



Universidad Católica de Temuco
Escuela de Informática

Apuntes Sistemas de Operativos

Primera Parte

Alejandro Mauricio Mellado Gatica
Magíster en Telecomunicaciones

Abril 2003

Índice de contenido

Conceptos Básicos.....	4
Componentes de un sistema computación.....	4
El Hardware.....	4
El Sistema Operativo.....	4
Las Aplicaciones.....	4
Herramientas de escritorio.....	4
Utilitarios de Mantenimiento.....	4
Sistema de Bases de Datos.....	5
La red.....	5
Evolución de los Sistemas.....	5
Antecedentes	5
Generaciones de sistemas según el Profesor Tanenbaum.....	5
Primera generación (1945-1955).....	5
Segunda generación.....	6
Tercera generación (1965-1980).....	6
Cuarta generación (1980-1990).....	8
Algunos Conceptos y Componentes Básicos.....	9
Componentes de hardware.....	9
La memoria entrelazada.....	9
Registro de reubicación.....	9
Interrupciones y escrutinio.....	10
Los buffers.....	10
Protección del almacenamiento.....	10
Cronómetros y relojes.....	10
Canales de Entrada y Salida.....	11
Memoria virtual.....	11
Robo de ciclos.....	11
Multiprocesamiento.....	12
Acceso directo a memoria.....	12
Software.....	13
Lenguaje de Máquina.....	13
El uso de spools.....	13
Compiladores rápidos y sucios vs. Compiladores optimizados.....	13
Intérpretes.....	13
Firmware.....	14
Componentes de un Sistema Operativo.....	14
Procesos.....	14
Los sistemas de Archivos.....	15
Interprete de comandos (Shell).....	15
La Interfaz gráfica de usuario.....	16
Llamadas al Sistema.....	16
Llamadas al sistema para manejo de procesos.....	16
Llamadas al sistema para señalización.....	16
Llamadas el Sistema para manejo de archivos.....	17
Procesos.....	18

Modelo del proceso.....	18
Jerarquía de procesos.....	20
Estados de los Procesos.....	20
Implementación de Procesos.....	21
Comunicación de Procesos.....	21
Referencias.....	22

Componentes de un sistema computación

Un sistema de computación esta compuesto por un conjunto de elementos que se dividen en dos componentes: El hardware que es la componente física del sistema (la parte tangible) y el Software que corresponde a la componente lógica del sistema (los programas).

El Hardware

El hardware es la componente física de un sistema de computación. En esta componente están incluidos los computadores de distintas arquitecturas, los periféricos tales como impresoras o escáneres, equipos de comunicación y toda maquina que tenga capacidad de procesamiento.

El Sistema Operativo

Un sistema operativo es un administrador de recursos, a que controla todos los recursos de hardware presentes en un computador (procesador, medios de almacenamiento, dispositivos de entrada y salida, dispositivos de comunicación de datos, etc). Otras funciones del sistema operativo son: Proporcionar una interfaz de interacción con el usuario, distribuir la carga de trabajo entre los usuarios y servir de plataforma de software a los programas de aplicación. Los sistemas operativos ha sido contruidos para ser ejecutados en una arquitectura de computador específica.

Las Aplicaciones

Las aplicaciones son el último nivel en la pirámide la de tecnología del software, se denominan aplicaciones a todos los programas de alto nivel que interactúan en forma directa o indirecta con el usuario.

Herramientas de escritorio

Las herramientas de escritorio son un conjunto de aplicaciones de orientadas al trabajo de oficina. Hoy en día en un conjunto de herramientas de escritorio encontramos procesadores de texto, planillas electrónicas, programas de presentaciones, organizadores, clientes de correo, etc.

Utilitarios de Mantenimiento

Son programas utilitarios de mantenimiento de un sistema todas aquellas herramientas que no ayudan a administrar el sistema operativo, dentro de estos programa encontramos defragmentadores, comprobadores y reparadores de disco duro (scandisk, fsck, etc.), optimizadores de memoria.

Sistema de Bases de Datos

Una base de datos es un conjunto ordenado de datos organizados de tal forma que estos se pueden consultar, ingresar, mantener y actualizar de manera fácil, eficiente y ordenada.

La base de datos puede concebirse como una colección de archivos interrelacionados, son creados con un DBMS. El contenido de una base de datos engloba a la información concerniente(almacenadas en archivos) de una organización, de tal manera que los datos estén disponibles para los usuarios, una finalidad de la base de datos es eliminar la redundancia o al menos minimizarla. Los tres componentes principales de un sistema de base de datos son el hardware, el software DBMS y los datos a manejar, así como el personal encargado del manejo del sistema.

Sistema Manejador de Base de Datos (DBMS) es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de una tarea específica. El objetivo principal de un sistema manejador base de datos es proporcionar un contorno que sea a la vez conveniente y eficiente para ser utilizado al extraer, almacenar y manipular información de la base de datos. Todas las peticiones de acceso a la base, se manejan centralizadamente por medio del DBMS, por lo que este paquete funciona como interfase entre los programas de usuario y la base de datos.

La red

No tan solo la red es el medio de comunicación, sino que es un conjunto de elementos que hacen posible la existencia de los sistemas distribuidos mediante el intercambio y procesamiento de datos. La red desde el punto de modelo de interconexión de sistemas abiertos (OSI) las comunicaciones de datos pueden ser vistas como un conjunto de siete capa iniciándose de los elementos físicos (el hardware) hasta los elementos de software de alto nivel. (Para más información Apuntes de Redes I).

Evolución de los Sistemas

Antecedentes

Como referencia la primera computadora digital real fue diseñada por Charles Babbage (1792-1871). Fue concebida bajo la perspectiva de una maquina analítica, en ese tiempo difícil de construir, ya que no se podían fabricar las piezas que necesitaba la maquina con precisión.

Generaciones de sistemas según el Profesor Tanenbaum

Primera generación (1945-1955)

Tubos de vacío y tableros enchufables.

Después de los esfuerzos de Babbage, se progreso poco en la construcción de computadores digitales hasta la segunda guerra mundial. A mediados de los 40 del siglo pasado, John Von

Neuman y Howard Aiken entre otros obtuvieron óptimos resultados en la construcción de máquinas de cálculo mediante el uso de tubos al vacío. Estas máquinas eran enormes que ocupaban cuartos enteros con decenas de tubos de vacío.

En esa época, un grupo de personas diseñó, construyó, programó, operó y dio mantenimiento a cada máquina. Toda la programación se realizó en lenguaje de máquina absoluto, a menudo alambRANDO tableros enchufables para controlar las funciones básicas de la máquina. (Los lenguajes de programación no se conocían).

Al inicio del año 1950, la rutina había mejorado un poco con la introducción de las tarjetas perforadas. Luego, entonces era posible escribir programas en tarjetas y leerlos, en vez de utilizar tableros enchufables, los demás procedimientos seguían iguales.

Segunda generación

Transistores y procesamiento por lotes.

La introducción del transistor a mediados de la década de 1950 cambió la imagen radicalmente. Los computadores se volvieron lo suficientemente confiables que podían fabricarse y venderse a clientes con la esperanza que continuarían funcionando el tiempo suficiente para realizar un trabajo útil. Esta máquina se instalaba en cuartos de computadores especialmente acondicionados con aire, con equipos de operadores profesionales para operarlas. Sólo las grandes corporaciones o bien las oficinas principales del gobierno o Universidades podían darse el lujo de poseer una de estas computadoras.

Para correr un trabajo, un programador primero escribía el programa en papel en Fortran o ensamblador y después lo perforaría en tarjetas. Luego llevaría la pila de tarjetas al cuarto de introducción al sistema y la entregaría a uno de los operadores.

Tercera generación (1965-1980)

Circuitos Integrados (CI) y Multiprogramación.

A inicios de los años 60 del siglo XX, muchos fabricantes de computadores tenían dos líneas de productos distintos y totalmente incompatibles, por un lado existían los computadores científicos de gran escala orientados a las palabras, como la 7094 de IBM, que se usaban para realizar cálculos numéricos de ciencia e ingeniería. Por otro lado, estaban los computadores comerciales orientados a los caracteres, como la 1401, que se utilizaba ampliamente para ordenamiento de cintas e impresión por parte de bancos y compañías de seguro.

El desarrollo y mantenimiento de estas dos líneas de computadores era una proposición costosa para los fabricantes. Además, muchos nuevos compradores de computadores necesitaban una máquina pequeña, pero después al crecer requerían de una máquina de mayor capacidad (más grande) que ejecutara todos los programas de la máquina anterior pero con mayor velocidad de procesamiento.

IBM intentó resolver estos problemas introduciendo en el mercado el sistema/360. El 360 era una serie de máquinas compatibles con el software que variaban del tamaño de una 1401 a una mucho más poderosa que la 7094. La diferencia de las máquinas estaba en el precio y

rendimiento. El sistema/360 estaba diseñado para realizar cálculos tanto científicos como comerciales.

A pesar de sus problemas el OS/360 y los sistemas operativos similares de tercera generación producidos por otros fabricantes de computadores satisfacían a sus clientes razonablemente bien. También se popularizaron varias técnicas ausentes en los sistemas de segunda generación que mejoraron bastante el rendimiento de los sistemas, una de la técnicas más importante es la Multiprogramación. En la máquina IBM 7094, cuando un trabajo corriente se detenía para esperar que se completará una operación en cinto u otra operación de entrada y salida (E/S), la Unidad Central de Procesos (CPU) permanecía ociosa hasta que terminará el proceso de entrada y salida. Con los cálculos científicos destinados a la CPU, la E/S es poco frecuente, de modo que el tiempo de espera de E/S puede ser con frecuencia del 80% o 90% del tiempo total, de modo que tenía que hacerse algo al respecto.

La evolución que solucionaba el problema consistía en partir la memoria en varias partes, con trabajos diferentes en cada partición. Mientras que un trabajo esperaba que se completara la E/S, otro trabajo podía estar utilizando la CPU. Si podían mantenerse suficientes trabajos en la memoria al mismo tiempo, la CPU podía mantenerse ocupada casi en un 100% del tiempo.

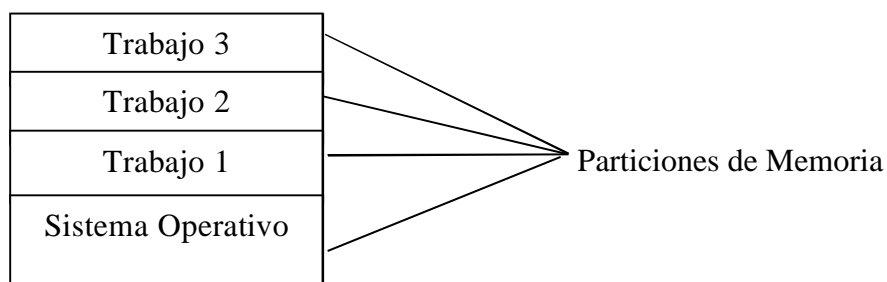


Figura 1. Sistema de Multiprogramación con 3 trabajos en memoria

Otra característica de importancia en los sistemas operativos de la tercera generación, era la capacidad de leer trabajos de tarjetas contenidas en el disco, tan pronto como se llevaba al cuarto de computación. Por lo tanto, siempre que se terminaba un trabajo, el sistema podía cargar uno nuevo del disco en la partición no vacía y ejecutarlo. Esta técnica se denomina manejo de cola de impresión.

Con los sistemas de tercera generación, el tiempo entre la entrega de un trabajo y la devolución de la salida comprendía a menudo varias horas, un solo error de sintaxis podía hacer que la compilación fallase y el programador perdiera medio día.

El deseo de obtener un tiempo de respuesta corto, marcó el camino para el tiempo compartido, variante de la multiprogramación, en el cual cada usuario tiene un terminal en línea.

El primer sistema de tiempo compartido (CTSS) serio, fue creado en el MIT en una unidad 7094 especialmente modificada.

Después del éxito del sistema CTSS, el MIT, los laboratorios Bell y General Electric decidieron embarcarse en el desarrollo de una “computadora de servicio público”, máquina que soportaría miles de usuarios simultáneamente (tiempo compartido). Los diseñadores de este sistema llamado MULTICS (Multiplex Information and Computing Service, Información Multicanalizada y Servicio de Información) imaginaron una máquina inmensa que ofrecía poder de cómputo para todos los habitantes de Boston.

Otro avance importante durante la tercera generación fue el crecimiento de los minicomputadores, comenzando con la DEC PDP-1 en 1961.

Uno de los científicos de computación de los laboratorios Bell que había trabajado en el proyecto Multics, Ken Thompson, halló después una pequeña PDP-7 que nadie utilizaba y empezó a escribir una versión reducida de Multics para un usuario. *Brian Kernighan dudaba un tanto en broma de este sistema de denominarlo UNICS (Uniplex Information and Computing Services, Información Unicanalizada y Servicio de Computación), más tarde pasaría a llamarse UNIX.* Después este nuevo sistema se trasladaría a una PDP-11/20, donde trabajó suficientemente bien como para convencer a la administración de los laboratorios Bell de invertir en una PDP 11/45 mayor para continuar el trabajo.

Después otro científico de los laboratorios Bell, Dennis Ritchie, unió esfuerzos con Thompson para reescribir el sistema en un lenguaje de alto nivel llamado C, diseñado e implementado por Ritchie. Los laboratorios Bell le dieron la licencia Unix a las Universidades casi gratuitamente y al cabo de uno o dos años cientos de ellas lo utilizaban. Unix pronto se migró al Interdata 7/32, al VAX, al Motorola 68000 y muchos otros computadores. Unix es el sistema que se ha migrado a más computadores que ningún otro sistema operativo en la historia y su uso sigue aumentando.

Cuarta generación (1980-1990)

Computadores Personales

Con la creación de los LSI (Integración a Gran Escala), chips que contienen miles de transistores en un centímetro cuadrado de silicio, se inicia la era del computador personal. Donde el minicomputador hizo posible que un departamento de una empresa o una Universidad tuviera su propia computadora, el chip microprocesador hizo posible que un individuo tuviera su propio computador personal.

La vasta diseminación del poder de cómputo, especialmente el poder de cómputo altamente interactivo generalmente con gráficas excelentes, condujo al desarrollo de una nueva industria que, la industria del software. La que produciría software para computadores personales. Gran parte de este software era “amigable” con el usuario, lo que significaba que estaba dirigido a usuarios que no conocían los computadores y además no tenían el interés de aprender (en el caso de los usuarios esto se ha extendido hasta nuestros días).

Dos sistemas operativos dominaban la escena del computador personal: el MS-DOS, escrito por Microsoft Inc. (derivado del CPM), orientado a los computadores IBM,

compatibles (que utilizaban la CPU 8088) y sus sucesores (x86); y el Unix, que dominaba en los computadores personales mayores que hacían uso de la familia de procesadores 68000. Aunque la versión inicial de MS-DOS era relativamente primitiva, las versiones subsiguientes fueron incluyendo características de Unix (Como referencia Microsoft era proveedor de un Unix llamado Xenix).

Un avance interesante empezó a tomar su sitio a mediados de los 80 del siglo XX, es el desarrollo de las redes de computadores personales que corren sistemas operativos en red y sistemas operativos distribuidos. En un sistema operativo en red los usuarios tienen el conocimiento de la existencia de múltiples computadores y se puede entrar a máquinas remotas y copiar archivos de una máquina a otra. Cada máquina ejecuta su sistema operativo local y tienen un o más usuarios propios.

Un sistema operativo distribuido, es aquel que se presenta ante sus usuarios como un sistema uni-procesador tradicional, aunque en realidad este compuesto por múltiples procesadores. En un sistema distribuido real, los usuarios no tienen conocimiento de donde se están ejecutando sus programas o donde están ubicados sus archivos; todo esto se debe manejar de manera automática y eficiente por el sistema operativo.

Algunos Conceptos y Componentes Básicos

Como se ha mencionado anteriormente, un sistema de computación esta compuesto por hardware y software. Ha estos dos elementos se le ha agregado un tercer elemento llamado Firmware, que esta compuesto por los programas en microcódigo almacenados en una memoria de control de alta velocidad. Al firmware también se atribuyen los programas objetos de uso común colocados en memorias de solo lectura (ROM y PROM). A continuación se hará una revisión general de las componentes de hardware, software y firmware.

Componentes de hardware

La memoria entrelazada

El entrelazado de memoria se usa para acelerar el acceso al almacenamiento primario. Normalmente, mientras se obtiene acceso a alguna de las localidades de un banco de memoria primario no pueden existir otras referencias en proceso. El entrelazado de memoria coloca las localidades de memorias contiguas en diferentes bancos de almacenamiento, de manera que puedan existir en proceso muchas referencias al mismo tiempo.

Registro de reubicación

Un registro de reubicación permite re-ubicar programas de manera dinámica. La dirección base del programa en memoria principal se coloca en el registro de reubicación. El contenido del registro de reubicación se suma a cada dirección generada por un programa

en ejecución. El usuario puede programar como si su programa se iniciara en la posición cero.

Interrupciones y escrutinio

Una técnica con la que una unidad puede verificar el estado de otra unidad de funcionamiento independiente es el escrutinio (polling); la primera unidad verifica si la segunda se encuentra en cierto estado; si no es así, entonces la primera unidad continúa con la tarea que estaba realizando. (El escrutinio puede llegar a ser un proceso muy costoso).

Por medio de interrupciones, una unidad puede de inmediato obtener la atención de otra para informarle el cambio de estado. La interrupción hace que se almacene el estado de la unidad interrumpida antes de procesar la interrupción. Después se procesa la interrupción y se restablece el estado de la unidad interrumpida.

Los buffers

Un buffer es un área de memoria principal para retener datos durante transferencias de entrada/salida. Mientras se efectúa una transferencia de entrada y salida, su velocidad depende muchos factores relacionados con el hardware de entrada/salida, pero normalmente no tiene que ver con operación del procesador.

En el caso del canal de entrada con buffer único, el cual deposita datos en el buffer, el procesador procesa estos datos, el canal de entrada deposita los siguientes datos, etc. Mientras el canal de entrada está depositando datos, no puede hacerse ningún procesamiento; mientras se procesa un dato no pueden depositarse más datos.

Protección del almacenamiento

La protección del almacenamiento es esencial en los sistemas multiusuario. La protección limita el margen de direcciones a las que puede hacer referencia un programa para evitar que los programas se interfieran entre sí. La protección del almacenamiento para un programa puede implantarse en bloque contiguo de localidades de memoria mediante registros de límites que definan las direcciones superiores o inferiores del bloque de memoria. Mientras que se ejecuta el programa, se verifican todas las direcciones de memoria a las que se hace referencia para ver si se encuentran entre las direcciones del registro de límites. La protección del almacenamiento también puede llevarse a la práctica mediante el uso de claves de protección del almacenamiento asociadas con áreas de la memoria principal; un programa puede hacer referencia a localidades de memoria solamente en aquellas áreas cuyas claves coincidan con la clave del programa.

Cronómetros y relojes

Un cronómetro de intervalos es útil en un sistema multiusuario para evitar que un usuario monopolice el procesador. Después de un intervalo designado, el cronómetro genera una interrupción para llamar la atención del procesador; éste puede asignarse entonces a otro

usuario. Con un reloj horario, el computador puede controlar la hora de un reloj de pared en incrementos de millonésimas de segundos o menos.

Canales de Entrada y Salida

Debido a la necesidad de mayor poder de computo en los primeros sistemas, preferentemente usados en ambientes comerciales de procesamiento, los sistemas tendían a quedar limitados por la entrada/salida (E/S) de datos. Mientras se realizaba la entrada y salida, los procesadores se mantenían ocupados controlándola. En algunos sistemas era posible controlar solo una operación de E/S a la vez. Una mejora importante de esta situación fue el desarrollo de los canales de E/S. Un canal es un computador de propósito especial dedicado al control de E/S independiente del procesador central del sistema. Un canal puede obtener acceso directo al almacenamiento primario para guardar o recuperar información.

En los sistemas actuales controlados por interrupciones, un procesador ejecuta una interrupción *start_io* para iniciar la transferencia de E/S por un canal y el canal genera una interrupción *fin* para informar al procesador que ya terminó la operación de E/S.

Para la transferencia de datos a alta velocidad entre dispositivos y la memoria principal se usa un canal selector que puede poseer un solo sub-canal y solamente puede atender a un dispositivo a la vez.

Los canales que hacen uso de multiplexores tienen muchos sub-canales, pudiéndose intercalar muchos flujos de datos a la vez. Un canal multiplexor de bytes alterna las transmisiones de dispositivos lentos, como terminales, impresoras y líneas de comunicación de baja velocidad. Un canal multiplexor de bloque alterna la transmisión de varios dispositivos de alta velocidad, como impresoras láser y unidades de disco.

Memoria virtual

Los sistemas de almacenamiento virtual permiten a los programas hacer referencia de direcciones que no necesariamente corresponden al conjunto limitado de direcciones reales disponibles en la memoria principal. El hardware traduce dinámicamente las direcciones virtuales generados por un programa en ejecución en direcciones de instrucciones y datos dentro de almacenamiento primario. Los sistemas de direccionamiento virtuales generalmente permiten a un programa hacer referencia a espacios de direcciones virtuales, por lo general mucho mayores que los espacios de direcciones disponibles dentro del almacenamiento primario. Esto permite a los usuarios desarrollar programas independientes de las restricciones en memoria principal y facilita la operación en sistemas multiusuario compartidos.

Los sistemas de almacenamiento virtual emplean la técnica de paginación de bloques de tamaño fijo que se intercambian entre la memoria principal y la memoria secundaria.

Robo de ciclos

Un punto de conflictos entre los canales y el procesador es el acceso a la memoria principal. Como no puede realizarse más que un acceso (a un determinado banco de memoria principal) a la vez, y como es posible que los canales y el procesador deseen obtener acceso simultáneo a la memoria principal, los canales suelen tener prioridad. Eso se denomina robo de ciclos; el canal literalmente roba ciclos de almacenamientos al procesador. El canal utiliza solamente un pequeño porcentaje de los ciclos, pero al concederles prioridad se consigue un mejor aprovechamiento de los dispositivos de E/S.

Este tipo de lógica se ha incorporado en los sistemas operativos actuales; los programas limitados por E/S tienen por lo general prioridad sobre los programados limitados por el procesador en los mecanismos de planificación del sistema operativo.

Multiprocesamiento

Los sistemas multiprocesados poseen varios procesadores que comparten una memoria principal común y un solo sistema operativo. El multiprocesamiento crea la posibilidad de cierto tipo de conflictos que no se presentan en los sistemas de un solo procesador. Es necesario ordenar secuencialmente el acceso a una localidad de memoria compartida, para que dos procesadores no traten de modificarla al mismo tiempo, con la posibilidad de alterar inadecuadamente su contenido. El orden secuencial también es necesario cuando un procesador desea modificar un localidad que otro procesador desea leer.

Acceso directo a memoria

Para obtener mejor rendimiento en un sistema de computación es necesario reducir al mínimo el número de interrupciones que ocurren en la ejecución de un programa. El acceso directo a memoria (Access Memory Direct, DMA) requiere de sólo una interrupción por cada bloque de caracteres transferidos en una operación de E/S.

Cuando se inicia una operación de E/S, los caracteres se transfieren hacia la memoria principal mediante el robo de ciclos.

Cuando un dispositivo está listo para transmitir un carácter de un bloque, “interrumpe” al procesador. Pero con el DMA no es necesario guardar el estado del procesador; éste sufre un retraso, más que una interrupción. Bajo el control de hardware especial, el carácter se transfiere a la memoria principal. Una vez terminada la transferencia, el procesador prosigue su operación.

El DMA es una característica de rendimiento particularmente útil en sistemas que manejan un gran volumen de transferencias de E/S. El hardware encargado del robo de ciclos y de operar los dispositivos de E/S en el modo DMA se conoce como canal de DMA.

Software

Lenguaje de Máquina

El lenguaje de máquina es el lenguaje de programación que el computador puede entender directamente. Cada instrucción en lenguaje de máquina es interpretada por el hardware que realiza las funciones indicadas. Las instrucciones en lenguaje de máquina son, por lo general, bastante primitivas. La disposición de estas instrucciones dentro de los programas en lenguaje de máquina es lo que permite la especificación de algoritmos útiles. Los conjuntos actuales de instrucciones de lenguaje de máquina incluyen a menudo algunas características muy poderosas.

El lenguaje de máquina es dependiente de la máquina, un programa escrito para una máquina o arquitectura en particular no puede ser ejecutado en otra máquina de distinta arquitectura.

El uso de spools

En el modo spool (Simultaneous Peripheral Operations On Line, operaciones periféricas simultáneas en línea), un dispositivo de alta velocidad como un disco duro se interpone entre un programa en ejecución y un dispositivo de baja velocidad relacionado con la entrada y salida del programa. Por ejemplo, en lugar de escribir directamente en una impresora, las salidas se escriben en el disco (archivos). Así los programas pueden finalizar más rápidamente su ejecución, dejando los recursos disponibles para que se inicien otros programas. Cuando la impresora haya terminado de imprimir continuará con otra impresión del contenido del spool de impresión.

Compiladores rápidos y sucios vs. Compiladores optimizados

En el desarrollo de programas, las compilaciones se realizan con frecuencia y los programas casi siempre se ejecutan antes de que aparezca un error. Para dicho propósito son útiles los compiladores rápidos y sucios. Tales compiladores producen rápidamente un programa objeto, pero el código generado puede ser bastante ineficiente, en cuanto a su consumo de memoria y su velocidad de ejecución. Una vez que se ha depurado el programa y se ha preparado para entrar en producción, se usa un compilador optimizador para generar código de máquina de alta eficiencia. Los compiladores optimizadores se ejecutan más lento y producen un código de más alta calidad.

Intérpretes

Los intérpretes no producen programas objetos, sino que ejecutan directamente los programas fuentes. Los intérpretes evitan el tiempo del ensamblaje o la compilación. El intérprete se ejecuta lentamente en comparación con el código compilado, ya que cada instrucción debe ser traducida por el intérprete para ser ejecutada.

Firmware

El forma general el firmware es un micro-código de software que por lo general viene instalado en dispositivos de procesamiento de uso específico y con capacidades limitadas por el hardware. Otra forma de ver el firmware es como un módulo de software programado a bajo nivel que controla el funcionamiento del equipo y da acceso a todas sus capacidades. De manera simplificada se definir como cualquier software almacenado en cualquier tipo de memoria ROM o EPROM.

Como la mayoría de las piezas de hardware no sólo son simples circuitos eléctricos y chips. Estas llevan incluida en su interior pequeños programas en microcódigo que les permiten comunicarse con las demás partes de un sistema de computación.

En algunos casos el firmware puede ser actualizado transfiriendo desde un computador al dispositivo archivos especiales (imágenes binarias los microprogramas) que los fabricantes ponen a disposición en sus sitios en Internet. Los dispositivos que por lo general contienen firmware son: módems, servidores de impresión, cámaras IP, etc.

Existen casos en que el firmware esta compuesto por un kernel de sistema operativo completo como es el caso de los Servidores de Impresión, los módems ADSL, los Routers, etc. En estos casos se podría afirmar en jerga española que el software esta embebido (o empotrado) a un dispositivo, ya que el software viene de fabrica o es actualizado, en ambos casos no existe más interacción del usuario con el dispositivo de la que se ha establecido en el uso específico de dicho dispositivo.

Componentes de un Sistema Operativo

Procesos

Un concepto importante en todos los Sistemas Operativos es el proceso. Un proceso es básicamente un programa en ejecución y consta de un programa ejecutable, los datos, la pila de programa, su contador de programa, apuntador de pila y otros registros. Como también todo la información necesaria para ejecutar el programa.

La manera más de entender lo que es un proceso es ver como estos trabajan en los sistemas de tiempo compartido. En forma periódica, el sistema operativo decide suspender la ejecución de un proceso e iniciar la ejecución de otro.

Cuando un proceso se suspende temporalmente, debe reiniciarse después exactamente en el mismo estado que se encontraba cuando se detuvo. Esto significa que toda la información relativa al proceso debe guardarse en forma explícita en algún lugar durante la suspensión. En muchos sistemas operativos, toda la información referente a cada proceso, diferente del contenido de su espacio de direcciones, se almacena en una tabla del sistema operativo llamada “tabla de procesos”, la cual es un arreglo o lista enlazada de estructuras, un registro del arreglo para cada proceso en existencia.

Por lo tanto, un proceso suspendido consta de su espacio de direcciones, generalmente denominado “imagen del núcleo” y su registro de la tabla de procesos, que contiene sus registros, y otro tipo de información.

Los sistemas de Archivos

Un archivo es un conjunto de datos al cual se le asigna un nombre, éste por lo general reside en algún medio de almacenamiento secundario (disco duro, disco flexible, cdrom, etc). Para hacer posible el uso de archivo en cualquier sistema es necesario establecer cierta estructura, o sea, establecer un *sistema de archivos* que permita el acceso y uso de los archivos en forma organizada y jerarquizada. Los sistemas de archivos son un componente muy importante en cualquier sistema operativo y por lo general contienen:

Métodos de acceso: que se ocupan de la forma (estructura) en que se obtiene acceso a los datos almacenados.

Administración de archivos: que se ocupan de ofrecer los mecanismos para almacenar, compartir y asegurar los archivos, y hacer referencia a ellos.

Administración de almacenamiento secundario: que se ocupa de asignar espacio a los archivos en dispositivos de almacenamiento secundario.

Mecanismos de integridad de archivos: que se ocupan de garantizar que no se corrompa la información en un archivo. Cuando se asegura la integridad de los archivos, cualquier información que deba estar en un archivo estará ahí; la información que no deba estar en un archivo se mantendrá fuera.

Para obtener una buena organización de archivos, los sistemas de archivos además de poseer archivos, poseen directorios. Estos directorios se estructuran en forma de árbol y son contenedores de archivos.

Interprete de comandos (Shell)

El interprete de comandos es la interfaz primaria entre un usuario y un sistema operativo en un computador, ya sea en un sistema operativo monousuario o multiusuario (en caso que este posea terminales físicos y/o virtuales).

Cuando un usuario entra a algún sistema, se inicia como primer proceso del usuario un interprete de comandos.

En el caso del MS-DOS por ejemplo el sistema se inicia en forma predeterminada en un interprete de comando único (command.com). Los Unix en cambio poseen varios interpretes de comandos (sh, csh, tcsh, bash, etc) y pueden ser usados por muchos usuarios a la vez.

La Interfaz Gráfica de usuario

Los sistemas operativos de última generación en la actualidad poseen un Interfaz Gráfica de Usuario (Graphic User Interface, GUI) que facilita la interacción usuario-computador mediante el uso de ventanas gráficas contenedoras de objetos tales como botones, menús, campos de texto, etc; y cuyos accesos se establecen mediante ratones (puntero en pantalla). En algunos casos las GUI dejan oculta la interfaz de comandos, mediante las llamadas hechas por las aplicaciones de ventanas al intérprete de comandos.

Llamadas al Sistema

Llamadas al sistema para manejo de procesos

Las llamadas al sistema son mecanismos que provee el sistema operativo para hacer uso de los sus recursos. Para manejo de proceso se puede usar por ejemplo la llamada fork, que es la única forma de crear un nuevo proceso. Este crea un duplicado exacto del proceso original, incluyendo todos los descriptores de archivos. Después de fork, el proceso original y la copia (el padre y el hijo) toman caminos separados. Todas las variables tienen valores idénticos en el instante de fork, pero como es copiada toda la imagen del núcleo del proceso padre para crear el proceso hijo, los cambios subsiguientes en uno de ellos no afectan al otro. La llamada fork produce un valor, que es cero en el proceso hijo o PID (Process Identification) en el proceso padre. Mediante el uso del PID producido, los dos procesos observan cual es el proceso padre y cual es proceso hijo.

```
while (TRUE) {                                     /* Repetir Siempre */
    Read_command(comando, parámetros);             /* Leer entrada desde el terminal */

    if (fork()!=0) {                                 /* Ramificar un proceso hijo */
        wait(&status);                               /* Código padre */
    } else {
        execve(comando, parámetros, 0);             /* Código hijo */
    }
}
```

Ejemplo de intérprete de comandos (Shell) simplificado

Llamadas al sistema para señalización

Existen casos en que los procesos necesitan manejar interrupciones. Cuando se envía una señal a un proceso que no ha anunciado su disposición para aceptar esa señal, el proceso se elimina sin más discusión. Para evitar esta eliminación, los procesos pueden emplear una

llamada al sistema (SIGNAL) para anunciar que está preparado para aceptar algún tipo de señal y para proporcionar la dirección de manejo de la señalización. Después de una llamada a SIGNAL, si se genera una señal de tipo relevante (interrupción), el estado del proceso es ingresado a su pila, seguida de una llamada al manejador de señales. Cuando se termina el procedimiento de manejo de señales, simplemente se regresa al estado original. El sistema de tiempo de ejecución en el espacio de direcciones del usuario, restituye después el estado del proceso desde la y continúa la ejecución desde el punto en que fue interrumpido.

Llamadas al Sistema para manejo de archivos

Muchas llamadas al sistema están relacionadas con los archivos y con el sistema de archivos. Para crear un nuevo archivo, se usa la llamada CREAT. Sus parámetros se componen por el nombre del archivo los atributos y/o modos de protección.

Un ejemplo de Unix sería :

```
fd = create("archivo",0751);
```

CREAT no solo crea un Nuevo archivo, sino que también lo abre para su escritura, importar el modo del archivo. El descriptor de archivo producido, fd, puede utilizarse para escribir el archivo. Para leer o escribir un archivo existente, primero debe abrirse el archivo utilizando *open*. Esta llamada especifica el nombre del archivo que se abrirá, ya sea un nombre de trayectoria absoluta o relativa al directorio de trabajo y un código de 0,1 ó 2 que significa leer, escribir o ambos. El descriptor del archivo producido puede utilizarse entonces para lectura o escritura. Después el archivo puede cerrarse por medio de la llamada *close*, que libera el descriptor de archivos para volver usarlo con una nueva llamada *creat* u *open*.

Las llamadas al sistema para manejo de archivos son indudablemente *read* y *write*. Aunque muchos programas leen y escriben en forma secuencial, para algunos programas de aplicación es necesario poder acceder a alguna parte de un archivo al azar. En asociación con cada archivo está un apuntador que indica la posición corriente del archivo. Cuando se lee o escribe en forma secuencial, éste normalmente apunta al siguiente byte que se leerá o escribirá. La llamada *lseek* cambia el valor del apuntador de la posición, de modo que las llamadas subsiguientes en *read* o *write* pueden iniciar cualquier parte del archivo o más allá del final de éste.

Llamada *lseek* posee tres parámetros; el primero es el descriptor de archivo, el segundo es una posición en el archivo y el tercero indica si la posición en el archivo es relativa al inicio de éste, la posición corriente o bien el final del archivo.

Llamadas al sistema para manejo de directorios

Una de las llamadas para manejo de directorio es el *link*. El objetivo de *link* consiste en permitir que el mismo archivo figure con dos o más nombre, a menudo en directorios distintos. Un uso común consiste en permitir que varios miembros del mismo equipo de programación compartan un archivo común, donde cada uno de ellos hace que el archivo se muestre en su propio directorio, posiblemente con nombre distintos.

En Unix por ejemplo el comando para hacer uso de la llamada *link* es:

```
ln -s /home/midir/dir1 dir2
ln -s /home/midir/archivo1 archivo2
```

Procesos

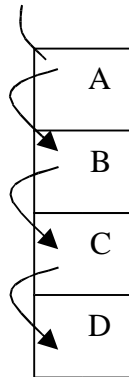
Todos los computadores modernos realizan varias tareas al mismo tiempo. Mientras se ejecuta programa de un usuario, un computador puede estar también el mismo tiempo leyendo un disco o imprimiendo en una impresora. En un sistema de multiprogramación, la Unidad Central de Procesos (CPU) también cambia de un programa a otro, ejecutando cada uno en porciones de tiempo muy reducidas (milisegundos). En rigor, en cualquier instante de tiempo la CPU está ejecutando sólo un programa, pero en un segundo puede ejecutar varios programas, con lo cual se percibe virtualmente un procesamiento en paralelo.

Modelo del proceso

En este modelo, todo el software ejecutable en el computador, que con frecuencia incluye al sistema operativo, se organiza en procesos secuenciales o simplemente procesos (para simplificar la operación). Un proceso es básicamente un programa en ejecución, en donde intervienen los valores corrientes del contador de programa, registros y variables. De manera conceptual, cada proceso tiene su propia CPU virtual. Sin embargo en la realidad la CPU esta constantemente cambiando de un proceso a otro. Para comprender el sistema es más sencillo pensar en un conjunto de procesos que se ejecutan en seudo paralelismo que intenta llevar el control de la forma en que la CPU cambia de un programa a otro, a éste cambio se le llama multiprogramación.

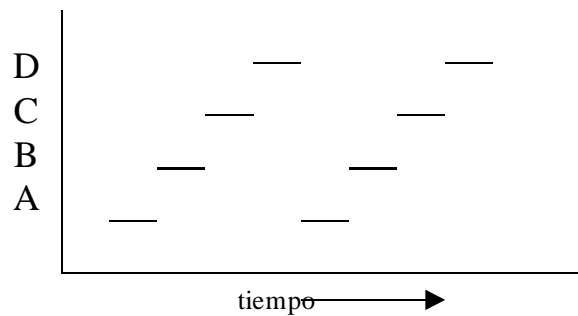
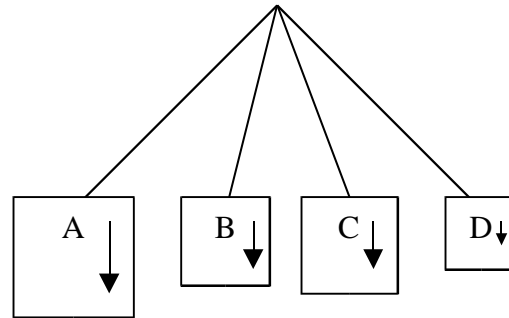
Ejemplos

a)



Ejemplo con un
contador
de programa

b) Ejemplo con cuatro contadores de programa



c) Ejemplo de porciones de tiempo por programa

- a) Computador Multiprogramado con cuatro programas en memoria,
- b) Cuatro procesos cada uno de los cuales posee flujo de control y cada uno se ejecuta independientemente de los otros.
- c) Los cuatro procesos contemplados en un tiempo largo, todos los procesos han procesado, pero en un instante t sólo se ejecuta un proceso en realidad.

Como la CPU conmuta entre un proceso y otro la velocidad a la cual un proceso realiza sus cálculos no serán uniformes y quizás no serán clonables si los mismos procesos se vuelven

a ejecutar. Por lo tanto, los procesos no deben programarse con suposiciones con respecto al tiempo.

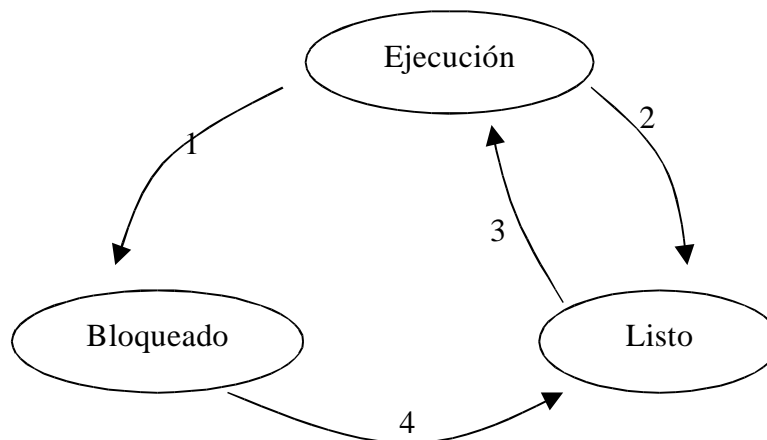
Jerarquía de procesos

Los sistemas operativos que soportan el concepto de proceso deben contar con alguna manera de crear todos los procesos que sean necesarios. En sistemas diseñados para ejecutar sólo una aplicación, es posible hacer que todos los procesos que se necesiten estén presentes cuando lleguen al sistema. Sin embargo, en la mayoría de los sistemas se necesita tener alguna manera de crear y destruir proceso conforme se necesite durante la operación.

Estados de los Procesos

Los procesos pueden estar en básicamente en tres estados; ejecución, listo y bloqueado. Un proceso en ejecución hace uso de la CPU durante algún tiempo, el estado bloqueado es cuando el proceso es incapaz de ejecutarse hasta que suceda algún evento externo, el estado listo es cuando el proceso se detiene temporalmente para permitir que otro proceso se ejecute por la CPU.

Un proceso se bloquea porque lógicamente no puede continuar, debido a que espera una entrada que aún no está disponible. También es posible que un proceso que está conceptualmente listo para ser ejecutado, se detenga ya que el sistema operativo ha decidido distribuir la CPU a otro proceso por un momento.



En la figura se muestran cuatro transiciones. La transición 1 ocurre cuando un proceso no puede continuar. Las transiciones 2 y 3 son ocasionadas por el planificador de procesos, que es parte del sistema operativo. La transición 2 ocurre cuando el planificador de

procesos decide que el proceso en ejecución ha estado tiempo suficiente ejecutándose y el turno de permitir que otro proceso tome el tiempo de CPU. La transición 3 ocurre cuando todos los procesos han utilizado su parte del tiempo y es hora que el primer proceso vuelva a usar la CPU. La Transición 4 ocurre cuando aparece el evento externo que estaba esperando un proceso. Si ningún proceso se ejecuta es ese instante, la transición 3 se activará de inmediato y el proceso iniciará su ejecución. De lo contrario tal vez tenga que esperar en estado *listo* un poco más hasta que la CPU este disponible.

Implementación de Procesos

Para implementar el modelo de procesos, el Sistema Operativo posee una tabla o un arreglo de estructuras, llamada tabla de procesos, que contiene un registro por cada proceso. Este registro contiene información acerca del estado de un proceso, su contador de programa, su apuntador de pila, distribución de memoria, las condiciones de sus archivos abiertos, su información de rendición de cuentas y planificación, y todo lo referente al proceso que deba guardarse cuando el proceso cambia de estado de ejecución a listo, de manera que pueda reiniciarse después como si nunca se hubiese detenido.

Comunicación de Procesos

Los procesos con frecuencia necesitan la comunicación con otros procesos. Cuando un proceso de usuario desea leer el contenido de un archivo, éste debe señalar al proceso el archivo que desea. Después el proceso del archivo tiene que indicar al proceso de disco que lea el bloque que se pide. En una tubería (hilo-threads) del shell, la salida del primer proceso debe pasarse al segundo proceso. En resumen, se necesita establecer comunicación entre procesos, preferentemente de manera bien estructurada sin hacer uso de interrupciones.

Aunque cada proceso es una entidad independiente, con su contador de programa y estados internos. A menudo necesita interactuar con otros procesos. Un proceso puede generar una salida que otro proceso utilizará como entrada (entrada y salida estándar).

Un comando de ejemplo en Unix estaría dado por la instrucción:

```
ps aux | grep more  
grep usuario /etc/passwd | more
```

Referencias

H.M.Deitel, Sistemas Operativos, Addison Wesley, Iberoamericana 1993.

Andrew S. Tanenbaum, Sistemas Operativos Modernos (Pearson Education)