CAPÍTULO 1

**Introducción a los sistemas**

**informáticos**

Un sistema operativo (SO) explota los recursos de hardware. El sistema operativo también gestiona la memoria secundaria y los dispositivos de entrada/salida (E/S) en nombre de los usuarios.

Este capítulo ofrece una visión de conjunto del hardware de los sistemas informáticos.

**ELEMENTOS BÁSICOS**

En un alto nivel, un sistema informático consta de procesador, memoria y componentes de E/S. Estas componentes están interconectados de alguna forma para llevar a cabo la función principal del computador. Así pues, se tienen cuatro elementos principales:

•*Procesador:* Controla la operación del computador y lleva a cabo las funciones de pro- cesamiento de datos. *(CPU, Central Processing Unit).*

•*Memoria Principal:* Almacena los datos y los programas. se le conoce como *memoria real o memoria primaria.*

•*Interconexión de sistemas:* Ciertos mecanismos y estructuras que permiten la comuni- cación entre procesadores, memoria principal y los módulos de E/S.

**2 Introducción a los sistemas informáticos**

Un módulo de memoria consta de un conjunto de ubicaciones definidas por direcciones enumeradas secuencialmente. Un módulo de E/S transfiere datos desde los dispositivos externos hacia la memoria y el procesador y viceversa.

**REGISTROS DEL PROCESADOR**

Dentro del procesador, hay un conjunto de registros que es más rápido y pequeño que la memoria principal. Los registros del procesador sirven para dos funciones:

• *Registros visibles de usuario:* Un programador puede minimizar las referencias a memoria principal mediante un uso óptimo de estos registros.

• *Registros de control y de estado:* Son utilizados por el procesador para el control de las operaciones del sistema operativo para controlar la ejecución de los programas.

No hay una separación clara de los registros en estas dos categorías. Por ejemplo, en al-gunas máquinas el contador de programa es visible para los usuarios, pero en otras muchas no lo es.

**Registros visibles de usuario**

Un registro visible de usuario es aquél que puede ser referenciado por medio del lenguaje de máquina que ejecuta el procesador. Las clases de registro que, normalmente, están disponibles, son los registros de datos, los registros de dirección y los registros de códigos de condición.

Los **registros de datos** pueden ser asignados por el programador a diversas funciones. Pueden ser empleados por cualquier instrucción de máquina que lleve a cabo operaciones sobre los datos.

Los **registros de dirección** contienen direcciones en la memoria principal de datos e ins-trucciones o una parte de la dirección que se utiliza en el cálculo de la dirección completa. Pueden ser de propósito general o pueden estar dedicados a un modo especí-fico de direccionamiento. se incluyen:

• *Registro índice:* El direccionamiento indexado es un modo común de direccionamiento que implica sumar un índice a un valor base para obtener la dirección efectiva.

• *Punterode segmento:* La memoria se divide bloques de palabras de tamaño variable. En este modo, se utiliza un registro que alberga una dirección base de un segmento.

• *Puntero de pila:* Si hay un direccionamiento de pila visible para los usuarios, la pila es-tará, por lo general, en la memoria principal, existiendo un registro dedicado a señalar

**4 Introducción a los sistemas informáticos**

Una última categoría de registros que son, al menos, parcialmente visibles para los usuarios, son aquellos que contienen **códigos de condición** (*indicadores o flags).* Los códigos de condición son bits activados por el hardware del procesador como resultado de determinadas operaciones

**Registros de control y de estado**

Varios registros se emplean para controlar las operaciones del procesador. la mayor parte de estos registros no son visibles para los usuarios.

Por supuesto, máquinas diferentes tendrán organizaciones diferentes de registros y podrán usar terminologías distintas. los registros **MAR, MBR, IOAR y IOBR** mencionados anteriormente, los siguientes registros son esenciales en la ejecución de instrucciones;

• *Contador de programa (PC, Program Counter):* Contiene la dirección de la instrucción a ser leída.

• *Registro de instrucción (IR, Instruction Regíster):* Contiene la última instrucción leída.

Entre los campos e indicadores más comunes se incluyen los siguientes:

• *Signo:* Contiene el bit del signo de la última operación aritmética efectuada.

• *Cero:* Se activa cuando el resultado de una operación aritmética es cero.

• *Acarreo:* Se activa cuando, como resultado de una suma o una resta, se produce un acarreo más allá del bit más significativo.

*• Igualdad:* Se activa si una comparación lógica da como resultado la igualdad.

*• Desbordamiento:* Empleado para seflalar un desbordamiento aritmético.

*• Habilitar/inhahilitar interrupciónes:* Empleado para habilitar o inhabilitar interrupcio-nes.

*• Supervisor:* Indica si el procesador está ejecutando en modo supervisor o en modo usuarlo.

Una serie de factores inciden en el diseño de la organización de los registros de control y estado. Un punto clave es el soporte del sistema operativo.

**EJECUCIÓN DE INSTRUCCIONES**

La tarea básica que realiza un computador es la ejecución de los programas.

El procesamiento requerido para una instrucción simple se llama *ciclo de instrucción.* El ciclo

Los dos pasos se llaman *ciclo de lectura (fetch) y ciclo de ejecución.* La ejecución del programa se detiene sólo si se apaga la máquina, ocurre algún tipo de error irrecuperable o se encuentra una instrucción en el programa que detiene el computador.

**Lectura y ejecución de instrucciones**

el procesador lee una instrucción de la memoria.

La instrucción leída se carga en un registro del procesador conocido como *registro de instrucción* (IR). El procesador interpreta la instrucción y realiza la acción requerida. estas acciones pueden clasificarse en las siguientes cuatro categorías:

• *Procesador-memoria:* Se transfieren datos del procesador a la memoria o viceversa.

• *Procesador-EIS:* Se transfieren datos desde o hacia un dispositivo periférico, realizán- dose la transferencia entre el procesador y un módulo de E/S.

• *Tratamiento de datos:* El procesador realiza alguna operación aritmética o lógica sobre los datos,

• *Control:* La instrucción pide que se altere la secuencia de ejecución.

El procesador contiene un único registro de datos, llamado *acumulador* (AC).

**Funciones de E/S**

Los módulos de E/S pueden intercambiar datos di-rectamente con el procesador. Al igual que el procesador puede iniciar una lectura o escritura en la memoria, el procesador también puede leer datos de un módulo de E/S o escribir datos en el módulo

Durante la transferencia, el módulo de E/S emite órdenes de lectura o escritura en la memoria, librando de responsabilidades al procesador en el intercambio. Esta operación se conoce como *acceso directo a memoria* (DMA, *Direct Memory Access)*

**INTERRUPCIONES**

Casi todos los computadores tienen un mecanismo mediante el cual otros módulos (E/S, memoria) pueden interrumpir la ejecución normal del procesador.

Las interrupciones aparecen, principalmente, como una vía para mejorar la eficiencia del procesamiento.

**TABLA 1.1 Clases de Interrupciones**

De programa: Generadas por alguna condición que se produce como resultado de la ejecución de una instrucción, como el desbordamiento aritmético, la división por cero, el intento de ejecutar una instrucción ilegal de la máquina o una referencia a una zona de memoria fuera del espacio permitido al usuario.

De reloj: Generadas por un reloj interno del procesador. Esto permite al sistema operativo llevar a cabo ciertas funciones con determinada regularidad.

De E/S: Generadas por un controlador de E/S, para indicar que una operación ha terminado normalmente o para indicar diversas condiciones de error.

Por fallo del hardware: Generadas por fallos tales como un corte de energía o un error de paridad de la memoria.

El programa de usuario lleva a cabo una serie de llamadas a ESCRIBIR. Las llamadas a ESCRIBIR son, en realidad, llamadas a un programa de E/S, que es una utilidad del sistema que llevará a cabo la operación concreta de E/S. El programa de E/S consta de tres secciones:

• Una secuencia de instrucciones, etiquetada con un 4 en la figura, de preparación para la operación concreta de E/S.

• La orden concreta de E/S. Sin el uso de interrupciones, una vez que se emita esta orden, el programa debe esperar a que el dispositivo de E/S lleve a cabo la función pedida.

• Una secuencia de instrucciones, para completar la operación. Esto puede incluir la activación de un código de condición que indique el éxito o el fracaso de la operación.

**Las interrupciones y el ciclo de instrucción**

Con las interrupciones, el procesador se puede dedicar a la ejecución de otras instrucciones mientras una operación de E/S está en proceso.

Cuando el dispositivo de E/S esté disponible, es decir, cuando esté preparado para aceptar más datos desde el procesador, el módulo de E/S de dicho dispositivo enviará una señal de *solicitud de interrupción* al procesador.

Desde el punto de vista del programa de usuario, una interrupción es solamente eso: una interrupción de la secuencia normal de ejecución. Cuando el tratamiento de la interrupción se termina, la ejecución continúa.

Para dar cabida a las interrupciones, se añade un *ciclo de interrupción* al ciclo de instrucción. En el ciclo de interrupción, comprueba si ha ocurrido alguna interrupción, lo que se indicará con la presencia de una señal de interrupción. Si no hay interrupciones pendientes, el procesador sigue con el ciclo de lectura y trae la próxima instrucción del programa en curso. Si hay una interrupción pendiente, el procesador suspende la ejecución del programa en curso y ejecuta una rutina de *tratamiento de la interrupción.*

La rutina de tratamiento de la interrupción forma parte generalmente del sistema operativo.

**Tratamiento de interrupciones**

El acontecimiento de una interrupción desencadena una serie de sucesos, tanto en el hard- ware del procesador como en el software. se produce en el hardware la siguiente secuencia de sucesos:

1. El dispositivo emite una señal de interrupción al procesador.

2. El procesador finaliza la ejecución de la instrucción en curso antes de responder a la interrupción.

3. El procesador pregunta por la interrupción, comprueba que hay una y envía una señal de reconocimiento al dispositivo que generó la interrupción. Este reconocimiento le permite al dispositivo suprimir la señal de interrupción.

4. El procesador necesita ahora prepararse para transferir el control a la rutina de interrupción.

5. El procesador carga ahora el contador de programa con la ubicación de entrada del programa de tratamiento de la interrupción.

Una vez que se ha cargado el contador de programa, el procesador procede con el próximo ciclo de instrucción, que comienza trayendo la próxima instrucción. Debido a que esta instrucción se determina por el contenido del contador de programa, el resultado es que el control se le transfiere al programa que trata la interrupción. La ejecución de este programa se traduce en las operaciones siguientes:

6. En este punto, el contador de programa y la PSW relativa al programa interrumpido han sido salvadas en la pila del sistema

7. La rutina de tratamiento de la interrupción puede ahora proceder a procesar la interrupción.

8. Cuando se completa el tratamiento de la interrupción, se recuperan de la pila los valores de los registros que se salvaron y se restauran los registros.

9. El acto final es restaurar los valores de la PSW y del contador de programa a partir de la pila.

Es importante salvar toda la información sobre el estado del programa interrumpido para su reanudación posterior.

**Interrupciones múltiples**

Hay dos enfoques para tratar las interrupciones múltiples. El primero es inhabilitar las in-terrupciones mientras se esté procesando una. Una *interrupción inhabilitada* quiere decir que el procesador ignorará la señal de interrupción.

La limitación de este enfoque es que no tiene en cuenta las prioridades relativas o necesi-dades críticas en tiempo.

Un segundo enfoque es definir prioridades para las interrupciones y permitir que una interrupción de una prioridad más alta pueda interrumpir a la rutina de tratamiento de una interrupción de prioridad más baja

**Multiprogramación**

Aún con el uso de interrupciones, puede que un procesador no esté aprovechado de una manera muy eficiente. Una solución a este problema es permitir que varios programas de usuario estén activos a un mismo tiempo.

la secuencia en que es- tos se ejecutan dependerá de su prioridad relativa y de si están esperando una E/S. Esta situación de varios programas que se ejecutan por turnos se conoce como *multiprogramación* y será discutida más adelante en el capítulo 2.

**JERARQUÍA DE MEMORIA**

Las limitaciones de diseño de la memoria de un computador se pueden resumir en tres preguntas: ¿qué cantidad?, ¿qué velocidad? y ¿qué coste?

La cuestión de qué cantidad está relativamente abierta. Según sea la capacidad, probable-mente se construirán aplicaciones que la utilicen. La cuestión de la velocidad es, en cierto sentido, fácil de responder. la memoria debe ser capaz de ir al ritmo del procesador.

Como se puede suponer, estas tres características compiten entre sí: coste, capacidad y tiempo de acceso. A lo largo de este abanico de tecnologías, se cumplen las siguientes relaciones:

• A menor tiempo de acceso, mayor coste por bit

• A mayor capacidad, menor coste por bit

• A mayor capacidad, mayor tiempo de acceso

A medida que se desciende por la jerarquía se tienen las siguientes condiciones:

1. Disminución del coste por bit

2. Aumento de la capacidad

3. Aumento del tiempo de acceso

4. Disminución de la frecuencia de acceso a la memoria por parte del procesador

Así pues, las memorias más pequeñas, caras y rápidas son reemplazadas por memorias de más capacidad, más baratas y lentas.

es posible organizar los datos en la jerarquía de modo que el porcentaje de accesos a los niveles inmediatamente inferiores sea considerablemente menor que el del nivel superior.

Normalmente, un procesador tendrá unas pocas decenas de registros, la memoria principal, también conocida como memoria real, que es la memoria interna principal del computador. La memoria principal se suele ampliar con una pequeña memoria caché de alta velocidad.

Las tres formas de memoria descritas son, generalmente, volátiles y emplean tecnologías de semiconductores.

Se pueden incluir otras formas de memoria en la jerarquía. los grandes computadores centrales *(mainframes)* incluyen una forma de memoria interna conocida como *memoria expandida*. esta tecnología no encaja en la jerarquía sino que constituye una rama paralela: Los datos se pueden mover entre la memoria principal y la memoria expandida, pero no entre la memoria expandida y la memoria externa. Otras formas de memoria secundaria son los discos ópticos y la memoria de burbuja. Por software, niveles adicionales a la jerarquía. Una parte de la memoria principal puede ser utilizada como un buffer para guardar temporalmente los datos transferidos con el disco. más conocida como *caché de disco*, mejora el rendimiento de dos formas:

• Las escrituras a disco se agrupan.

• Algunos datos destinados a la salida pueden ser referenciados por un programa antes del próximo volcado a disco.

**MEMORIA CACHE**

es invisible para el sistema operativo

**Motivación**

La frecuencia con que el procesador puede ejecutar las instrucciones está claramente limitada por el tiempo de ciclo de memoria.

**Principios de la cache**

La memoria caché intenta obtener una velocidad cercana a la de las memorias más rápidas y, al mismo tiempo, aportar una memoria grande al precio de las memorias de semiconductores, que son menos costosas. La caché contiene una copia de parte de la memoria principal.

**Diseño de la cache**

elementos clave. Estos aspectos se encuadran en las siguientes categorías:

•Tamaño de cache.

•Tamaño del bloque.

•Función de correspondencia (*mapping*)

•Algoritmo de reemplazo.

•Política de escritura.

Ya se ha hablado sobre el **tamaño de cache**. que caches razonablemente pequeñas pueden tener un impacto significativo sobre el rendimiento. el **tamaño del bloque,** que es la unidad de intercambio de datos entre la cache y la memoria principal. A medida que el tamaño del bloque aumenta, la tasa de aciertos aumentará al comienzo, debido al principio de cercanía.

Cuando se trae a cache un nuevo bloque de datos, la **función de correspondencia** *(map-ping)* determina la posición de la cache que ocupará el bloque. Dos limitaciones influyen en el diseño de la función de traducción. En primer lugar, cuando un bloque se trae a la cache, puede que otro tenga que ser reemplazado.

En segundo lugar, cuanto más flexible sea la función de traducción, más compleja será la circuitería necesaria para determinar si un bloque dado está en la cache.

El **algoritmo de reemplazo** escoge, bajo las restricciones de la función de traducción, el bloque que hay que reemplazar

Si se modifica el contenido de un bloque de la cache, entonces hace falta escribirlo de nuevo a la memoria principal, antes de reemplazarlo.

**TÉCNICAS DE COMUNICACIÓN DE E/S**

Para las operaciones de E/S son posibles las tres técnicas siguientes:

• E/S programada

• E/S dirigida por interrupciones

• Acceso Directo a Memoria (DMA: Direct Memory Access)

**E/S programada**

Con E/S programada, el módulo de E/S llevará a cabo la acción requerida y luego activará los bits apropiados en el registro de estado de E/S. En particular, no interrumpe al procesador

Con esta técnica, el procesador es el responsable de extraer los datos de la memoria principal cuando va a hacer una salida o poner los datos en la memoria principal cuando se hace una entrada. en el conjunto de instrucciones se incluyen instrucciones de E/S de las siguientes categorías:

• *Control:* Empleadas para activar un dispositivo externo y decirle qué debe hacer.

• *Comprobación:* Empleadas para comprobar varias condiciones de estado asociadas con un módulo de E/S y sus periféricos.

• *Lectura, escritura:* Empleadas para transferir los datos entre los registros del procesador y los dispositivos externos.

**E/S dirigida por interrupciones**

El problema de la E/S programada es que el procesador tiene que esperar un largo rato a que el módulo de E/S en cuestión esté listo para recibir o transmitir más datos

Una alternativa es que el procesador envíe una orden de E/S al módulo y se dedique a ha- cer alguna otra tarea útil. El módulo de E/S interrumpirá entonces al procesador para reque- rir sus servicios cuando esté listo para intercambiar los datos.

Cada línea tiene una prioridad diferente. Otra solución sería habilitar una única línea de interrupción, pero utilizando líneas adicionales para indicar la dirección del dispositivo.

**Acceso directo a memoria**

La E/S dirigida por interrupciones, aunque es más eficiente que la simple E/S programada, todavía requiere de la intervención activa del procesador para transferir los datos entre la memoria y un módulo de E/. Así pues, ambas formas de E/S adolecen de dos desventajas inherentes:

1. La velocidad de transferencia de E/S está limitada por la velocidad con la que el procesador puede comprobar y dar servicio a un dispositivo.

2. El procesador participa en la gestión de la transferencia de E/S

Cuando se tienen que mover grandes volúmenes de datos, se necesita una técnica más efi-ciente: el acceso directo a memoria (DMA, *Direct Memory Access).* La función de DMA se puede llevar a cabo por medio de un módulo separado sobre el bus del sistema o puede estar incorporada dentro de un módulo de E/S. Cuando el procesador desea leer o escribir un bloque de datos, emite una orden hacia el módulo de DMA, enviándole la información siguiente:

• Si lo que se solicita es una lectura o una escritura

• La dirección del dispositivo de E/S involucrado

• La dirección inicial de memoria desde la que se va a leer o a la que se va a escribir

• El número de palabras a leer o escribir

El procesador continúa entonces con otro trabajo. Habrá delegado la operación de E/S en el módulo de DMA.

El módulo de DMA debe tomar el control del bus para transferir los datos con la memoria.

2 parte del capitulo tres(yo tengo la primera parte)

• Un error en una sola rutina, como la de tratamiento de interrupciones, puede dañar los bloques de control de procesos.

• Un cambio de diseño en la estructura o en la semántica del bloque de control de procesos podría afectar a varios módulos del sistema operativo.

Estos problemas se pueden abordar exigiendo a todas las rutinas del sistema operativo que pasen a través de una rutina de manejo, cuya única tarea serla la de proteger los bloques de control de proceso y que se constituiría en el único árbitro para leer y escribir en estos bloques.

**CONTROL DE PROCESOS**

**Modos de ejecución**

La mayoría de los procesadores dan soporte para dos modos de ejecución por lo menos.

El modo menos privilegiado a menudo se conoce como *modo de usuario,* ya que los programas de usuario ejecutan normalmente en ese modo. Al modo más privilegiado normalmente se le conoce como *modo del sistema, modo de control* o*, modo del núcleo.* Este último término se refiere al núcleo del sistema operativo.

La razón por La que se usan dos modos debe quedar clara. Es necesario proteger al sistema operativo y a las tablas importantes del mismo, tales como los bloques de control de procesos, de las injerencias de los programas de usuario.

**TABLA 3.10 Funciones Básicas del Núcleo de un Sistema Operativo**

**Gestión de Procesos**

• Creación y terminación de los procesos

• Planificación y expedición de los procesos

• Cambio de procesos

• Sincronización de procesos y soporte para la comunicación entre procesos

• Gestión de los bloques de control de procesos

**Gestión de memoria**

• Asignación de espacios de direcciones a los procesos

• Intercambio

• Gestión de páginas y segmentos

**Gestión de E/S**

• Gestión de buffers

• Asignación de canales de E/S y dispositivos a los procesos

**Funciones de Soporte**

• Tratamiento de interrupciones

• Contabilidad

• Supervisión

**Creación de procesos**

Una vez que el sistema operativo decide, crear un nuevo pro-ceso, éste puede proceder como sigue:

1. Asignar un único identificador al nuevo proceso.

2. Asignar espacio para el proceso.

3. Debe inicializarse el bloque de control del proceso.

4. Se deben establecer los enlaces apropiados.

*5.* Puede haber otras estructuras de datos que crear o ampliar.

**Cambio de proceso**

puede producirse en cualquier momento en que el sistema operativo haya tornado el control a partir del proceso que está actualmente ejecutándose.

En primer lugar, se van a tener en cuenta las interrupciones del sistema. La primera es originada por algún tipo de suceso que es externo e independiente del proceso que está ejecutándose. La segunda tiene que ver con una condición de error o de excepción generada dentro del proceso que está ejecutándose. En una **interrupción ordinaria, el control** se transfiere primero a un gestor de interrupciones, quien lleva a cabo algunas tareas básicas y, después, se salta a una rutina del sistema operativo que se ocupa del tipo de interrupción que se ha producido. Algunos son los siguientes:

*• Interrupción de reloj:* El sistema operativo determina si el proceso que está en ejecución ha estado ejecutando durante la fracción máxima de tiempo permitida.

*• Interrupción de E/S:* El sistema operativo determina exactamente que se ha producido una acción de E/S.

**TABLA 3.11 Mecanismos para la interrupción de la Ejecución de un Proceso [KRAK88]**

Mecanismo Causa Uso

Interrupción Externa a la ejecución de Reacción a un suceso

la instrucción en curso asincrónico externo

Cepo Asociada con a ejecución de Tratamiento de un error o de

la instrucción en curso una condición excepcional

Llamada del supervisor Solicitud explicita Llamada a una función

del sistema operativo

*• Fallo de memoria:* El procesador encuentra una referencia a una dirección de memoria virtual de una palabra que no está en memoria principal.

**Cambio de contexto**

en el ciclo de interrupción, el procesador comprueba si se ha producido alguna interrupción, Si no hay pendiente ninguna interrupción, el procesador continúa Si hay alguna interrupción, el procesador hace lo siguiente:

1. Salva el contexto del programa que está ejecutándose.

2. Asigna al contador de programa el valor de la dirección de comienzo de un programa de *tratamiento de la interrupción.*

El procesador continúa entonces con el ciclo de lectura de instrucción y trae la primera instrucción del programa de tratamiento de interrupciones, que atenderá a la interrupción.

***Cambio de estado de los procesos***

Puede producirse un cambio de contexto sin cambiar el estado del proceso que está actualmente en estado de Ejecución. salvar el contexto y restaurarlo posteriormente involucra un pequeño coste extra. Los pasos involucrados en un cambio completo de proceso son los siguientes:

1. Salvar el contexto del procesador, incluyendo el contador de programa y otros registros.

2. Actualizar el bloque de control del proceso que estaba en estado de Ejecución.

3. Mover el bloque de control del proceso a la cola apropiada

4. Seleccionar otro proceso para ejecución

5. Actualizar el bloque de control del proceso seleccionado.

6. Actualizar las estructuras de datos de gestión de memoria.

7. Restaurar el contexto del procesador a aquel que existía en el momento en el que el proceso seleccionado dejó por última vez el estado de Ejecución, cargando los valores previos del contador de programa y de otros registros.

Así pues, el cambio de proceso, que supone un cambio de estado, requiere un esfuerzo considerablemente mayor que un cambio de contexto.

**Ejecución del sistema operativo**

• El sistema operativo funciona de la misma forma que un software corriente

• El sistema operativo abandona frecuentemente el control y debe depender de que el procesador le permita recuperarlo.

***Núcleo fuera de todo proceso***

cuando el proceso en ejecución es interrumpido o hace una llamada de supervisor, se salva el contexto del procesador para este proceso y se pasa el control al núcleo. El sistema operativo puede llevar a cabo cualquier función deseada y luego restaurar el contexto del proceso interrumpido para reanudarlo.

el punto clave es que se considera que el concepto de proceso se aplica solo a los programas del usuario.

***Ejecución dentro de los procesos de usuario***

El enfoque es que el sistema operativo es principalmente una colección de rutinas que el usuario llama para llevar a cabo varias funciones y que son ejecutadas dentro del entorno del proceso de usuario

***Sistema operativo basado en procesos***

Una última alternativa es la de implementar el sistema operativo como una colección de procesos del sistema. Al igual que en las otras opciones, el software que forma parte del núcleo ejecutará en modo de núcleo. En este caso, sin embargo, las funciones más importantes del núcleo se organizan en procesos separados.

Este enfoque tiene varias ventajas. Impone unas normas de diseño de programas que promueven el uso de un sistema operativo modular con unas interfaces mínimas y claras entre los módulos. Además, algunas funciones no criticas del sistema operativo se pueden implementas como procesos separados.

**Micronúcleos**

Un micronúcleo es un pequeño núcleo de sistema operativo que proporciona las bases para ampliaciones modulares.

El enfoque de los micronúcleos fue popularizado por el sistema operativo Mach y sus im-plementaciones en la línea de computadores de Next. En teoría, el enfoque del núcleo se supone que brinda una gran flexibilidad y modularidad. Otra aplicación muy conocida del enfoque de micronúcleos es Windows NT, que proclama no solo la modularidad, sino la portabilidad, como los beneficios clave.

La filosofía en que se basa el micronúcleos es que solo las funciones absolutamente esen- ciales del sistema operativo deben permanecer en el núcleo. Las aplicaciones y los servicios menos esenciales se construyen sobre el micronúcleo.

Las componentes del sistema operativo externas al micronúcleo interactúan una con otra sobre una base común, normalmente a través de mensajes distribuidos a través del micronúcleo. De este modo, el micronúcleo funciona como un distribuidor de mensajes: Valida los mensajes, los pasa entre las componentes y otorga el acceso al hardware. Esta estructura es ideal para entornos de proceso distribuido, ya que el micronúcleo puede pasar mensajes tanto en local como en remoto, sin necesidad de cambios en otras componentes del sistema operativo.

**PROCESOS E HILOS**

En la discusión llevada a cabo, se ha presentado el concepto de proceso incluyendo las dos características siguientes:

*• Unidad de propiedad de los recursos:* A cada proceso se le asigna un espacio de direcciones virtuales para albergar a la imagen del proceso

*• Unidad de expedición:* Un proceso es un camino de ejecución a través de uno o más programas.

En la mayoría de los sistemas operativos, estas dos características son, de hecho, la esencia de un proceso. Para distinguir estas dos características, la unidad de expedición se conoce corno **hilo** *(thread)* o proceso ligero *(lightweight process),* mientras que a la unidad de propiedad de los recursos se le suele llamar **proceso** o **tarea.**

**Varios hilos en un solo proceso**

La utilización que más salta a la vista del concepto de hilo es la de una disposición en la que pueden existir varios hilos dentro de un mismo proceso.

En Mach, una tarea se define como la unidad de protección o unidad de asignación de recursos. A las tareas se les asocian los siguientes elementos:

• Un espacio de direcciones virtuales, que contiene la imagen de la tarea.

• Acceso protegido a los procesadores, otros procesos, archivos y recursos de E/S.

En una tarea puede haber uno o más hilos, cada uno con lo siguiente:

• El estado de ejecución del hilo

• El contexto del procesador, que se salva cuando no está ejecutando; una forma de contemplar al hilo es con un contador de programa independiente operando dentro de una tarea

• Una pila de ejecución

• Almacenamiento estático para las variables locales

• Acceso a la memoria y a los recursos de la tarea, que se comparten con todos los otros hilos de la tarea.

Los beneficios clave de los hilos se derivan de las implicaciones del rendimiento. Por tanto, si hay una aplicación o una función que pueda implementarse como un conjunto de unidades de ejecución relacionadas, es más eficiente hacerlo con una colección de hilos que con una colección de tareas separadas.

**[ETW88]** da cuatro ejemplos de uso de los hilos en un sistema de multitarea:

*• Trabajo interactivo* y *de fondo:* Esto se produce en el sentido de la interacción directa con el usuario, no en el de sesiones interactivas y de fondo.

*• Proceso asíncrono:* Los elementos asíncronos del programa se pueden implementar como hilos.

*• Aceleración de la ejecución:* Un proceso con hilos múltiples puede computar un lote de datos mientras lee el lote siguiente de un dispositivo.

*• Organización de los programas:* Los programas que suponen una variedad de actividades o varios orígenes y destinos de entrada y salida pueden hacerse más fáciles de diseñar e implementar mediante hilos.

**Otras estructuras**

Los conceptos de unidad de asignación de recursos y de unidad de expedición se han englobado tradicionalmente dentro del concepto único de proceso, es decir, como una relación de uno a uno entre hilos y procesos, también se han investigado las otras dos combinaciones, las llamadas relaciones de muchos a muchos y las relaciones de uno a muchos.

***Relación de muchos a muchos***

La idea de tener una relación muchos a muchos entre los kilos y los procesos se ha explorado en el sistema operativo experimental. En TRIX, se tienen los conceptos de dominio y de hilo. Un dominio es una entidad estática que consta de un espacio de direcciones y de unos “puertos” a través de los cuales se envían y reciben los mensajes. Un hilo es un camino sencillo de ejecución, con una pila de ejecución, el estado del procesador y la información de planificación.

El uso de un mismo hilo en varios dominios parece motivado, más que nada, por el deseo de brindar al programador una herramienta de estructuración. Hay varias formas de implementar esta aplicación:

1. El programa entero se puede implementar como un único proceso.
2. El programa principal y el subprograma de E/S pueden implementarse como dos procesos separados.
3. Tratar al programa principal y al subprograma de E/S como una actividad única que se debe implementar como un único hilo.

***Relación de uno a muchos***

En el campo de los sistemas operativos distribuidos, ha habido un gran interés en el concepto de hilo, principalmente como una entidad que se puede mover entre los espacios de direcciones.9 Un ejemplo notable de esta investigación es el sistema operativo Clouds.

Un hilo en Clouds es una unidad de actividad desde la perspectiva del usuario.

El enfoque de Clouds ofrece una forma efectiva de aislar al usuario y al programador de los detalles del entorno distribuido.

**EJEMPLOS DE DESCRIPCION Y CONTROL DE PROCESOS**

**Sistema UNIX, versión V**

UNIX utiliza un servicio de procesos simple pero potente, que es muy visible para el usuario.

***Estados de un proceso***

Un total de nueve estados de proceso son los reconocidos por el sistema operativo UNIX. Las diferencias pueden resumirse rápidamente a continuación:

• **UNIX** emplea dos estados de Ejecución, que indican si el proceso está ejecutando en modo de usuario o en modo núcleo.

• Se hace una distinción entre los estados: Listo para Ejecutar y en Memoria, frente al estado de Expulsado. Estos dos estados son, básicamente, el mismo, como se indica por la línea de puntos que los une.

Hay dos procesos que son únicos en UNIX. El proceso 0 que se crea cuando el sistema arranca. Además, el proceso 0 genera el proceso 1, conocido como proceso *init;* todos los demás procesos del sistema tienen al proceso 1 como antepasado.

***Descripción de procesos***

Un proceso en UNIX es un conjunto más bien complejo de estructuras de datos que proporcionan al sistema operativo toda la información necesaria para administrarlos y expedirlos.

Cuando un proceso no está ejecutándose, la información del estado del procesador se al-macena en la zona de contexto de los registros.

**WINDOWS NT**

El diseño de los procesos de Windows NT está dirigido por la necesidad de dar soporte a varios entornos de sistemas operativos. Los procesos aportados por los distintos sistemas operativos son diferentes en varios aspectos, incluyendo los siguientes:

• Cómo se les denomina a los procesos

• Si hay hilos disponibles dentro de los procesos

• Cómo se representan los procesos

• Cómo se protegen los recursos de los procesos

• Qué mecanismos se emplean para la comunicación y la sincronización entre procesos

• Cómo están relacionados los procesos entre sí

Por consiguiente, la estructura nativa de los procesos y de los servicios que brinda el núcleo de NT es relativamente simple y de propósito general, permitiendo a cada subsistema emular la estructura y la funcionalidad particular de los procesos de un sistema operativo. Las características más importantes de los procesos de NT son las siguientes:

• Los procesos de NT se implementan como objetos.

• Un proceso ejecutable puede tener uno o más hilos.

• Los objetos proceso y los objetos hilo tienen capacidades predefinidas de sincronización.

• El núcleo de NT no conserva ninguna relación entre los procesos que crea, incluyendo las relaciones padre-hijo.

***Objetos proceso y objetos hilo***

La estructura orientada a objetos de NT facilita el desarrollo de un servicio de procesos de propósito general. NT hace uso de dos tipos de objetos relacionados con los procesos: los procesos y los hilos. Un proceso es una entidad correspondiente a un trabajo de usuario o una aplicación, que dispone de sus propios recursos, tales como memoria y archivos abiertos. Un hilo es una unidad de trabajo que se puede expedir para su ejecución secuencial y que es interrumpible, de forma que el procesador pueda pasar de un hilo a otro.

***Hilos múltiples***

NT da soporte para la concurrencia entre procesos, pues los hilos de procesos diferentes pueden ejecutarse concurrentemente. Un proceso con hilos múltiples puede lograr la concurrencia sin la sobrecarga del empleo de varios procesos.

***Soporte para subsistemas***

Los servicios generales de procesos e hilos deben dar soporte a las estructuras de hilos y procesos particulares de varios clientes del SO. Es responsabilidad de cada subsistema el aprovechar las características de los procesos e hilos de NT para emular los servicios de pro-cesos e hilos del sistema operativo correspondiente.

**MVS**

***Tareas de MVS***

El sistema operativo MYS emplea un entorno de procesos muy estructurado. En MYS, la memoria virtual se divide en grandes regiones llamadas *espacios de direcciones. E*l MVS puede manejar y, a menudo, maneja miles de tareas concurrentemente.

Explícitamente, el MVS hace uso de solo tres estados para una tarea: Listo, Activo (en ejecución) y Esperando. Sin embargo, se puede descargar a memoria secundaria un espacio de direcciones entero.

De esta forma, en MVS se pueden hallar, a su manera, dos conceptos aparentemente tan modernos como son el de hilos dentro de un proceso y el de contemplar al sistema operativo como un conjunto de procesos.

***Estructuras de control de tareas***

Enumera las estructuras principales de control que utiliza el MYS para administrar las tareas de usuario y las tareas del sistema. El bloque de control de espacio de direcciones (ASCB*) y* el bloque ampliado de espacio de direcciones (ASXB*)* contienen información sobre los espacios de direcciones. La distinción es que MVS necesita la información del ASCB cuando está ejecutando en el espacio de direcciones reservado del sistema, mientras que el ASXB contiene información adicional necesaria solo cuando MVS está tratando con un espacio de direcciones específico de un usuario.

Una vez que se selecciona un espacio de direcciones, MVS puede trabajar con las estructuras de dicho espacio, conocidas como la zona local de colas del sistema. MVS expide al SRB que tenga prioridad mayor y, si n**O** hay ninguno, elige al TCB de prioridad mayor. Consta de las Siguientes tareas:

*• Tarea de Control de Regiones (RCT, Region Control Task):* Responsable de la administración del espacio de direcciones en nombre de todas las tareas que ejecutan en él.

*• Volcado (dump):* Responsable de volcar a disco el espacio de direcciones si éste termina de una forma anormal.

*• Tarea de Control Iniciada (STC, Started Control Task):* El TCB del programa que da inicio al espacio de direcciones.

*• Iniciador:* Responsable de cargar el flujo de trabajos por lotes.

*• Tarea de usuario:* La aplicación puede estar formada por una o más tareas de usuario.

La ventaja de dividir la información de control en global y local es que, si es necesario, se puede descargar al disco tanta información de un espacio de direcciones como sea posible, reservando así la memoria principal.