

5. El hardware, el software y los Sistemas Operativos

Al control y operación de los sistemas de ordenador contribuyen una serie de elementos hard, soft y firmware.

El **hardware** está formado por los dispositivos, como son los procesadores, memorias, dispositivos de entrada/salida y las conexiones; el **software** engloba los datos y programas en lenguaje máquina que son interpretados por el hardware; sus elementos más comunes son los compiladores, ensambladores, cargadores, editores, montadores, aplicaciones del usuario y del sistema operativo, y **firmware** se refiere a los programas en microcódigo ejecutados desde memorias de control (ROMs y PROMs).

5.1 Elementos hardware

Al hablar de la evolución de los sistemas operativos, se nombran diversos mecanismos y dispositivos hardware que fueron desarrollándose de forma paralela. Entre los elementos dirigidos a realizar o facilitar las operaciones de entrada y salida se pueden destacar los siguientes:

Periféricos

Los dispositivos periféricos permiten el almacenamiento de cantidades masivas de información fuera de la memoria principal. Destacan las cintas magnéticas, con diferentes densidades de grabación, que son inherentemente secuenciales; y los discos, que son dispositivos de acceso directo, y en consecuencia permiten el acceso a datos individuales y concretos sin necesidad de recorrer todos los anteriores en un determinado orden.

Procesadores satélite

Algunos periféricos (terminales inteligentes) se han diseñado para operaciones 'on-line', conectados al procesador central, y otros para operaciones 'off-line', que se ejecutan bajo el control de unidades no conectadas con el ordenador central. Estas unidades, conocidas como procesadores satélites o procesadores 'stand-alone', posibilitan el control de periféricos sin cargar al procesador y en ellos se realizan operaciones auxiliares; como por ejemplo: transferencias entre tarjeta y cinta, cinta e impresora, etc...

Almacenamiento intermedio o 'buffering'

Un 'buffer' es un área de memoria principal para el mantenimiento de datos durante las transferencias de entrada/salida de datos. La velocidad de transferencia de datos de entrada/salida depende de muchos factores relacionados con el hardware de entrada/salida pero no con la operación del procesador. Por ejemplo, en una lectura, los datos son situados en el buffer por un canal de entrada/salida. Cuando se ha completado la transferencia el procesador puede acceder a los datos. Hay diferentes sistemas; por ejemplo, de buffer simple, de buffer doble, etc. En estos últimos se usa la técnica 'flip-flop' o basculación, que permite el solapamiento de operaciones de entrada/salida con las del procesador.

Interrupciones

Son útiles en ambientes en los que se pueden dar muchas operaciones asíncronas, pero ocasionalmente se necesita una sincronización. Así, una unidad puede obtener la inmediata atención de otra registrando un cambio de estado. Por ejemplo, generalmente es necesario un mecanismo de interrupción para poder solapar las tareas de E/S con las del procesador central.

Canales de entrada/salida

Un canal es un sistema de propósito especial ideado para la ejecución de operaciones de E/S independientemente del procesador central, pudiendo acceder a memoria principal, para almacenar o recuperar información. El procesador ejecuta una instrucción de arranque de la operación de E/S (start-io) y la transfiere al canal; el canal cuando termina de procesar dicha operación provoca una interrupción de entrada/salida para informar del evento sistema operativo.

Básicamente, hay dos tipos de canales: los selectores y los multiplexores. Los selectores se usan para transferir datos como ráfagas a gran velocidad entre los dispositivos y memoria principal; poseen un solo subcanal y sólo pueden atender a un dispositivo de cada vez. Los multiplexores tienen muchos subcanales, pudiendo intercalar muchas cadenas de datos una vez. Hay distintos tipos, los multiplexores de bytes, para dispositivos lentos, y los multiplexores de bloques que multiplexan ráfagas, para dispositivos de alta velocidad.

Controladores de dispositivos

En los sistemas en los que priman las operaciones de E/S y la obtención de alto rendimiento específico (por ejemplo, 'mainframes' muy costosos con gran cantidad de

usuarios y un número importante de dispositivos periféricos), se suelen utilizar canales de E/S, que liberan la UCP de realizar transferencias de datos con medios externos.

Sin embargo, hoy en día, teniendo en cuenta el grado de sofisticación que está alcanzando el hardware, son pocos los ordenadores cuya arquitectura está basada en el uso de canales. Las funciones del canal se encuentran integradas en el controlador de dispositivo. La UCP solamente lanza la operación de E/S, arrancando una serie de rutinas ('drivers' de dispositivo) que se encargan de suministrar al controlador la información necesaria para llevar a cabo dicha operación de E/S. Cuando ésta termina, el controlador interrumpe a la UCP para indicar que a acabado su tarea e informar de posibles errores.

Acceso directo a memoria (DMA)

Una forma de optimizar los sistemas informáticos es minimizar el número de interrupciones que tienen lugar durante la ejecución de las operaciones de E/S. Este acceso elimina la necesidad de interrumpir al procesador por cada byte que se transmite durante una operación de entrada/salida. Si los caracteres se envían a 1200 baudios, el terminal puede aceptar y emitir un carácter aproximadamente cada 8 milisegundos. Una rutina de interrupción bien diseñada para sacar caracteres de un buffer a pantalla, requiere 20 microsegundos por carácter lo que deja 7980 microsegundos de los 8000, para el cálculo de la UCP.

Sin embargo, un dispositivo de alta velocidad, como la cinta, el disco o las redes de comunicaciones pueden transmitir caracteres cada menos microsegundos; por ejemplo a la UCP le pueden llegar interrupciones cada 4 microsegundos pero necesita para responder a cada una de ellas 20 microsegundos. Para resolver este problema se utiliza el acceso directo a memoria o DMA. Cuando la transmisión de un bloque entero se completa, se genera una interrupción simple. Los caracteres individuales se transmiten a y desde memoria principal en base a un robo de ciclo; el canal tiene prioridad mientras el procesador espera. DMA es una forma de ejecución particularmente útil en sistemas que soportan un amplio número de transferencias de datos, y el hardware responsable de la E/S en modo DMA, se denomina canal DMA.

Robo de ciclo (cycle stealing)

Un punto de conflicto entre los canales y el procesador es el acceso a memoria principal. Dado que en un momento sólo puede estar en curso un acceso a un banco de memoria principal, y como es posible que los canales y el procesador intenten acceder simultáneamente, lo que se hace normalmente es dar prioridad a los canales; a esto se

denomina robo de ciclo. Como los canales sólo usan un pequeño porcentaje de ciclos, dándoles prioridad se optimiza el uso de los dispositivos. Esta lógica se usa en los sistemas operativos actuales; generalmente los mecanismos de planificación de los sistemas operativos dan más prioridad a los programas con operaciones de E/S que a los programas que requieren mucho procesador.

Instrucciones privilegiadas

Generalmente los sistemas de ordenador tienen diferentes modos de funcionamiento o estados de ejecución. Variando el estado de una máquina se facilita la seguridad de los sistemas. Normalmente, cuando la máquina está en un estado en particular el programa en ejecución sólo puede ejecutar un conjunto de instrucciones. Por ejemplo, en estado problema no se puede realizar la ejecución directa de instrucciones de entrada/salida previniéndose así los accesos incontrolados que podrían llegar a destruir el sistema operativo; sin embargo, en estado supervisor (que es cuando el sistema operativo tiene el control), se tiene acceso a todo el conjunto de instrucciones.

El cambio de modo de usuario a modo supervisor se realiza automáticamente por hard al darse cualquiera de las circunstancias siguientes:

- Un proceso de usuario hace una llamada al sistema operativo pidiendo alguna/s función que requiera la ejecución de instrucciones reservadas. Una llamada de este tipo recibe el nombre de extracódigo o llamada al supervisor.
- Tiene lugar una interrupción.
- En un proceso de usuario se da una condición de error o excepción. Esta condición puede tratarse como una interrupción interna y ser gestionada en primera instancia por una rutina de interrupción.
- Se intenta ejecutar alguna instrucción reservada estando en modo usuario. Este puede considerarse como un tipo particular de error y tratarse como el caso anterior.

La vuelta al modo usuario desde el modo supervisor se lleva a cabo mediante una instrucción también privilegiada.

Para sistemas de alta seguridad se hace necesaria la existencia de más estados con los que afinar la protección (cada uno tendrá sólo los privilegios necesarios para ejecutar su tarea). En base a esto los ordenadores han evolucionado así como el número de instrucciones privilegiadas; es decir, las que no son accesibles en estado problema. Estas instrucciones especiales llevan a cabo funciones tales como: autorizar e inhibir las

interrupciones, conmutar un procesador entre los diferentes procesos, modificar los registros de protección de memoria, controlar las operaciones de E/S, parar el procesador central, cargar el temporizador, etc.

Otro conjunto, es el formado por los elementos relacionados con la gestión de memoria, entre los que hay que destacar los siguientes:

Jerarquía de almacenamiento

Hoy en día los sistemas presentan varios niveles de almacenamiento: memoria principal, memoria secundaria y memoria caché o de almacenamiento temporal y los registros del procesador. Las instrucciones y los datos referenciados por un programa en ejecución deben estar situados en memoria principal. La memoria auxiliar soportada en cintas, discos, etc., está diseñada para mantener información que eventualmente puede pasar a memoria principal. La memoria 'caché' es muy rápida y está diseñada para incrementar la velocidad de ejecución de los programas de forma transparente. Los registros del procesador constituyen el almacenamiento más rápido seguida por la memoria caché, a nivel menor la memoria principal, y por último la secundaria. Estos niveles de almacenamiento forman la jerarquía de memoria en la que decrece el coste y velocidad, incrementándose la capacidad.

Memoria virtual

Los sistemas con memoria virtual permiten a los programas referenciar direcciones que no necesitan corresponderse con direcciones físicas disponibles en memoria real, y su traducción a reales se realiza dinámicamente en tiempo de ejecución mediante hardware. En consecuencia, permiten la creación de programas independientes de las restricciones de memoria principal y facilitan la operación de los sistemas multiusuario. Los sistemas de memoria virtual usan técnicas de paginación o división en bloques de tamaño fijo para el trasvase entre memoria principal y memoria auxiliar, técnicas de segmentación que identifica unidades lógicas en los programas y datos para facilitar el control de acceso y la compartición, o técnicas combinadas.

Por ejemplo, el soporte hard de memoria virtual englobaría:

- Un mecanismo traductor dinámico de direcciones
- Un dispositivo para el almacenamiento externo de páginas o segmentos
- Un apuntador a la tabla con la información de correspondencia
- Un conjunto de registros de asociativos

Intercalación o 'interleaving' de memoria

Se usa para acelerar el acceso a memoria principal. Habitualmente mientras una posición de un banco de memoria está siendo accedida, no puede estar en curso ninguna otra referencia a ese banco. Con la intercalación de memoria, situando alternativamente conjuntos de posiciones en diferentes bancos de memoria se permite el acceso simultáneo a posiciones sucesivas de memoria principal.

Protección de memoria

Es esencial en todo sistema de ordenador y en especial en los multiusuario. Este mecanismo limita el rango de direcciones que un programa puede referenciar; es decir, limita su espacio de direcciones. La protección de memoria se implementa de muy diferentes formas: mediante registros de límites, que definen las direcciones más baja y más alta para un bloque de memoria (para sistemas de asignación contigua de memoria), o también mediante el uso de claves de protección ligadas a áreas de memoria principal (en sistemas paginados o en segmentados), etc.

Direccionamiento por base más desplazamiento

Al hacerse necesario memorias de más tamaño, la arquitectura de los ordenadores ha tenido que cambiar. Por ejemplo, un sistema diseñado para soportar 16 Mbytes necesitaría para su direccionamiento, direcciones de 24 bits. Su incorporación en cada instrucción en máquinas con direccionamiento simple sería muy costoso, y en máquinas con instrucciones de múltiple dirección intolerable. Por tanto, para poder abarcar rangos amplios de direcciones, los sistemas direccionan por base más desplazamiento; es decir, a todas las direcciones se les suma el contenido del registro base. Este esquema tiene una ventaja adicional, facilitar la independencia de los programas de la posición de carga.

Registro de reubicación

Permite la reubicación dinámica de los programas. El contenido de este registro, la dirección base del programa en memoria, se añade a cada dirección desarrollada en la ejecución del programa. Esto permite que un programa no tenga que ejecutarse siempre en una posición determinada.

$$\text{Reb. Reubicación: } \quad \text{Dir. base o de carga} + \text{Dir. lógicas} = \text{Dir. Relativa}$$

Por último, hay que destacar dos elementos que son esenciales en la implementación de políticas de planificación y en el control de los recursos consumidos por los diferentes usuarios.

Temporizadores y el reloj de tiempo real

El valor del temporizador se va decrementando según va pasando el tiempo de reloj. De forma, que cuando llega a cero, es decir se ha consumido el tiempo asignado, se produce una interrupción, denominada interrupción por reloj, por la cual, el procesador, por ejemplo, es cedido a otro proceso.

5.2 Elementos software

El software está compuesto por los programas de instrucciones y datos que definen los algoritmos para resolver los problemas. Entre todos los elementos software que componen un sistema operativo destacan los siguientes:

Lenguaje máquina

Es el lenguaje que comprende el computador directamente. Cada instrucción es interpretada y ejecutada por el hard. Se dice que es un lenguaje dependiente de la máquina porque dado que las instrucciones de este lenguaje nombran registros específicos del ordenador y procesan datos en la forma física en que están, un programa escrito en este lenguaje no puede ejecutarse en un ordenador diferente.

Ensambladores y lenguajes de alto nivel

La programación en lenguaje máquina es costosa en tiempo y difícilmente libre de errores. El ensamblador se desarrolló para acelerar la programación y reducir los errores de codificación. Estos lenguajes utilizan abreviaturas significativas y palabras para sustituir a las cadenas de números del lenguaje máquina. En consecuencia, los programas en ensamblador deben ser ensamblados o traducidos a lenguaje máquina para que el ordenador pueda procesarlos. Estos lenguajes también son dependientes de la máquina ya que sus instrucciones se corresponden una a una con las del lenguaje máquina.

Para mejorar el proceso de codificación de programas en ensamblador se desarrollaron las macros. Una macro al ensamblarse se expande; es decir es sustituida por una serie de instrucciones en ensamblador, que realizan la misma tarea. La tendencia a crear instrucciones cada vez más poderosas condujo al desarrollo de los lenguajes de alto nivel que permite escribir programas independientemente de la máquina.

Con estos lenguajes se aumenta considerablemente la velocidad de programación, los programas se hacen transportables y el usuario puede desarrollar aplicaciones sin tener que estar familiarizado con la estructura interna del computador.

En general hay dos tipos de lenguajes de alto nivel: Los orientados al procedimiento y los orientados al problema. Los primeros son de propósito general que se pueden utilizar en la resolución de una gran variedad de problemas; son por ejemplo: COBOL, FORTRAN, PASCAL, BASIC, PL/I, etc. y se pueden clasificar a su vez, según su orientación, en tres grupos: A la estructura de control (COBOL), a las estructuras de datos (PASCAL) y a la estructura de objetos (SMALLTALK, EIFIEL, C++). Los orientados al problema, son los que se usan en problemas particulares como por ejemplo son: SPSS para estadística, GPSS y ASCL para simulación, etc.

Compiladores 'quick-and-dirty' y optimizados, e Intérpretes

Los programas escritos tanto en ensamblador como en lenguajes de alto nivel, tienen que traducirse a lenguaje máquina para que el ordenador pueda entenderlos. Los programas encargados de ello son los compiladores, que pueden ser 'quick-and-dirty' u optimizados. Los primeros son útiles durante la depuración de los programas dada la rapidez con que generan los objetos. Una vez depurados, para ponerlos en producción se pueden utilizar los optimizados que aunque son más lentos producen a un código máquina muy eficiente. Estos compiladores son tan efectivos que la calidad del código producido excede o por lo menos es igual a la del producido al ensamblar un programa en ensamblador. Hoy en día los sistemas operativos grandes se desarrollan en lenguaje de alto nivel y se pasan a código máquina mediante compiladores optimizados.

Los intérpretes realizan la misma función, pero la traducción y ejecución de los programas se realiza en un mismo tiempo no generándose el objeto. Estos traductores están extendidos sobre todo en los ambientes de ordenadores personales, pero son muy lentos comparados con los compiladores.

Cargador y Montador de enlaces (linking-loader & linkage-editor)

Al principio, los programadores escribían sus programas en lenguaje máquina, con todas las instrucciones necesarias para resolver un problema en concreto. Hoy en día, los programas de usuario contienen sólo una pequeña parte de las instrucciones y datos necesarios y para realizar ciertas operaciones comunes como las de entrada/salida, utilizan rutinas del sistema que por lo general están situadas en bibliotecas de subrutinas. En consecuencia, el programa en lenguaje máquina producido por el traductor debe ser combinado con otros para conseguir una unidad ejecutable útil.

Este proceso de combinación o montaje de programas lo realizan los cargadores y montadores de enlaces antes de que se ejecute el programa. En tiempo de carga, un cargador de enlaces combina todos los programas necesarios y los carga directamente en memoria principal. El montador de enlaces también realiza esta combinación de programas, pero crea una imagen de carga que deja en memoria secundaria para futuras referencias, lo que es particularmente útil en ambientes de producción.

Cargadores absolutos y reubicantes

Los programas deben situarse en memoria principal para poder ser ejecutados. La correspondencia entre instrucciones y campos de datos y posiciones de memoria principal que se conoce como ligazón o 'binding' y es una tarea, muy importante que recae unas veces sobre los usuarios programadores, otras sobre el traductor, otras sobre un programa del sistema llamado cargador y otras sobre el sistema operativo. La tendencia es retardar esta operación lo más posible; así, mientras en los programas en lenguaje máquina se realiza en el momento de la codificación en los sistemas de memoria virtual se hace dinámicamente al ejecutarse el programa. Con esta demora se flexibiliza tanto la labor del usuario, que programa independientemente del ordenador, como la del sistema, pero a costa de la sofisticación de traductores, cargadores y hardware de los sistemas operativos.

Un cargador es un programa que sitúa las instrucciones y datos de un programa en posiciones de memoria principal, y puede ser absoluto o reubicante. El absoluto, sitúa estos campos en las posiciones concretas que indica el programa ya en lenguaje máquina; la binding se habrá hecho en la traducción. El reubicante podrá cargar un programa en diferentes posiciones de memoria principal dependiendo de diferentes factores como la disponibilidad de memoria principal en el momento de carga (load time). En este caso, la binding se realizará en tiempo de carga.

Spooling (Simultaneous Peripheral Operation On Line)

Técnica por la que un dispositivo de alta velocidad se interpone entre un programa en ejecución y un dispositivo de baja velocidad que el programa utiliza para sus operaciones de entrada/salida. El programa puede terminar su ejecución antes y otros programas pueden iniciarse antes.

Sistema de control de entrada/salida (IOCS)

Los programas de canal detallados, necesarios para controlar la entrada/salida, y las diferentes rutinas para coordinar la operación de canales y procesadores son complejos. El desarrollo de un programa supervisor para gestionarlo elimina esta carga

de los programas de aplicación. A este programa se le denomina sistema de control de entrada/salida o I.O.C.S.

