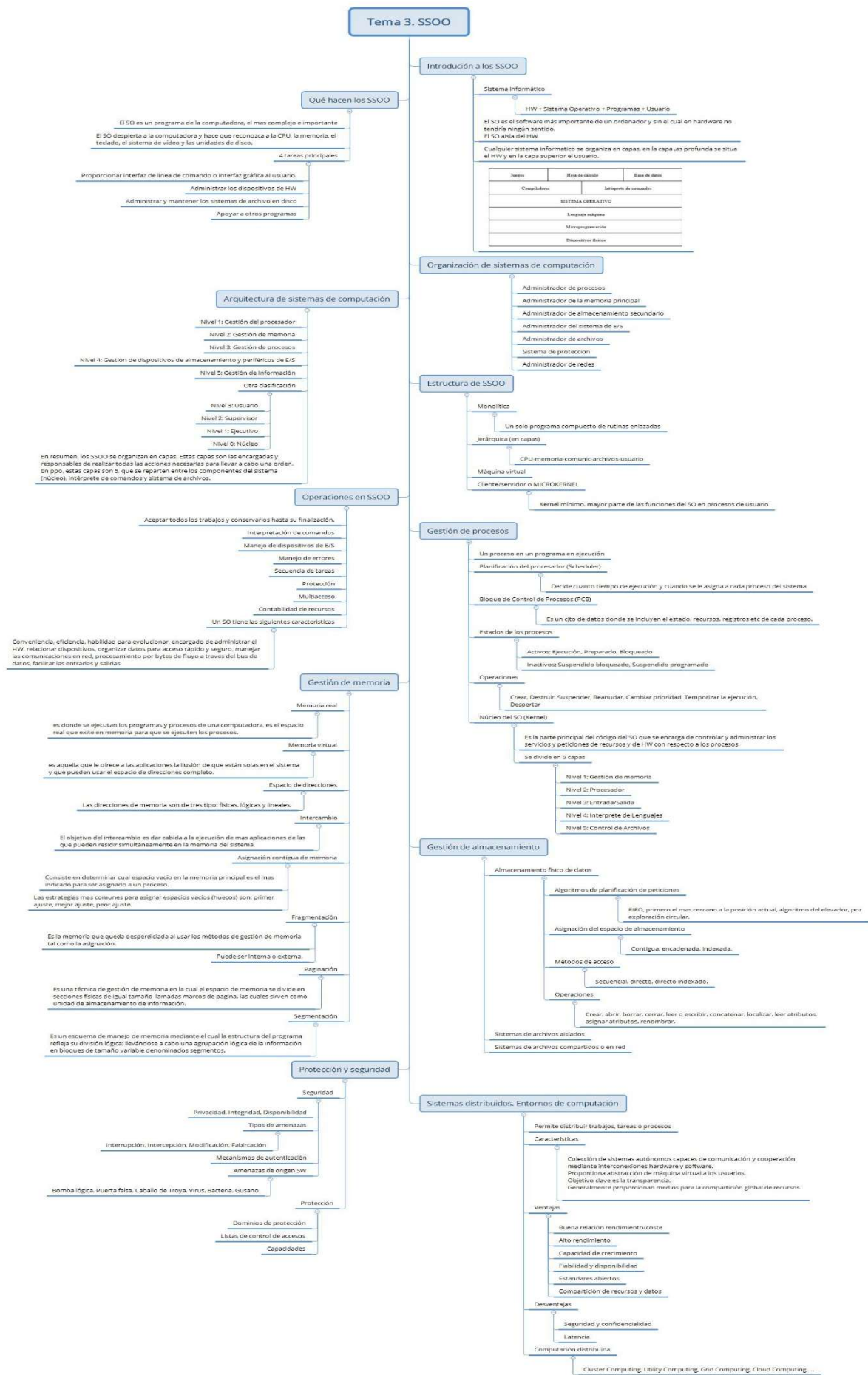
Desventajas de los sistemas distribuidos:

- Necesidad de un nuevo tipo de software:
 - Más complejo.
- Red de interconexión introduce nuevos problemas:
 - Pérdida de mensajes y saturación.
 - Latencia puede provocar que al recibir un dato ya esté obsoleto.
 - La red es un elemento crítico.
- Seguridad y confidencialidad
- Definición alternativa de SD:
 - “Un sistema distribuido es aquél en el que no puedes trabajar con tu máquina por el fallo de otra máquina que ni siquiera sabías que existía” (Lamport)

Modelos de computación distribuida:

- Cluster Computing
- Utility Computing
- Grid Computing
- Volunteer Computing
- Cloud Computing
- Autonomic Computing
- Mobile (Nomadic) Computing
- Ubiquitous (Pervasive) Computing

12 RESUMEN ESQUEMÁTICO



13 BIBLIOGRAFÍA BÁSICA

Deitel Sistemas Operativos

Stallings William. Sistemas Operativos. Editorial Prentice Hall

Tanenbaum Andrew S. Sistemas Operativos Modernos. Prentice Hall

Muñoz F.J. Sistemas Informáticos Monousuario y Multiusuario. Mc Graw Hill

Guevara Calume R.C. Informática Básica

Lluís Durán. Sistemas Operativos. Marcombo

Royo Vallés D. Sistemas Operativos. La Universidad Virtual