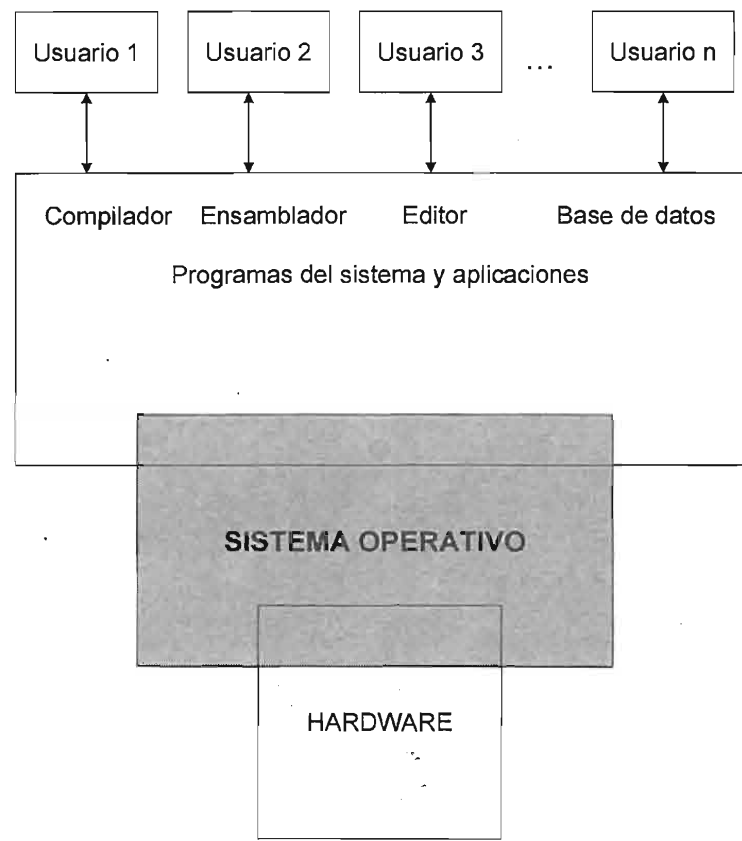


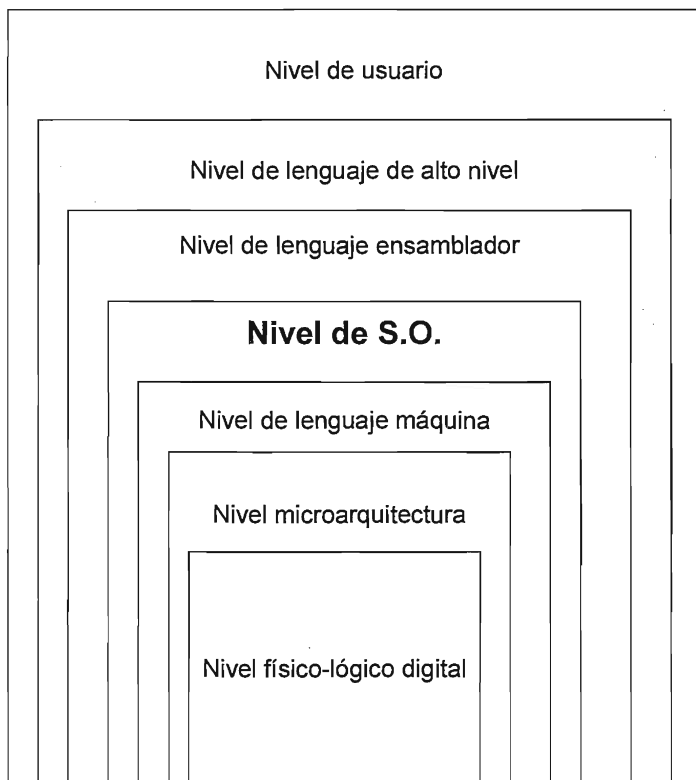
INTRODUCCIÓN

SISTEMA INFORMÁTICO
CONCEPTOS DE HARDWARE
EVOLUCION DE LOS S.O.
ESTRUCTUA DEL S.O.

SISTEMA INFORMÁTICO



CAPAS DE UN SISTEMA INFORMÁTICO

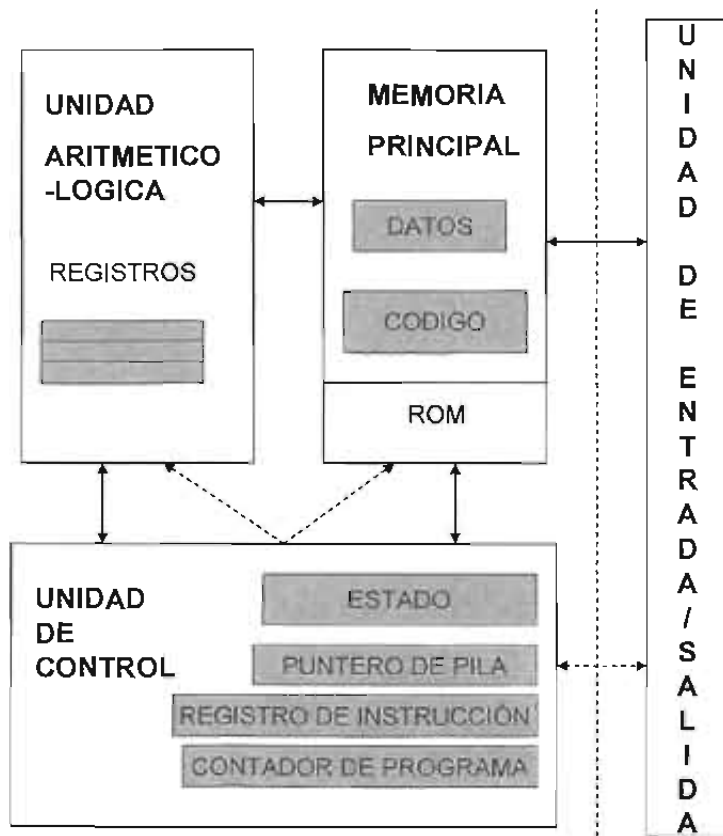


↳ HARDWARE

CAPAS DE UN SISTEMA INFORMÁTICO

- Nivel físico-lógica digital
 - Circuitos combinacionales y secuenciales (contadores, registros, biestables, puertas lógicas, etc.)
- Nivel microarquitectura
 - Se estudian los registros del procesador y la unidad aritmético-lógica
- Nivel de lenguaje máquina
 - Se estudian los códigos de las operaciones, modos de direccionamiento, tipos de datos, registros del procesador, registros de control de dispositivos de E/S, etc.
- Nivel de Sistema Operativo
 - Surge como necesidad de proteger los recursos del computador
- Nivel de lenguaje ensamblador
 - Es similar, conceptualmente hablando, al lenguaje máquina pero usa nemotécnicos para las instrucciones, macroinstrucciones y la posibilidad de servirse de los servicios ofrecidos por el S.O. que está debajo.
- Nivel de lenguaje de alto nivel
 - Su objetivo es facilitar el desarrollo de la programación. Se encuentra orientado hacia los problemas que trata y no hacia la máquina
- Nivel de usuario
 - En este nivel el usuario no necesita conocimientos informáticos sino entender el programa y el problema que trata

ESTRUCTURA DEL COMPUTADOR



ESTRUCTURA DEL COMPUTADOR

- **UNIDAD ARITMETICO-LOGICA**
 - Realiza los cálculos aritméticos y lógicos sobre uno o dos datos
 - Los datos con los que se realizan los cálculos se almacenan previamente en la memoria o en registros. Los resultados se llevan también a la memoria o a los registros.
 - La lógica de funcionamiento de las instrucciones la determina el lenguaje máquina o ensamblador de la máquina
- **MEMORIA**
 - RAM y ROM. En ella reside el programa máquina a ejecutar y los datos de entrada y salida del programa
- **UNIDAD DE CONTROL**
 - Se encarga de regular el funcionamiento de los distintos elementos para que funcionen coordinadamente

MODELO DE PROGRAMACIÓN

- Se caracteriza por los siguientes aspectos:
 - Elementos de almacenamiento
 - Registros generales
 - Contador de programa
 - Puntero/os de pila
 - Registro de estado
 - Memoria principal
 - Contadores de E/S
 - Juego de instrucciones
 - Define las operaciones que es capaz de hacer el ordenador.
 - Modos de ejecución
 - Modo usuario (tiene limitaciones: no permite realizar operaciones de E/S ni modificar parte del registro de estado ni los registros de soporte de gestión de la memoria)
 - Modo núcleo privilegiado (sin limitaciones)

MODELO DE PROGRAMACIÓN

- Secuencia de funcionamiento
 - La unidad de control se encarga de su funcionamiento.
 - Se trata de una secuencia que se repite sin cesar:
 - » Lectura de la instrucción apuntada por el PC
 - » Incremento del PC
 - » Ejecución de la instrucción
- Los elementos que pueden provocar un cambio de este ciclo son:
 - Instrucción máquina de salto o bifurcación
 - Interrupciones externas o internas. El contador del programa es modificado, saltando a otro programa (el S.O.)
 - Una instrucción máquina de llamada al sistema (p.ej.: TRAP) que provoca un efecto similar a la interrupción



PC = Contador de programa

INTERRUPCIONES

- Se activa con una señal que llega a la unidad de control
- Cuando llega la señal (debe encontrarse activada este tipo de interrupción), la unidad de control realiza un ciclo de activación de interrupción. Los pasos de este ciclo son:
 - Salva algunos registros del procesador
 - Eleva el modo de ejecución del procesador, pasándolo a núcleo
 - Carga un nuevo valor en el contador de programa
 - En muchos procesadores inhibe las interrupciones
- Los procesadores suelen incluir varias líneas de solicitud de interrupción, cada una de las cuales puede tener una determinada prioridad. Si se activan al mismo tiempo varias de estas líneas, se atiende la de mayor prioridad, quedando las demás a la espera. Las mas prioritarias suelen ser:
 - Excepciones hardware asíncronas
 - Excepciones hardware síncronas (de programa)
 - Interrupciones externas
 - Llamadas al sistema (TRAP)
- El procesador suele incluir un mecanismo de inhibición selectiva que permite detener todas o determinadas líneas de interrupción. La inhibición puede afectar a todas o a las de nivel inferior. Cuando se habilitan, las puede que se hayan perdido las que se produjeran durante la inhibición o puede que hayan sido encoladas.

INTERRUPCIONES

- Las interrupciones se pueden generar por diversas causas
 - Excepciones hardware síncronas (excepciones software)
 - Problemas de ejecución
 - Operación inválida en la unidad aritmética
 - División por cero
 - Operando no normalizado
 - Desbordamiento en el resultado
 - Resultado inexacto en la unidad aritmética
 - Dispositivo no existente
 - Región de memoria inválida
 - Desbordamiento de la pila
 - Violación de los límites de memoria asignada
 - Código de operación máquina inválido
 - Depuración
 - Punto de ruptura
 - Fallo de página
 - Excepciones hardware asíncronas
 - Error de paridad en el bus
 - Error de paridad en la memoria
 - Fallo de alimentación
 - Límite de temperatura excedido
 - Interrupciones externas. Se trata de interrupciones producidas por elementos externos al procesador:
 - El reloj
 - Los controladores de dispositivos de E/S
 - Otros procesadores
 - Instrucciones máquina de llamadas al sistema

¿QUÉ ES UN S.O.?

Un S.O. es un programa, pero es un programa un tanto especial o diferente de los demás en el sentido que realiza funciones distintas y no relacionadas y que se describen a continuación:

• EL S.O. COMO PROGRAMA

Decimos que el S.O. es un programa (o conjunto de ellos). Desde este punto de vista cumple con los esquemas elementales de todo programa informático (algoritmos + estructuras de datos).

Un S.O. es un programa en el cual los datos lo componen tablas de información acerca de los objetos que gestiona: recursos físicos (disco, impresoras, CPU... etc.) y recursos lógicos (procesos de usuarios, ficheros, tiempo de CPU... etc.) y la algoritmia es la lógica que se encarga de asignar de forma eficiente esos recursos físicos a los recursos lógicos.

• EL S.O. COMO MÁQUINA EXTENDIDA

Hemos dicho que la arquitectura de los ordenadores es difícil de programar, sobre todo para E/S. Una función fundamental del S.O. es ocultar estos detalles al usuario y presentar una interface de uso común para todos los usuarios. El programador no necesita conocer los detalles del disco u otro hardware para hacer uso de él pues el S.O. le proporciona esa información a través del software específico (controladores de hardware) de cada uno.

• EL S.O. COMO ADMINISTRADOR DE RECURSOS

Otra función muy importante es la de gestor de los recursos del sistema. Los programas utilizan todo tipo de hardware y es el S.O. el encargado de asignarlos a cada proceso de forma correcta.

HISTORIA DE LOS SISTEMAS OPERATIVOS (1)

- **1ª GENERACIÓN (1945-1955): TUBOS DE VACÍO Y TABLEROS DE CONMUTACIÓN**
 - A mediados de la década de 1940, Howard Aiken en Harvard, John von Neuman en el Institute for Advanced Study en Princeton. J. Presper Eckert y William Mauchley en la University of Pennsylvania y Konrad Zuse en Alemania, entre otros, lograron construir máquinas calculadoras usando tubos de vacío.
 - En estos primeros días, un solo grupo de personas diseñaba, programaba, operaba y mantenía la máquina. Toda la programación se realizaba en lenguaje máquina absoluto, a menudo alambrando tableros de comunicación para controlar las funciones básicas de la máquina.
 - No existían los lenguajes de programación (ni siquiera los ensambladores) ni, por supuesto, los S.O.
 - La forma de operar usual consistía en que el programador se anotaba para recibir un bloque de tiempo en la hoja de reservas, luego pasaba a la sala de máquinas, insertaba su tablero de conmutación en la máquina, y pasaba las siguientes horas esperando que no se fundiera ninguno de los 20.000 tubos de vacío.
 - Prácticamente todos los problemas eran cálculos numéricos directos.
 - A principios de la década de 1950, el trabajo había mejorado un poco con la introducción de las tarjetas perforadas. Ahora se podía escribir el programa en tarjetas e introducirlas para ser leídas, en lugar de usar tableros de conmutación; por lo demás, el procedimiento era el mismo.

HISTORIA DE LOS SISTEMAS OPERATIVOS (2)

2ª GENERACIÓN (1955-1965): TRANSISTORES Y SISTEMAS POR LOTE

A mediados de 1950 se introdujo el transistor en los computadores y esto cambió el panorama totalmente. Aumenta la fiabilidad y permite que se vendan a particulares con garantía de poder hacer un trabajo seguro. Aparece la separación entre diseñadores, constructores, operadores, programadores y personal de mantenimiento.

Estas máquinas tenían equipos profesionales para operarlas. Como ejemplo un operador se encargaba del montaje de la máquina y liberaba al programador de la tarea.

Para ejecutar un trabajo (un programa o serie de ellos), un programador escribía primero el programa en papel (FORTRAN o ensamblador) y luego lo perforaba en tarjetas. Después, llevaba el grupo de tarjetas al cuarto de entrada y lo entregaba a uno de los operadores. Cuando este acababa el resultado era enviado a un cuarto de salida donde el programador recogía los resultados del trabajo. Si el siguiente trabajo necesitaba de otro compilador, había que cargarlo; esto implicaba un desperdicio de tiempo.

La solución a esto fue el sistema por lotes. El principio de esta forma de trabajo consistía en reunir una serie de trabajos en el cuarto de entrada, leerlos y grabarlos en una cinta magnética usando una computadora pequeña (IBM 1401, útil para leer tarjetas, copiar cintas e imprimir salidas, pero no para cálculos) y luego procesarlos en una más grande (IBM 7094).

Como se procesaban varios trabajos similares seguidos, se ideó un programa que residía en memoria cuya función primordial consistía en activar el siguiente trabajo después de finalizar el anterior es decir, se encargaba de la secuencia automática de trabajos. Este programa conocido como monitor residente fue el origen de lo que hoy son los Sistemas Operativos.

HISTORIA DE LOS SISTEMAS OPERATIVOS (3)

Este primitivo S.O. o monitor residente tenía la siguiente estructura:

CARGADOR
SECUENCIADOR DE TRABAJOS
INTERPRETE DE TARJETAS DE CONTROL
ÁREA DE PROGRAMAS DE USUARIO

Una vez que estaban finalizados los trabajos, se pasaba la cinta de salida de nuevo a la IBM 1401 para efectuar la salida impresa fuera de línea (no conectada a la computadora principal).

Las computadoras grandes de la segunda generación se usaban primordialmente para cálculos científicos y de ingeniería. Estas máquinas generalmente se programaban en FORTRAN y lenguaje ensamblador. Los sistemas operativos típicos eran FMS (Fortran Monitor System) e IBSYS (el S.O. de IBM para la 7094).

HISTORIA DE LOS SISTEMAS OPERATIVOS (4)

3G. (1965-1980): CIRCUITOS INTEGRADOS Y MULTIPROGRAMACIÓN

A principios de la década de 1960, la mayoría de los fabricantes tenían dos líneas de producción, una orientada al trabajo científico y otra orientada a los aspectos administrativos. Para paliar este problema IBM trató de crear una máquina que sirviera para ambos tipos de trabajo e ideó la System/360. Creó una variedad de modelos para que sirvieran a diferentes propósitos (tanto en tamaño como en naturaleza) y que diferían sólo en el precio y en el rendimiento.

Puesto que todas las máquinas tenían la misma arquitectura y conjunto de instrucciones, los programas escritos para una máquina podían ejecutarse en todas las demás, al menos en teoría.

La 360 fue la primera computadora en usar circuitos integrados (IC) y esto le dio una ventaja competitiva muy fuerte frente a la competencia. Esta idea le dio un éxito inmediato y todos los competidores adoptaron esta estrategia.

Esta idea de tener "una familia" fue también su gran debilidad. La idea era que todo el software, incluido el sistema operativo, funcionara en todos los modelos. El software tenía que funcionar en máquinas potentes y no tanto, con procesos orientados al cálculo científico y con otros orientados al trabajo administrativo, con pocos y con muchos periféricos... etc., si añadimos que además se le pedía que lo hiciera siempre de forma eficiente, podemos vislumbrar el origen de los problemas con los que se encontró el sistema operativo.

HISTORIA DE LOS SISTEMAS OPERATIVOS (5)

La pretensión de dar respuesta a todas estas necesidades dio origen a un S.O. enorme, muy complejo y con millones de líneas en ensamblador escritas por diferentes programadores y que contenía miles de errores. Cada intento de corregirlos daba lugar a otra tanda de ellos y sin visos de solución definitiva.

A pesar de todo dio respuesta "satisfactoria" a la mayoría de los clientes y sirvió para introducir en los S.O. una forma de gestión de los procesos conocida como multiprogramación.

En la 7094, cuando un trabajo hacía una pausa para hacer una operación de E/S, la CPU permanecía ociosa esperando a que finalizara dicha operación. En cálculos científicos la pérdida de tiempo no es exagerada pero en el procesamiento de datos comerciales esto puede significar el 80-90% del tiempo total.

La decisión que se tomó para solucionar el problema fue dividir la memoria en varias secciones, con un trabajo distinto en cada partición. Mientras un trabajo estaba esperando por una operación de E/S otro podía estar usando la CPU. Si se consigue mantener en memoria diversos programas podría mantenerse la CPU trabajando casi todo el tiempo.

Mantener en memoria más de un programa implica tener un hardware especial que impida que un programa invada la partición de otros.

El procesamiento off-line permitió simultanear las operaciones de E/S con el uso de la UCP. Pero si queremos que este trabajo se realice sobre una sola máquina; se debe colocar ciertos elementos entre los dispositivos y la UCP para permitir una separación similar de la ejecución. Además es necesario desarrollar una arquitectura adecuada para permitir este almacenamiento temporal

BUFFER

Un buffer, o almacenamiento temporal, es un método para hacer simultánea la E/S de un trabajo con su propio cómputo. Consiste en reservar espacio de memoria para traer o llevar información del dispositivo de E/S, así el trabajo se activa porque la lectura o escritura sobre el dispositivo puede hacerse en un momento diferente al de su utilización lógica por la UCP.

El manejo de buffers es generalmente una función del sistema operativo. El monitor residente o los manejadores de dispositivo incluyen buffers del sistema para cada dispositivo de E/S.

Los trabajos de esta época se caracterizaban por estar limitados por E/S con lo que la utilización de buffers no era suficiente.

SPOOLING

La utilización del disco supuso una evolución importante para los sistemas operativos.

Al disponer de un acceso directo a la información, la utilización de la misma no es imperativo que se haga de forma secuencial. Esto va a permitir que se pueda simultanear la utilización de un dispositivo de E/S de un programa con el procesamiento de otro programa. A esta forma de trabajo se la conoce como spool (simultaneous peripheral operation on-line). Fundamentalmente se trata de utilizar el disco como un gran buffer de almacenamiento masivo.

El spooler puede llevarse al extremo de llevar a disco toda una cinta y gravar la misma de nuevo al disco, posteriormente se imprime la salida. A esta operación se la conoce como pasar por etapas (staging) la cinta.

La utilización del spooling proporciona una estructura de datos muy importante: el deposito de trabajos (job pool). Al disponer de varios trabajos almacenados en disco, se puede seleccionar el orden de entrada (con la cinta el orden es secuencial) y mejorar el rendimiento del sistema, esto hace necesaria una planificación del trabajo.

MULTIPROGRAMACIÓN

Creado el software que permita un control del spool es lógico pensar que el siguiente paso es poder simultanear la ejecución de diversos trabajos la UCP. Teniendo en cuenta que el paso significa cambiar de un proceso a otro por abandono del anterior de la UCP. Esto paso implica para el sistema operativo la necesidad de determina cual es el proceso al que se le va a asignar la UCP, esta decisión se conoce como **planificación de la UCP.**

TIEMPO COMPARTIDO

El tiempo compartido (o multitarea) es una extensión lógica de la multiprogramación. La UCP ejecuta las diversas tareas alternando entre ellas pero los cambios se hacen por iniciativa del sistema operativo, el cual va asignando tiempo (quantum) a cada proceso y va permitiendo que todos se vayan ejecutando "simultáneamente". Esto implica la necesidad de una mejor planificación de trabajos para decidir cuestiones como: ¿a quien se le asigna la CPU?, ¿duración del quantum? ...etc.

Además, permite una interacción entre el programa y el usuario. Antes los programas se ejecutaban de forma secuencial y esto tenía implicaciones para los programadores pues tenían que prever todas las posibilidades de desarrollo de los programas.

ESTRUCTURA DEL SISTEMA OPERATIVO

- **ADMINISTRACIÓN DE PROCESOS**
 - Una de las tareas primordiales de todo S.O. es la ejecución de programas del usuario. Pero en el sistema existen otros programas en ejecución que no son los creados por los usuarios. En un sistema con diversos procesos para su ejecución hace falta llevar una gestión de los mismos, de ésta se encarga el S.O.
 - Un proceso necesita determinados recursos: tiempo de CPU, memoria, archivos, dispositivos de E/S etc.
 - Un S.O. es responsable de las siguientes actividades relacionadas con la administración de procesos:
 - Crear y eliminar los procesos de usuario y del sistema.
 - Suspender y reanudar la ejecución de los procesos.
 - Proporcionar mecanismos para la sincronización de los procesos.
 - Proporcionar mecanismos para la comunicación de los procesos.
 - Proporcionar mecanismos para el manejo de bloques mutuos.

ESTRUCTURA DEL SISTEMA OPERATIVO (2)

- **ADMINISTRACIÓN DE MEMORIA PRINCIPAL**
 - La memoria es un depósito de datos de acceso rápido compartido por la UCP y los dispositivos de E/S. La UCP sólo tiene acceso a la memoria para los programas y datos que utiliza, es por ello que deben encontrarse ambos en la misma antes de ser utilizados. Si existen diversos procesos en ejecución es evidente que deberían encontrarse en la memoria los diferentes códigos y datos que se vayan a usar. Se hace necesario gestionar el espacio para la ubicación de los mismos.
 - El S.O. es responsable de las siguientes actividades relacionadas con la administración de la memoria:
 - Llevar un control de cuáles son las zonas de memoria que se están usando y quién las usa.
 - Decidir qué procesos se cargarán en memoria cuando haya espacio disponible.
 - Asignar y recuperar el espacio en memoria según se requiera.

ESTRUCTURA DEL SISTEMA OPERATIVO (3)

- **ADMINISTRACIÓN DE ALMACENAMIENTO SECUNDARIO**
 - Los programas y datos se encuentran almacenados en dispositivos de almacenamiento masivo (discos). De ellos son traídos a memoria y a ellos son llevados después de ser usados.
 - Es responsabilidad del S.O. las siguientes actividades relacionadas con la administración del disco:
 - Administración del espacio libre.
 - Asignación del almacenamiento.
 - Planificación de las operaciones sobre el disco.

ESTRUCTURA DEL SISTEMA OPERATIVO (4)

- **ADMINISTRACIÓN DE E/S**
 - Uno de los objetivos del S.O. es ocultar al usuario las particularidades de los dispositivos del hardware. Por ejemplo, en UNIX, las particularidades de los dispositivos de E/S se ocultan del resto del sistema operativo por medio de un sistema de E/S, el cual consiste en:
 - Un sistema de memoria caché mediante buffers.
 - Una interfaz general con los manejadores de dispositivos.
 - Manejadores para dispositivos de hardware específicos (cuya función es conocer las particularidades del dispositivo).

ESTRUCTURA DEL SISTEMA OPERATIVO (5)

- **ADMINISTRACIÓN DE ARCHIVOS**
 - La administración de archivos puede considerarse uno de los componentes más visibles de un S.O.
 - Para usar cómodamente un sistema de computación, el S.O. ofrece una perspectiva lógica uniforme de almacenamiento de la información. Un S.O. se vale de las propiedades físicas de sus dispositivos de almacenamiento para definir una unidad de almacenamiento lógico: el **archivo**. Por medio del S.O., los archivos se relacionan con los dispositivos físicos.
 - El S.O. es responsable de las siguientes actividades relacionadas con la administración de archivos:
 - La creación y eliminación de archivos.
 - La creación y eliminación de directorios.
 - El manejo de operaciones primitivas para manipular archivos y directorios.
 - La copia de seguridad de archivos en medios de almacenamiento estables.

ESTRUCTURA DEL SISTEMA OPERATIVO (6)

- **SISTEMA DE PROTECCIÓN**
 - Los distintos procesos de un S.O. deben ser protegidos unos de otros. Los segmentos de memoria, la UCP, los archivos y otros recursos pueden ser usados únicamente por los procesos que han recibido la correspondiente autorización.
 - Por ejemplo, el hardware de direccionamiento a memoria asegura que un proceso sólo puede trabajar dentro de su espacio de direcciones. El cronómetro asegura que ningún proceso puede obtener el control de la UCP sin perderlo eventualmente. Por último no se permite que un usuario realice por su cuenta operaciones de E/S, para proteger así la integridad de los dispositivos físicos.

ESTRUCTURA DEL SISTEMA OPERATIVO (7)

- **EL INTÉRPRETE DE COMANDOS (SHELL)**
 - Uno de los programas de sistema más importantes para el usuario es el intérprete de comandos o shell.
 - Algunos S.O., en especial los de microcomputadores, incluyen al intérprete de comandos en su núcleo. Otros S.O., sobre todo en las computadoras de gran tamaño, tratan al intérprete de comandos como un programa especial que se ejecuta al iniciar un trabajo o cuando el usuario se conecta por primera vez al sistema (en sistemas de tiempo compartido).

SERVICIOS DEL SISTEMA

- Un S.O. no debe considerarse hoy en día como una herramienta que sólo gestiona los recursos del sistema sino que además proporciona una serie de servicios que hace más cómoda la labor del usuario de un sistema informático. Para ello suelen proporcionar una serie de utilidades que, si bien no pueden considerarse S.O. en el sentido que nos interesa a nosotros, se consideran utilidades imprescindibles en todo sistema informático y suelen venir a disposición del usuario con el S.O.. estas podemos dividir las en:
 - **Manipulación de archivos.** Estos programas crean, eliminan, copian, renombran, imprimen, vuelcan, muestran y en general manipulan archivos y directorios.
 - **Información de estado.** Algunos programas simplemente solicitan al sistema fecha, hora, espacio de memoria o disco disponible, número de usuarios... etc.
 - **Modificación de archivos.** Es normal que exista algún editor que permita la manipulación de ficheros en sus aspectos básicos.
 - **Apoyo a lenguajes de programación.** Es usual que se ofrezcan compiladores, intérpretes, ensambladores para los lenguajes más habituales. Hoy en día se ofrecen por separado.
 - **Carga y ejecución de programas.** Una vez ensamblado o compilado el programa, se debe cargar en la memoria para ejecutarlo. El sistema puede ofrecer cargadores absolutos, cargadores relocables, editores de enlace y cargadores de superposición.
 - **Comunicaciones.** Estos programas proporcionan el mecanismo para crear conexiones virtuales entre procesos, usuarios y diferentes sistemas de computación; permiten a los usuarios enviar mensajes a la pantalla de los demás, transferir archivos entre máquinas, enviar correo y conectarse en forma remota a otros computadores.
 - **Programas de aplicación.** Es normal que se entreguen con el S.O. programas como: generadores de gráficos, sistemas de Bases de datos, juegos... etc.