



Trabajo Final

Diseño de Sistemas Operativos

Realizado por:

Marcelino Tena Blanco; NIA: 100383266

Curso: 2019-2020

Contenido

Tema 1: Introducción	4
Qué es un sistema operativo.....	4
Cómo es un sistema operativo y qué objetivos tiene	6
Tema 2: Funcionamiento del sistema operativo.....	7
Arranque del sistema	7
Arranque de hardware	7
Arranque de software	7
Tratamiento de eventos.....	8
Eventos software.....	8
Eventos hardware	8
Cómo suceden los eventos y cómo se solucionan	8
Tema 3: Gestión de procesos y gestión de dispositivos.....	10
Gestión de procesos.....	10
Concepto de Proceso	10
Cómo funciona la gestión de procesos	11
Gestión de periféricos	12
Concepto de periférico.....	12
Cómo funciona la gestión de periféricos.....	13
Tema 4: Sistema de archivos.....	15
Concepto de archivo	15
Operaciones con archivos	15
Concepto de directorio	16
El sistema de archivos	17
Estructura	17
Implementación de directorios	18
Métodos de asignación	18
Gestión del espacio libre	19
Tema 5: Gestión de memoria.....	20
Partes de memoria que tiene un proceso.....	20
Soporte de memoria virtual	20
La gestión de memoria.....	21
Mecanismos para la gestión de memoria	21
Políticas de la gestión de memoria	22
Tema 6: Aspectos avanzados	23
La seguridad en los sistemas operativos	23

Los fallos y errores en el sistema operativo	24
Los sistemas distribuidos	24
Los sistemas de tiempo real	24
Resumen general.....	26
Referencias.....	28

Tema 1: Introducción

En el mundo en el que vivimos estamos rodeados de ordenadores. Esto es un hecho. Desde nuestro teléfono móvil o nuestro computador personal hasta grandes centros de información, bases datos o estaciones espaciales. Estos ordenadores nos ayudan a realizar tareas rutinarias haciéndonos la vida más fácil. Para ayudarnos, hace falta que el conjunto hardware y software sea eficiente. Por eso, en este tema, voy a tratar sobre que es aquello a lo que le llamamos software, más concretamente, a aquello que nos deja ejecutar programas (el programa ejecutado es un proceso) y nos ayuda a organizarlos para que exista un control en nuestro ordenador. Hablo del sistema operativo.

Qué es un sistema operativo

Un sistema operativo es un conjunto de programas con la capacidad de poder controlar y gestionar los recursos proporcionados por el hardware. De esta forma, por ejemplo, cuando abres un nuevo proceso, el sistema operativo es quien añade los datos y el código a memoria, administra el tiempo de ejecución, administra el lugar donde están esos datos y es capaz de que ese proceso pueda acceder al distinto hardware que tengas conectado.

Un sistema operativo es capaz de funcionar en distintos dispositivos, a lo que le llamamos portabilidad. Esto significa que se puede adaptar a distintos hardware, ya que existen drivers genéricos que ayudan a reconocer muchos elementos de entrada/salida fácilmente. El sistema operativo también es delicado, porque al más mínimo fallo que ocurra cuando esté ejecutándose, saltará el kernel panic o el bsod (blue screen of death), que son formas de prevenir pérdida de información e intentarán reparar el suceso. Por último, el sistema operativo sirve para todo, desde labores de oficina como escribir documentos o hacer hojas de cálculo hasta desarrollar aplicaciones pasando por poder jugar a videojuegos.

Una vez hablado sobre las características generales de un sistema operativo, voy a exponer las funciones principales a grandes rasgos:

- Gestiona los procesos activos: es capaz de crear, destruir, parar, reanudar y gestionar la comunicación de procesos. Estos procesos pueden ser ejecutados en modo usuario o en modo kernel:
 - El modo usuario se da en aquellas ejecuciones que no pueden acceder al núcleo del sistema operativo, es decir, a las funciones principales que le hace funcionar correctamente.
 - El modo kernel se activa cuando una proceso obtiene privilegios y es capaz de cambiar los parámetros esenciales del sistema, como puede ser la planificación de procesos o el uso de memoria principal fuera del límite del proceso. También es capaz de acceder a todos los registros del CPU.
- Gestiona los recursos del procesador: es capaz de decidir en qué determinado momento un proceso puede utilizar el procesador, cuanto tiempo puede utilizarlo y en qué momento ese proceso es expulsado. El planificador es el programa que es capaz de realizar estas acciones, además de otras varias como bloquear un proceso o suspenderlo. Existen diversos tipos de algoritmos de planificación como:
 - Round Robin, el cual establece un límite de tiempo igual para cada uno de los procesos y cuando acaba ese tiempo se le expulsa de la CPU.
 - FIFO, donde el primer proceso en la cola es el primero que entra en el procesador y hasta que no acabe no entra el siguiente.

- Shortest Job First o el trabajo más corto primero, el cual estima un periodo de tiempo que durará el proceso y ordena la lista para que los procesos más cortos sean los primeros en entrar en el CPU.
- Gestiona la entrada y salida: es capaz mediante procesos intermedios de poder gestionar todo el hardware conectado al ordenador. Estos procesos son denominados drivers y son los responsables de que cuando tocamos una tecla en el teclado, el ordenador reaccione de una forma o que cuando movemos el ratón, el puntero se desplace. Una curiosidad es que gran parte de un sistema operativo se basa en drivers: drivers de almacenamiento, de impresora, de interfaz humana...
- Gestiona la memoria principal: un recurso indispensable para un ordenador es la memoria principal, ya que los sistemas operativos se basan en memoria. Todo lo que hacemos en un ordenador es agregar, procesar y quitar memoria. Y esta memoria está ubicada en la memoria principal, también denominada memoria de acceso aleatorio (RAM) debido a que se accede a ella mediante direcciones de memoria, es decir, no se accede a ella de forma secuencial. Esta memoria es de rápido acceso y de escasa cantidad debido a su construcción para tener una rápida interacción entre procesador y memoria del proceso. Además de ser escasa, otra pega es que es volátil, es decir, en el momento en el que el procesador se quede sin energía, toda la RAM desaparecerá.
- Gestiona el almacenamiento secundario y el sistema de archivos: es capaz de conocer dónde encontrar un fichero en un almacenamiento externo. Esto es debido a que el almacenamiento externo está dividido en sectores de forma física y en bloques en forma lógica. Estos bloques pueden ser de distintos tamaños y es la forma utilizada para que un sistema operativo reconozca un archivo. Un sistema de archivos tiene diferenciadas varias partes:
 - Superbloque: es el bloque cero de la partición y da a conocer al sistema operativo datos importantes como la capacidad de la partición, el número de bloques, etc.
 - Bloques para conocer la localización de archivos: según el sistema archivos, se pueden utilizar distintas formas para localizar archivos, desde el uso de i-mapas e i-nodos como ocurre en sistemas de ficheros de Linux, hasta tablas de acceso de fichero como ocurre en Windows.
 - Los bloques de datos: es el lugar donde se almacenan en sí los datos de los ficheros. Un fichero puede tener el tamaño del bloque, o puede disponer de varios bloques, pero nunca puede particionar un bloque. Es decir, si tú tienes un tamaño de bloque de 2048 Bytes y añades al disco un archivo de 1024 Bytes, entonces desperdicias 1024 Bytes. A este suceso se le denomina fragmentación interna. En el caso de que tengas un archivo mayor que el tamaño de bloque, se suelen usar técnicas de enlazamiento, las cuales mediante una lista FIFO son capaces de direccionar a los siguientes bloques, o, técnicas secuenciales, las cuales no hacen falta listas sino que un fichero esté en varios bloques seguidos. La última técnica puede generar fragmentación externa cuando eliminas un fichero muy pequeño donde a lo mejor no puede entrar otro y queda el bloque inutilizado (ilustración 1).

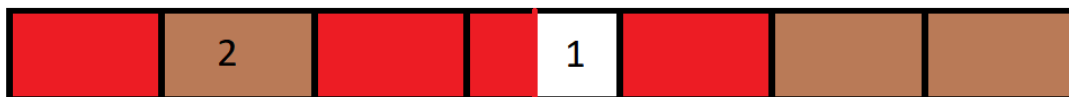


Ilustración 1 Fragmentación interna(1), y fragmentación externa(2). Las bloques en rojo son bloques ocupados, los bloques en verde son libres y la parte blanca es espacio desaprovechado.

Ahora, para poder interaccionar con el sistema operativo, es necesario disponer de una interfaz de usuario. Actualmente el usuario interactúa desde entornos gráficos, pero para usuarios más experimentados se suele usar la interfaz de comandos. Esto es debido a que existen muchos programas a los que añadirles una interfaz gráfica es inútil por varios motivos, como por ejemplo ser muy sencillos o que sea innecesaria para ejecutarse, como un instalador de paquetes.

Por último, el sistema operativo ofrece soluciones de máquina extendida. Una máquina extendida es una simulación de hardware. Es decir, es un sistema operativo corriendo en un hardware simulado. Esto es muy útil cuando quieres probar como funcionaría unos dispositivos específicos en un sistema operativo, por ejemplo: *se está diseñando un taladro capaz de maniobrar solo. Tiene un sistema operativo determinado. El equipo desarrollador quiere comprobar si la emisión y recepción de datos es correcta para conocer si, por ejemplo, cuando el sistema manda al taladrador a encenderse, este se encienda, o que cuando el taladro envíe la profundidad, pasa bien el valor.* Pues para depurar de una forma más rápida sin tener que estar taladrando superficies todo el rato, se puede usar una máquina extendida, donde tú como desarrollador podrías conocer todos los detalles a profundidad.

Cómo es un sistema operativo y qué objetivos tiene

La estructura de un sistema operativo es una parte crucial del mismo, ya que es como se fundamenta. Existen varias formas de estructurar un sistema operativo según lo que se necesite:

- Monolítico: son sistemas donde todos los procesos del kernel están en un mismo programa. En el caso de que quieras cambiar algo del núcleo, tendrás que compilarlo todo de nuevo. Todas las funciones se ejecutan en modo kernel.
- En subsistemas: se sustenta en un sistema monolítico más interfaces o módulos que puedes ir agregando o quitando.
- Por capas: se sustenta en un sistema monolítico pero modificado por capas, de forma que puedes compilar solo una parte en el caso de que quieras cambiar algo.
- Microkernel: es un sistema estructurado que se compone de procesos que ayudan al mismo sistema, los cuales se les denomina servicios.

Los objetivos principales del sistema operativo son:

- Ser eficiente y ofrecer resultados de forma rápida.
- Ofrecer portabilidad en el caso de que quieras tener el sistema en varios ordenadores.
- Ofrecer compatibilidad con programas para que las personas sean capaces de programar y ejecutar sus programas en el sistema.
- Ofrecer estabilidad, para que en trabajos donde no se debe cometer ningún error no sucedan imprevistos por culpa del sistema.
- Claridad, para que todo el mundo pueda ser capaz de poder usarlo.
- Ofrecer la capacidad de expansión, de forma que puedas ir agregando nuevos drivers o programas.

Tema 2: Funcionamiento del sistema operativo

A continuación vamos a hablar sobre qué ocurre cuando enciendes el ordenador y sobre cómo se tratan los eventos.

A grandes rasgos, cuando enciendes un computador, lo que realizas es una comprobación rápida de que el hardware que está conectado responde y cargas los componentes principales del sistema operativo.

Los eventos son señales generadas a partir de un suceso que puede ocurrir por parte del software, mediante llamadas al sistema o interrupciones de concurrencia como los mutex, o por parte del hardware, mediante excepciones o interrupciones de hardware, los cuales envían señales para informar al sistema operativo.

Arranque del sistema

Antes de comenzar a contar los pasos de cómo funciona el arranque de un sistema, debemos tener en cuenta que existen dos modos de ejecución:

- El modo kernel, el cual puede acceder a todo el hardware sin restricciones y puede realizar cualquier acción sin que nadie lo impida. También puede usar los registros de desarrollador del procesador.
- El modo usuario, el cual solo puede acceder a su parte de memoria y los registros de desarrollador del procesador están invisibles para él.

El sistema operativo tiene dos fases: la fase de arranque de software (el programa cargador del sistema operativo) y la fase de arranque de hardware. Lo primero que se realiza son las fases de arranque de hardware.

Arranque de hardware

Cuando se enciende el ordenador o se genera un reinicio, este genera una señal eléctrica que carga en el contador de programa del CPU una dirección específica de la memoria principal, donde la dirección apunta a un área de memoria de solo lectura, es decir, es una parte física que en vez de ser volátil como las demás, es estática. Esta zona hace las siguientes funciones:

1. Comprueba los dispositivos conectados al pc y revisa que son operativos. Aquí es donde se puede interrumpir el inicio y llamar a la BIOS. La BIOS (Basic Input-Output System) es un “mini-sistema” que ofrece los datos básicos del hardware y la posibilidad de hacer cambios drásticos de todo el computador, como cambiar el disco de donde quieres leer el sistema operativo.
2. Almacena en la memoria principal el cargador del sistema operativo, lo que significa que carga todos los procesos que son necesarios para el arranque del sistema operativo y los añade a memoria.
3. Al finalizar la carga del sistema de arranque del SO, entonces cambia el contador de procesos al sistema de arranque.

Arranque de software

Los datos que recoge el sistema de arranque para el sistema operativo se encuentran típicamente en los primeros bloques del disco. Hay que recordar que una cosa es el sistema de arranque que se encuentra en una ROM en la memoria principal y otra cosa es el sistema operativo, el cual se encuentra en el disco secundario.

El cargador del sistema operativo cumple con la misión de traer a memoria principal todos los componentes del sistema operativo. Una vez con ellos:

1. Comprueba que el sistema de archivos y los dispositivos están en buen estado.
2. Se crean las tablas del sistema operativo como las tablas de procesos, de memoria o de escritura.
3. Se cargan aquellos procesos demonio que no se pueden destruir debido a que incapacitaría todo el funcionamiento del SO, por ejemplo, el proceso *init* el cual es el padre de todos los demás procesos.

Una vez acabado el arranque, el sistema operativo toma el control y pasa a la fase de identificación de usuario o si no hiciera falta, pasaría a un estado inactivo esperando órdenes del usuario. Normalmente el estado de inactividad en los sistemas actuales gráficos se representa con el escritorio sin hacer nada.

Tratamiento de eventos

Existen 4 tipos de eventos, los cuales 2 son eventos de software y otros dos son de tipo hardware.

Eventos software

En los eventos software podemos encontrar las llamadas al sistema y las interrupciones de software

- Llamadas al sistema: es un mecanismo de solicitud de uso de un servicio del sistema operativo que proviene de un programa usuario típicamente donde los servicios pueden ser imprimir por pantalla, abrir un fichero o leer de teclado. Son eventos síncronos, es decir, son sencillos de reconocer en qué momento van a ocurrir.
- Interrupción de software: es un evento que no procede de la parte que es prescindible del tratamiento de un evento. Son eventos asíncronos porque no son sencillos de reconocer en qué momento van a ocurrir.

Eventos hardware

Podemos encontrar desde eventos de llamadas desde cualquier dispositivo, que son las interrupciones de hardware, hasta eventos por errores del CPU, llamados excepciones:

- Excepciones: son eventos imprevistos por parte del CPU, el cual avisa de cualquier error relacionado con los datos otorgados, como una división por cero, intentar acceder a un área de memoria del sistema operativo sin permiso o una instrucción de modo privilegiado en un modo no privilegiado. Es un evento asíncrono.
- Interrupciones de hardware: es la forma de comunicación del hardware al sistema operativo. Cuando el hardware quiere otorgar información o que le den información, envía una señal al sistema operativo para que la trate. Entonces, el sistema operativo toma la señal y la agrega a la tabla de interrupciones a la espera de ser atendida. Cuando llega el turno, el sistema trata la interrupción. Es un evento síncrono.

Cómo suceden los eventos y cómo se solucionan

Todos los eventos se tratan de la misma manera solo que algunos, como he dicho anteriormente, son asíncronos, es decir, se tratan de forma que interrumpen todo el sistema de forma espontánea y dan su evento; y otros son síncronos, lo que quiere decir que se conoce el momento en el cual van a suceder. El proceso de evento es el siguiente:

Para las llamadas al sistema:

1. El proceso ejecuta una llamada al sistema.
2. El sistema pasa a ser modo privilegiado y realiza la llamada.
3. El proceso pasa a la lista de bloqueados o de espera hasta que se ejecute una interrupción de hardware. Cuando esta se ejecute, pasan el proceso a la lista de listos a la espera de que el planificador le pase a ejecutar.
4. El proceso obtiene el resultado y sigue con sus instrucciones.

Para los fallos de software o de la CPU (excepciones):

1. Se realiza una interrupción por errores de software o de CPU.
2. El sistema operativo guarda los datos del proceso actual y lo añade a la lista de listos.
3. El sistema operativo pasa a modo privilegiado.
4. Según la gravedad del error, decide una acción.
5. En caso de que el sistema no puede proseguir, se reiniciará. Si puede proseguir, enviará al usuario el problema sucedido para que lo solucione.

Todos estos eventos suceden cuando el sistema operativo ya está cargado. Nunca se puede dar una interrupción en el momento de arranque del sistema operativo.

Tema 3: Gestión de procesos y gestión de dispositivos

En el tema actual, voy a resumir los conceptos importantes de la gestión de procesos y de la gestión de dispositivos. Ambos son dos temas que van unidos debido a que lo común es que un proceso utilice un dispositivo y el dispositivo ofrezca información al proceso.

Gestión de procesos

La gestión de procesos es una parte del sistema operativo muy importante ya que controla los programas ejecutados en el sistema. Estos procesos pueden estar en tres estados esenciales: en listo, ejecutado y bloqueado. Un proceso acabado se elimina de la tabla de procesos. A continuación hablo más sobre el tema

Concepto de Proceso

Un proceso es un conjunto de instrucciones y datos que están en memoria principal y que son procesados por la CPU. La ejecución de un programa da lugar a un proceso. Los procesos son gestionados por el sistema operativo.

Un proceso tiene:

- Recursos asociados, los cuales son una zona de memoria otorgada por el sistema operativo solo para ese proceso, donde almacenará las instrucciones, los datos y la pila. También tiene acceso a los archivos abiertos y a las señales que le llegan.
- Acceso a estructuras de control, las cuales son las tablas de memoria donde obtienen cualquier información sobre las memorias, las tablas de ficheros de donde se obtiene la información de un fichero, las tablas de dispositivos donde se puede conocer toda la información sobre los dispositivos conectados y la tabla de procesos, lugar donde se gestionan toda la información de procesos.
- Estados, los cuales pueden ser listo, en ejecución o bloqueado. Estos estados se gestionan a través del planificador, un servicio del sistema operativo que decide cual es el siguiente proceso que ejecutar. Esto se realiza para mantener un uso del procesador responsable.
- Jerarquía, un proceso puede ser padre o hijo. Todos los procesos tienen un padre excepto el proceso *init*, que es proceso inicial del sistema. *init* es un proceso demonio, lo cual significa que es un proceso que no va a finalizar nunca. Todos los procesos hijo que se queden sin padre, es decir, cuando el padre realice una llamada a *fork* y lance un hijo y después finalice, hará que el hijo se convierta en un proceso zombi haciéndolo hijo de *init*.
- Protección, un proceso ejecutado en modo manual solo puede acceder a la memoria de él mismo, a menos que se realice un tubería. Una tubería es una interfaz que se provee información a dos procesos. Contiene dos descriptores especiales: un descriptor de entrada y un descriptor de salida. Esto se realiza, por ejemplo, cuando quieres utilizar la salida de un proceso para la entrada de otro. También tienes los procesos ejecutados en modo kernel, los cuales tienen acceso a toda la memoria.

La creación de un proceso se realiza cuando el sistema comienza una nueva operación, cuando el usuario ejecuta un binario o cuando otro programa lo llama. Un caso especial es cuando el sistema operativo está arrancando, donde se ejecutan los hilos del kernel y el primer proceso, el cual es el *init*. La creación de un proceso implica la creación de su bloque central de procesos y de su imagen de memoria.

La finalización de un proceso puede darse de forma voluntaria, mediante un `exit()` o un `return` (que sea de forma voluntaria no significa que sea de forma satisfactoria) o de forma involuntaria, mediante el proceso `kill` o por el sistema operativo.

Un proceso puede ser duplicado mediante la llamada `fork`, la cual copia la parte de datos y la de instrucciones a un proceso hijo con distinto `id` ya que es un proceso nuevo. Un proceso puede cambiar todo su contenido por otro mediante la llamada a `exec`.

Cómo funciona la gestión de procesos

La gestión de procesos es un papel fundamental para que un sistema operativo funcione. Por ello es necesario conocer cómo funciona.

Las estructuras del sistema

El sistema tiene varias estructuras que son esenciales para llevar una gestión de procesos, las cuales son:

- Bloque de control de procesos (BCP): es una estructura que ofrece toda la información de un proceso. Esta información es el `pid` (identificador de proceso), el estado, el evento al que espera (en el caso de que esté bloqueado) o los archivos abiertos. En algunos casos el bloque de procesos puede almacenar más o menos datos, según la eficiencia del sistema.
- Tabla de procesos: es una estructura donde se almacena todos los bloques de procesos. Los procesos en esta tabla se identifican a través del `pid`, por lo tanto, ningún proceso puede tener el mismo `pid` que otro si ambos están activos.

Cambio de contexto

En un sistema multitarea es necesario realizar cambios de procesos, debido a que todos los procesos tienen derecho de poder llegar a ser ejecutados por la CPU. Por ello, se decidió crear los cambios de contexto.

Un cambio de contexto es la ejecución de un servicio llamado planificador que se ejecuta en un sistema multitarea. La misión principal es bloquear o parar la ejecución de un proceso para que el siguiente de la lista pueda ejecutar. Esto es debido a que en un ordenador con un único procesador solo se puede ejecutar un proceso. Por ello, se decidió implementar la ejecución concurrente que es alternar los usos de la CPU.

El funcionamiento del cambio del contexto es el siguiente:

1. Almacenar el estado actual del proceso en ejecución: el estado o contexto son todos los valores almacenados en los registros del CPU. Lo que se realiza es guardarlos en memoria principal, más concretamente, en la pila. Después inserta el proceso a la cola de listos.
2. Planificador: selecciona el siguiente programa a ejecutar según una estrategia. Estas estrategias pueden ser:
 - Round Robin: el bcp tiene un nuevo valor de tiempo que es el tiempo que puede estar en ejecución. En el momento que se acaba el tiempo, se le devuelve la rodaja de tiempo (que es igual para todos los procesos) para la próxima vez que se vuelva a ejecutar.
 - Shortest Job First: los trabajos que se estimen que sean más cortos son los primeros en ejecutarse. Esto acarrea el problema de la inanición que significa que un proceso no llega nunca a ejecutarse por ser muy largo. Para ello, la

solución es utilizar tiempos de envejecimiento, que ayuda a que los procesos que no hayan utilizado nunca la CPU pueden llegar a utilizarla.

- FIFO: First in First out. Es un algoritmo de cola el cual pone a ejecutar el primer algoritmo que entra, luego el segundo...
- LIFO: last in First out. Es un algoritmo de pila, el primero que sale es aquel que fue el último en entrar.
- Por prioridades: a los procesos se le puede asignar prioridades según lo importante que sean. En estos casos, siempre se ejecutarán los procesos con mayor prioridad. Cuando un proceso de mayor prioridad entra en la cola de listos y se está ejecutando uno de menor prioridad, automáticamente se realiza el cambio de contexto al de mayor prioridad.

Las políticas o estrategias del planificador se pueden mezclar, dando lugar a políticas como Round Robin + Prioridades. Existen más políticas pero solo he añadido las más conocidas.

3. Restaurar el estado del programa seleccionado: se obtienen los registros almacenados anteriormente del nuevo programa que se va a ejecutar desde la pila de la memoria y se insertan de nuevo en cada uno de los registros del CPU.
4. El programa se establece como ejecutando y el planificador cambia el CPU a la línea del proceso donde está la instrucción siguiente que iba a hacer.

En el caso de que se bloquee un proceso porque está esperando un archivo de disco o por otras circunstancias desarrolladas por eventos, entonces en el paso 1 en vez de volver a insertarlo el proceso en la cola de listos, se inserta en la cola de bloqueados hasta que reciba el evento esperado. Una vez recibido, vuelve la cola de listos.

Gestión de periféricos

Una vez hablado sobre la gestión de los procesos, vamos a hablar sobre la gestión de los periféricos. Los periféricos son muy importantes debido a que son los que nos ayudan a interactuar con el ordenador y son imprescindible para que muchas procesos funcionen.

Concepto de periférico

Un periférico es un dispositivo externo que está conectado a la CPU a través de módulos de entrada y salida. Permiten interactuar con el CPU, de forma que comparten información. Pueden ser:

- Dispositivos de interacción humana a máquina y viceversa: son dispositivos que permiten interactuar en el ordenador, por ejemplo, el ratón, un monitor táctil, el teclado...
- Dispositivos de interacción máquina a máquina: son dispositivos en los que no interacciona una persona, como los modems, un monitor de solo salida...
- Dispositivos de almacenamiento: son aquellos dispositivos que guardan información de forma secundaria. Esta información pueden ser fotos, archivos de un programa, videos, música...
- Dispositivos de interacción desde el medio físico a la máquina: son aquellos dispositivos que reciben información sobre el estado actual de un lugar. Pueden ser relojes, dispositivos de control de humedad...

Un periférico es la suma de las siguientes partes:

- Dispositivo: es el hardware con el que queremos que interactúe con el ordenador.

- Módulo de E/S: es el controlador del dispositivo. Es la parte del dispositivo que lo conecta con el CPU. Es necesario tener un software instalado en el ordenador para poder obtener y enviar los datos de los módulos, de lo contrario no se podría utilizar el dispositivo.

Otras características respecto de los dispositivos es que son mucho más lentos que las CPU, por lo que normalmente no se conectan directamente a la CPU ya que perdería eficacia. Para ello, existen unos registros en el controlador:

- Registro de control: envía la información que debe realizar el dispositivo.
- Registro de estado: guarda el estado de la última acción.
- Registro de datos: intercambia datos entre el dispositivo y la CPU.

Después tienen también una línea de interrupción, donde el dispositivo avisa al módulo de cualquier evento (interrupción de hardware).

Existen tres aspectos importantes respecto a los módulos:

- El direccionamiento: Se pueden acceder a los dispositivos mediante espacios de memoria conjuntos o espacios de memoria separados.
- Unidad de transferencia: según el tipo de dispositivo, la unidad de transferencia puede ser desde un bloque de bytes (leer, escribir...) hasta solo un carácter. Por ejemplo, un teclado manda solo el carácter pulsado al módulo o un disco externo manda el archivo pedido que se quiera leer.
- La interacción: según se necesite, los dispositivos pueden interactuar con el ordenador por:
 - Escritura directa en memoria: no hace falta mandar una interrupción ya que directamente escriben en memoria.
 - Escritura y salida mediante el procesador: es necesario que manden un interrupción para poder escribir en memoria.
 - Mediante DMA: el periférico y la CPU interfieren entre quien usa la memoria, de forma que tiene que pedir permiso para poder entrar a memoria y luego retirarlo.

Cómo funciona la gestión de periféricos

Cuanto más periféricos existan en un ordenador, es más necesario tener una forma de organizarlos para poder utilizar cada uno en su lugar adecuado. Por ello, vamos a ver cuáles son las estructuras de datos en el sistema y cómo funciona la gestión de periféricos.

Las estructuras del sistema

Las estructuras encargadas de llevar a cabo una organización de los dispositivos son:

- Tabla de dispositivos detectados: es una tabla la cual tiene un id de software, un id de hardware y una referencia al evento que produce donde solo están los dispositivos detectados.
- Descriptores de fichero: los cuales identifican para qué está abierto el periférico, si es para lectura, escritura...
- Tabla de drivers: es una tabla donde se almacena los drivers cargados en memoria. Un driver es el software encargado de hacer funcionar un periférico. Estos softwares están almacenados en una carpeta específica del sistema, donde el mismo sistema operativo es capaz de recuperar cada uno de ellos.
- Planificador de E/S: en el caso de que haya varios periféricos dando peticiones, entonces debe haber una organización para realizarlas en orden.

- Otras estructuras: para guardar la información suministrada por el dispositivo.

Por lo tanto, un driver de un dispositivo debe ofrecer:

- Cuál va a ser su petición, su id de hardware y de software.
- Cómo funcionará sus llamadas al sistema para abrir, cerrar, leer, escribir...
- Y un apartado donde se indicará si el dispositivo está cargado en el sistema o descargado.

Funcionamiento de la gestión de periféricos

Los pasos que sigue un periférico son los siguientes:

1. El dispositivo debe estar conectado. La tabla de dispositivos los detecta y almacena sus id y la referencia de interrupción.
2. El ordenador detecta qué tipo de dispositivo es y carga el driver en memoria. Eso hace que la tabla de drivers agregue su programa controlador.
3. Una vez pasado lo anterior, entonces el dispositivo está a la espera de que un programa lo necesite. Por ejemplo, un proceso utiliza la llamada del sistema open en el dispositivo cargado donde para ello necesita conocer el id del dispositivo.
4. El ordenador detecta que se quiere acceder a un periférico y lo busca en la tabla de dispositivos. A través del driver se comunica con el periférico.
5. Una vez realizada las acciones, el driver envía una interrupción de hardware y el planificador de E/S le dice cuando tiene que añadir su contenido. Por lo tanto, el periférico se espera.
6. Una vez que le den el paso, añade su información al buffer otorgado.
7. Si no hay más acciones, el CPU lo pone en estado de espera.

De esta forma, el ordenador es capaz de acceder a los dispositivos que se conectan a él. Existen algunos dispositivos que usan drivers genéricos como el ratón o el teclado, ya que son periféricos de plug and play, es decir, de enchufar y funcionar.

Tema 4: Sistema de archivos

En el siguiente tema vamos a estudiar qué es un archivo, qué operaciones se pueden realizar con archivos y cómo se estructuran. También voy a resumir cuál es el concepto de directorio y todo sobre cómo funciona un sistema de archivos.

Concepto de archivo

Un archivo es un conjunto de datos que tienen relación con un nombre y son almacenados en dispositivos de almacenamiento externo. El archivo o fichero es la unidad más pequeña de información que alberga una unidad lógica de almacenamiento. Los archivos suelen tener los siguientes atributos:

- Nombre: es un conjunto de caracteres que definen el archivo para que una persona lo identifique.
- Identificador: es una etiqueta que define el archivo solo para la máquina.
- Tipo: da la información necesaria para conocer como está estructurado un archivo. El tipo, por ejemplo, puede ser txt (un archivo de solo texto), un archivo ejecutable (un binario ejecutable), jpeg (una foto), mp4 (un video), mp3 (una canción)... Es un atributo fundamental para que el usuario pueda reconocer con qué programa se abre cada archivo. Esto es debido a que si abres, por ejemplo, un ejecutable en un programa que solo admite textos, entonces saldrán caracteres raros porque lo lee como si fuera un txt.
- Referencia: da el puntero hacia el inicio del archivo en el almacenamiento.
- Permisos: decide si el archivo puede ser leído, escrito o ejecutado, además de que quien puede hacer cada uno de esos permisos.
- Fecha: puede almacenar la hora de creación, la hora de modificación...
- Tamaño: almacena la capacidad que ocupa un archivo en el almacenamiento.

Normalmente, los atributos se “almacenan” en los directorios que también están en el disco secundario.

Operaciones con archivos

Un fichero es un TAD, un tipo abstracto de datos. Para que un archivo pueda ser definido, debemos tener unas llamadas al sistema que hagan las siguientes operaciones:

- Crear: para la creación de un archivo lo primero que debemos hacer es buscar espacio libre para almacenarlo, una ubicación en el dispositivo y crear todos sus atributos.
- Abrir: necesito el nombre del archivo y la ruta. Devuelve un descriptor de fichero, el cual es un puntero a la tabla de descriptores de fichero.
- Leer: necesito tener el descriptor de fichero del archivo que quiero leer. También necesito un buffer donde agregar el archivo y el número de bytes que se quieren leer.
- Escribir: necesito tener el descriptor de fichero del archivo que quiero escribir. También necesito un buffer donde estén los datos que quiero agregar y el número de bytes que se quieren escribir.
- Reposicionar: el descriptor de ficheros de un archivo posee un atributo que indica la posición dentro del fichero. Este atributo se puede cambiar con esta llamada, indicando donde quieres ponerlo.
- Cerrar: cierra el archivo que está abierto. En el caso de que varios programas tengan el archivo abierto, entonces reduce el contador de aperturas en 1.

- **Borrar:** existen dos tipos de borrado: el borrado total donde borras los atributos del fichero y los datos, y el borrado típico donde eliminas solo los atributos. La diferencia es la rapidez, ya que el borrado total tarda mucho más que el típico.

Como he detallado anteriormente, existe una estructura del sistema donde se almacena unos atributos específicos de los archivos abiertos. Este es la tabla de descriptores de fichero. La tabla de descriptores de fichero siempre tiene las tres primeras entradas (0,1,2) ocupadas, las cuales son el descriptor de entrada, el descriptor de salida y el descriptor de salida de error. Las demás entradas son archivos abiertos. Cuando realizas la operación de abrir en un fichero, este te devuelve la posición en la tabla de descriptores de fichero. Cada una de las posiciones en la tabla tiene los siguientes atributos:

- **El puntero de posición:** el descriptor almacena la posición actual dentro del fichero.
- **El contador de aperturas de archivo:** sirve para conocer cuántos procesos tienen abiertos el archivo.
- **Dirección del archivo dentro de la memoria principal,** cosa que se realiza para evitar que se escriba o se lea repetidamente en el disco lo que haría que fuese menos eficiente.
- **Permisos:** bloquea que este se pueda leer, escribir o ejecutar.

Una vez conocido como se tratan los archivos, vamos a dar paso a como se organizan, es decir, a los directorios.

Concepto de directorio

Un directorio es un elemento dentro del sistema de archivos que guarda la información de los archivos. Sigue un esquema en árbol, donde el nodo padre es llamado root o raíz y es el inicio de todo directorio. Existen dos formas de direccionar a un directorio o a un archivo:

- **Ruta absoluta:** la ruta empieza desde la raíz del volumen hasta donde está el directorio o el archivo.
- **Ruta relativa:** la ruta empieza desde la carpeta actual donde estás hasta el directorio o el archivo.

Los directorios se pueden realizar de distintas maneras según el sistema de archivos seleccionado. Los esquemas más comunes son:

- **Directorios de único nivel:** todos los archivos se encuentran en la raíz del sistema, donde no se pueden realizar más directorios. Tiene como pega que no puedes tener dos nombres de archivos iguales y que para una persona es imposible manejar muchos archivos sin organización. Todos los usuarios pueden acceder a todos los archivos.
- **Directorios en dos niveles:** se diferencia con el de único nivel en que puede tener varios usuarios con carpetas. Cada usuario tiene una carpeta y dentro de esa carpeta puede crear otras. Pero solo de un nivel, es decir, que solo puede crearlas en la raíz de su usuario. El problema está en el caso de que quieras crear más carpetas dentro de otras carpetas no puedes hacerlo, y que tampoco se pueden compartir carpetas entre usuarios de un mismo almacenamiento.
- **Directorios con estructura en árbol:** los directorios se almacenan en forma de árbol, donde un nodo tiene un padre y puede o no tener un hijo. En caso de que sea el nodo raíz, solo puede tener hijos.
- **Directorios en forma de grafos:** en el caso de que desde un directorio quieras acceder a otro, es necesario ir hacia atrás o ir hacia delante en el árbol. Esto puede llegar a ser

muy ineficiente, por lo que los grafos lo que realizan es crear estructuras de acceso rápido hacia otros archivos o directorios, de forma que desde un directorio puedas acceder a otro de forma rápida.

El sistema de archivos

Para que los archivos y las carpetas puedan tener una organización en el disco y puedan ser reconocidos, es necesario que ese almacenamiento tenga una estructura. Después, es necesario que esa estructura pueda ser reconocida por el sistema operativo. A esa estructura le llamamos sistema de archivos. Ejemplos de sistemas de archivos son FAT, EXT, NTFS...

Estructura

La forma por la cual todos los sistemas de archivos manejan un almacenamiento es a través de bloques. Un bloque es una estructura lógica de un conjunto de datos, donde un bloque es un conjunto de sectores. Un bloque suele tener una capacidad determinada como 512 B, 1024 B, 2048 B, 4096 B... Un sector es una estructura física capaz de almacenar una capacidad determinada. Un almacenamiento secundario posee tanto sectores como la capacidad total del dispositivo entre la capacidad de un solo sector.

Un bloque solo puede contener un archivo o directorio. En ningún momento puede tener varios archivos, porque entonces las referencias no funcionarían correctamente. Esto genera el problema de la fragmentación interna. La fragmentación interna es un problema que sucede cuando un archivo no rellena la capacidad total de un bloque. Cuanto más grande sea un bloque, más probabilidad habrá de que exista fragmentación interna.

Existen varias estructuras según si es por parte del dispositivo de almacenamiento o por parte del sistema operativo:

En el sistema de almacenamiento, todas las estructuras están en bloques:

- Bloque de control de arranque: detalla todos los datos importantes para proceder a iniciar un sistema operativo. Suele ser el primer bloque del almacenamiento. En el caso de que el disco no tenga sistema operativo, este bloque puede estar vacío.
- Bloque de control de volumen: tiene atributos sobre el volumen, como el tamaño del volumen, el inicio del primer bloque de datos, el número de bloques libres...
- Estructuras de directorios: existen unos bloques que almacenan los datos de los archivos y el nombre de su l-nodo asociado en el caso de sistemas de archivos provenientes de Linux, y tabla maestra de archivos, en caso de NTFS o FAT, provenientes de sistemas Windows.
- Bloques por cada archivo que exista en el almacenamiento que contiene todos los atributos de un archivo. En EXT se denominan l-nodos. En FAT y NTFS se encuentran en la tabla maestra de archivos.

En el sistema operativo, existen las siguientes estructuras en memoria:

- Tabla de montaje: contiene información acerca de cada partición que está conectada al sistema.
- Caché de directorios: almacena información sobre la estructura de los directorios recientemente accedidos.
- Tabla global de archivos abiertos: almacena los datos de los archivos abiertos en el sistema.

- Tabla de archivos abiertos por proceso: obtiene el puntero hacia el archivo abierto por el proceso que apunta a la posición en la tabla global de archivos.

De esta forma, un sistema operativo puede reconocer como está estructurado un disco y puede acceder a cada uno de los archivos que están en disco.

Implementación de directorios

Los directorios en el sistema de archivos se pueden implementar de varias formas. Según como se decida, puede verse afectadas la eficiencia, las prestaciones o la fiabilidad del sistema de archivos. Estas formas son:

- Lista lineal: consiste en crear una lista donde cada uno de los elementos tendrá el nombre del archivo y el bloque donde se almacena los datos. Es muy fácil de programar pero tiene la peculiaridad de que para crear archivos, eliminarlos o abrirllos se hace tedioso debido a que para encontrar el archivo tenemos que hacer una búsqueda directorio a directorio, haciendo que sea muy ineficiente.
- Tabla hash: con este método también se organiza los archivos en una lista, pero existe una tabla hash. La tabla hash almacena todos los archivos del sistema. Cuando se quiera crear un archivo nuevo en un ruta, lo que se realiza es un hash de ese archivo y se busca si está en el dispositivo. Si los hash son iguales, entonces ya existe el archivo. Es una forma muy rápida de encontrar los archivos. El problema está en que dos archivos distintos pueden crear un mismo archivo hash, lo que se denomina colisión.

Métodos de asignación

Cuando todos los archivos tienen la misma capacidad o menos que la capacidad de un bloque, todo funciona bien. En cambio, cuando un archivo supera la capacidad del bloque (que es lo común) entonces entra en acción la forma por la cual existen los métodos de asignación. Existen tres formas de enlazar bloques:

- Asignación continua: los bloques de los ficheros están uno detrás de otro. Para que el sistema reconozca cual es el bloque del principio y cuál es el del fin, debe de existir un atributo donde se den esos valores en cada uno de los l-nodos o en las filas de la tabla maestra de archivos. Es la solución más sencilla pero la que acarrea el mayor problema, la fragmentación externa. La fragmentación externa es un problema que surge al haber entre bloques de datos, bloques vacíos debido a la eliminación de ese archivo, lo que origina que no se pueda rellenar ese bloque. Para solucionarlo se necesita desfragmentar el disco, es decir, compactar los bloques de datos, lo cual es un proceso que lleva mucho tiempo.
- Asignación enlaza: Soluciona el problema de la fragmentación de disco de la asignación continua. Sigue habiendo solo dos atributos nuevos como antes, el de bloque de inicio y el de fin. Pero esta vez, al final de cada bloque se direcciona a otro bloque. Mientras ese bloque no sea el de fin, entonces seguirá direccionando. El problema está en que se quita un poco de espacio de cada bloque por el direccionamiento, originando que en archivos grande ocupe casi lo mismo que un archivo el conjunto de direccionamientos.
- Asignación indexada: para el uso de este método, es indispensable tener un tabla de localización de archivos. Lo que realiza este método es tener una lista de cada uno de los bloques de datos de archivo. La posición cero es la primera y la posición n es la última. Entre medias, existen las posiciones uno, dos, tres... hasta n-1. El problema de la asignación indexada es que los bloques de datos pueden estar distribuidos de forma

aleatoria por todo el volumen, haciendo que cada vez que quieras acceder al siguiente bloque, tengas que buscarlo.

Gestión del espacio libre

Es una parte importante, debido a que se necesita conocer si existen bloques libres y qué bloques libres hay en el almacenamiento. Entonces para ello, se crearon estos diversos métodos:

- Vector de bits: varios bloques determinados del sistema se utilizan para almacenar una serie de bits. Cada bit representa un bloque, y el valor del bit representa si está libre o no, donde un uno representa que sí está libre y un cero representa que no lo está. Este método es ineficiente debido a que para encontrar bloques nuevos tienes que ir posición por posición en el vector.
- Lista enlazada: el sistema de archivos almacena en el bloque de control el primer bloque vacío del sistema. Este bloque enlaza después al siguiente bloque vacío. De forma que todos los bloques están unidos y cuando quieras agregar un archivo lo único que tienes que hacer es conseguir el primer bloque y actualizar cual es el próximo primer bloque vacío. El problema está en que para conseguir varios bloques libres, tienes que leer cada uno de los bloques para conocer cuál es el siguiente libre.
- Agrupamiento: en el primer bloque vacío, se agregan todos los bloques vacíos. De este modo el sistema solo tendrá que leer una vez el primer bloque vacío y obtendrá la lista de todos los bloques vacíos.
- Recuento: En el caso de que tengamos una asignación continua, tenemos el hecho de que varios bloques se van a quedar libres a la vez, por lo que obtenemos solo el primer bloque más un número n de todos los bloques libres siguientes. Para poder almacenarlos, se necesita una lista donde se almacenarían los datos anteriores.

Con estos métodos, se pueden obtener los bloques vacíos. Algunos son más ineficientes que otros, cosa de la cual se tiene que preocupar el diseñador del sistema de archivos.

Tema 5: Gestión de memoria

En este tema vamos a conocer cómo se gestiona la memoria principal. Antes de eso, debemos recordar algunos conceptos importantes. Recordaremos que es un programa, un proceso, una imagen de proceso, un entorno monoprogramado y multiprogramado:

- Programa: conjunto de instrucciones que permiten a un ordenador realizar acciones diversas, como leer un documento, ver un vídeo, navegar por internet...
- Proceso: es un programa que está en memoria listo para ejecutar o ejecutándose. Contiene código, pila y datos.
- Imagen de proceso: es el grupo de direcciones de memoria principal de un proceso.
- Entorno monoprogramado: es aquel sistema operativo que solo soporta un único proceso como máximo en memoria.
- Entorno multiprogramado: es aquel sistema operativo que soporta más de un proceso en memoria, lo que hace que se mejore la utilización de la CPU porque en ningún momento estaría parada sin hacer nada.

Partes de memoria que tiene un proceso

Un proceso tiene varias partes diferenciadas donde se almacena distinta información:

- El código: son las instrucciones que tiene un proceso. Puede ser compartida, leída y, por supuesto, debe ser ejecutable.
- Los datos: creadas por las instrucciones. Puede ser compartida, leída y escrita.
- La pila: es una zona especial de memoria donde la CPU almacena el contenido de los registros cuando sea necesario. Va desde direcciones más altas de memoria a más bajas. No es compartida.
- La región de Heap: zona la cual es reservada en tiempo de ejecución que soporta memoria dinámica. No es compartida, ni leíble y ni se puede escribir. Crece hacia zonas más altas.
- Ficheros proyectados: es una zona de memoria la cual sirve para mostrar el archivo. Esta zona es compartida.
- Memoria compartida: es una zona la cual la comparten varios procesos. El tipo de protección la especifica el programa al crearla.
- Pilas de hilos: cada hilo necesita una pila propia, que será usada igual que la pila del proceso.

Soporte de memoria virtual

La memoria virtual es una técnica de gestión de memoria que aumenta el almacenamiento que el sistema operativo puede usar. Esto se diseñó debido a que existen procesos que poseen más memoria que la memoria principal, por lo que no se podía cargar entera. Entonces lo que se diseñó es un espacio de disco que funcionará como aumento de memoria principal, aunque antiguamente se utilizaban memorias expansivas o extensibles. Por lo tanto, esto genera que existan más direcciones de memoria que antes. Existen tres tipos de implementación de la memoria virtual:

- Paginación: divide los programas en partes denominadas páginas, a la misma vez que divide la capacidad total de la memoria también en páginas a lo que se le llama marco de página. Los procesos tienen una estructura denominada tabla de páginas. Esta estructura almacena la ubicación del marco que contiene cada una de sus páginas. Entonces, las direcciones se generan a partir del número de página y el

reposicionamiento. El número de página es usado como índice en la tabla y el desplazamiento se utiliza para componer la dirección real.

- Segmentación: divide los programas en tamaños variables. A estos tamaños se le considera segmentos. Un programa puede ser el conjunto de varios segmentos o de solo uno. La segmentación del programa la realiza el compilador. Para las direcciones de memoria se necesitan dos valores: el número de segmento y el desplazamiento en el segmento. Las bibliotecas dinámicas se suelen representar en segmentos independientes.
- Segmentación paginada: combina los dos métodos anteriores. La segmentación proviene porque los programas se segmentan y la paginación es debido a que se el segmento se particiona como los marcos de página. Entonces, cuando quieres llevarte una página, necesitas el número de segmento, el número de página y el desplazamiento. Entonces gracias al número de segmento recuperas la parte de memoria mediante el puntero de tablas de segmento. Mediante el número de página y la tabla recuperas la página dentro del segmento. Y mediante el desplazamiento consigues seguir el proceso.

La gestión de memoria

La gestión de la memoria tiene los siguientes objetivos:

- Traducir las referencias a direcciones físicas.
- Proteger los diferentes espacios de memoria para que un proceso no acceda a la memoria de otro sin permiso. Los procesos no pueden compartir direcciones de memoria.
- Tener la capacidad de poder compartir partes de memoria para que varios procesos puedan acceder a ella.
- Conocer cómo se modulan los diferentes programas para posteriormente poder organizarlos. Para ello, hay que tener en cuenta que los datos de un proceso no son siempre del mismo tamaño por lo tanto cada región se debería adaptar de una forma concreta.
- Ser capaz de rellenar la memoria física de los diferentes módulos y programas, de forma que evite que se pierda memoria por culpa de la fragmentación, poder ejecutar procesos mayores que la memoria física disponible.

Mecanismos para la gestión de memoria

La gestión de memoria es importante debido a que un mal uso de ella puede generar fallos en el sistema. También porque los sistemas operativos son degenerativos en memoria, es decir, tras el paso del tiempo que lleve un sistema operativo sin reiniciarse, menos memoria tendrá porque peor la estará gestionando. Por ello, muchos servidores son reiniciados después de un lapso para que vuelvan a funcionar en un óptimo rendimiento.

Para gestionar la memoria se utilizan diversas funciones:

- Reservar memoria: se da en el caso de que un programa quiera ejecutar y necesite un espacio en memoria para su imagen de proceso. Normalmente no se le asigna en memoria principal. Dependiendo de si tiene soporte para archivos o no se le dan unas condiciones u otras. Entonces, una vez obtenidas estas dos partes, es cuando se decide si es enviada la región de la memoria de swap (solo si es privada) o a la página de soporte (en caso de ser compartida). La pila del programa tiene argumentos del programa.

- Liberar espacio reservado: todo su memoria reservada se libera, para que otros programas sean capaces de reutilizarla. Este caso se puede dar mediante una solicitud explícita, mediante un programa que ha finalizado o en el caso de que se ejecute un `exec` para cambiar de programa.
- Cambiar tamaño: un proceso puede necesitar más tamaño o menos, por lo que mediante funciones como `malloc` administran más espacio de forma dinámica. Se debe evitar el solapamiento en caso de que se maximice la memoria del proceso.
- Duplicar espacios de memoria: se da cuando un proceso realiza una copia de sí mismo mediante un `fork`. Se realiza una copia exacta del proceso en cuestión.
- Proyección en memoria: se trata de un fichero que es llevado a memoria mediante un proceso. El sistema operativo es el responsable de ponerle las propiedades necesarias para permitir o no la compartición del fichero. Se utiliza desde cargar bibliotecas dinámicas hasta para acceder a ficheros sin uso de leer y escribir.

Políticas de la gestión de memoria

Las políticas utilizadas en la gestión de memoria se dividen en tres partes:

- El espacio de trabajo del kernel y los procesos: el espacio del kernel es compartido con todos los procesos. Eso no significa que todos los procesos puedan acceder a todo el espacio del kernel, sino todo lo contrario, cada proceso tiene sus permisos. Casi la totalidad de las llamadas al sistema están en memoria pero están protegidas contra lectura, escritura y ejecución.
- Los parámetros de la memoria: los parámetros que hay que tener en cuenta son el tamaño de la página, para poder dividir cada uno de los programas; el conjunto residente, que es el número de páginas que tiene un proceso en memoria principal en un estado dado; y el grado de multiprogramación, que es el número de procesos en memoria que puede haber en un tiempo determinado. Dos de los problemas que pueden suceder es el `swapping`, que es el intercambio constante y desequilibrado de datos entre memoria principal y secundaria; y la hiperpaginación, que se produce cuando aparecen muchos fallos de página a la vez.
- Aspectos de la administración de la paginación: las tablas de páginas las crea el sistema operativo en el momento de ejecutar un programa. Entonces, consulta la unidad de gestión de memoria (que es un dispositivo que gestiona los accesos a memoria dentro de la CPU) en la traducción de las direcciones de memoria virtual. Al final, se verifica y se actualiza en el caso de que exista un fallo de página. Existen dos tipos de movimientos:
 - Desde memoria secundaria a principal (por demanda): el sistema operativo pide que se lleve la página deseada a la memoria física.
 - Desde memoria principal a secundaria (por expulsión): el sistema decide retirar una página en memoria principal para poder entrar otra. En caso de que existiera memoria principal sufriendo, este paso no se haría y se pasaría la página directamente.

También, existen políticas de reemplazo. Estas políticas deciden qué páginas deben ser sacadas de memoria física y qué páginas deben ser agregadas a memoria secundaria. Estos reemplazos pueden ser locales (dentro de un proceso) o globales (intercambio de procesos), tiene algoritmos como FIFO, LRU, óptimo (reemplaza la página que más tiempo va a esperar) o reloj o puede tener la propiedad de que solo puede ser reemplazo local, es decir, que una parte del proceso tiene que estar si o si en memoria principal.

Tema 6: Aspectos avanzados

En el presente tema vamos a estudiar porqué hay que proteger nuestro sistema, qué es ser tolerando a los fallos, qué es un sistema distribuido y qué es un sistema empotrado.

La seguridad en los sistemas operativos

La seguridad en cualquier ámbito es importante. En un sistema el cual tiene muchos datos nuestros es todavía más. El ordenador ha hecho que muchas acciones que hacíamos a parte en nuestro día a día se hagan en él. Ahora, eso es un peligro porque si son capaces de acceder a él, lo podemos perder todo.

Para que algo pueda ser considerado seguro (ya sea una conversación o un sistema operativo) debe de haber:

- Confidencialidad: implica que solo los entes permitidos puedan leer un objeto. También implica conocer al autor del envío del objeto.
- Integridad: implica que solo los entes permitidos puedan modificar los objetos.
- Disponibilidad: implica que los elementos estén siempre operativos para los entes permitidos.

Una vez que consideramos que algo puede ser seguro, debemos conocer cómo podemos saber quiénes son los entes permitidos. Para ello existen dos formas de seguridad: la seguridad física o la seguridad lógica. Para la seguridad lógica existen varios métodos para autenticar al usuario los cuales son por contraseña, por biometría o por tarjetas o objetos que solo la persona en cuestión puede tener. En el caso de que alguna forma de seguridad fuera burlada, entonces estaríamos hablando sobre una amenaza.

Las amenazas son formas de romper los requisitos de seguridad. Existen cuatro tipos de amenazas:

- Interrupción: ataca contra la disponibilidad. Hace que un elemento del sistema sea inaccesible ya sea porque se haya destruido o inhabilitado.
- Intercepción: ataca contra la confidencialidad. Un ente no permitido se hace con el objeto.
- Modificación: ataca contra la integridad y a la confidencialidad. Un ente no permitido se hace con el objeto y lo modifica.
- Invención: ataca contra la integridad. Un ente no permitido falsifica el objeto y añade datos nuevos.

Para evitar estas amenazas, los sistemas operativos tienen dos principios básicos: el principio de necesidad de saber y el de diseño intrínseco de seguridad. Después, tenemos cinco principios complementarios que son el principio de mínimo privilegio (un usuario o programa debe tener el mínimo acceso a privilegios), el de ahorro de mecanismos (los mecanismos de seguridad deben ser simples pero efectivos), el de aceptación (los mecanismos no deben interferir demasiado con el usuario autorizado), el de mediación total (todos los accesos deben ser cotejados) y el de diseño abierto (los mecanismos de seguridad deben ser de libre acceso).

Para llevar a cabo los principios, debemos tener en cuenta las políticas y los mecanismos de seguridad. La política de seguridad identifica que nivel de seguridad se necesita en un sistema operativo y los mecanismos buscan cuales son las mejores formas para implantar las políticas. Las políticas las deciden las personas que están al cargo de proteger los ordenadores. Entre los mecanismos puedes elegir entre protección (evitar que se haga un uso indebido del sistema),

seguridad (solo los recursos pueden ser usados por entes permitidos) y salvaguarda (evitar más accidentes) y la matriz de protección (donde cada elemento dice que permisos puede acceder un usuario para un objeto).

Los fallos y errores en el sistema operativo

Un sistema operativo no está exento de no tener fallos. Eso provoca que en varias ocasiones nuestro sistema colapse y deje de funcionar. Por ello, el sistema operativo debe ser tolerante a los fallos. Esto significa que debe conservar la máxima capacidad operativa y rescatar los máximos datos posibles, por ello debe poder mantener el sistema lo más perfecto posible a pesar de que pueda haber fallos. Los sistemas operativos que más tolerancia a fallos deben soportar son aquellos que están relacionados con circunstancias críticas como el sistema a bordo de un vehículo o sistemas de difícil mantenimiento. Los fallos tienen distintas formas:

- Comportamiento anómalo: el sistema realiza acciones inesperadas.
- Peligro: las acciones que está tomando un sistema puede llevar a accidente.
- Error: malentendido en los requisitos, el diseño y/o el código.
- Fallo: el sistema no es capaz de realizar la acción.
- Defecto: error que proviene del software.

Para evitar los distintos fallos que pueden suceder, un sistema debe tener fiabilidad, disponibilidad, mantenibilidad, confiabilidad y estabilidad. Además el sistema debe tener una consistencia entre evitar todos los fallos posibles y intentar sufragar los fallos ya aparecidos. Una técnica para tolerar fallos es la redundancia que utiliza varios sistemas interconectados que mediante aparatos externos se puede examinar de donde procede el fallo.

Los sistemas distribuidos

Los sistemas distribuidos son sistemas que tienen varias piezas de una máquina separadas, donde un error en una de las piezas logra que todo el sistema colapse, ya que están todas conectadas y todas son necesarias. Desde un punto de vista físico, es un conjunto de procesadores independientes, y, desde el punto de vista lógico, son procesos que se ejecutan a la vez cada uno en un procesador distinto. Aunque existan varios procesadores, en realidad solo existe un equipo. Es decir, existe un único sistema operativo capaz de manejar distintos dispositivos a la vez.

Tienen como ventajas que comparten recursos, son escalables, tienen un muy alto rendimiento, mucha fiabilidad, disponibilidad y buena relación calidad coste.

Como desventajas tiene que son muy complejos, deben estar siempre interconectados, existe un grado menor de seguridad y el sistema operativo que utiliza es mucho más complicado.

Los sistemas de tiempo real

El tiempo real es un tipo de procesamiento en el que se tiene en cuenta el momento en el cual se produce el resultado lógico. Normalmente, los procesos que tienen este tipo de procesamiento son procesos críticos, donde el tiempo es crucial. Existen tres clasificaciones: Hard-real time (es estrictamente obligatorio obtener el resultado en el tiempo acordado), Soft-real time (el tiempo estipulado debe cumplirse pero no es obligatorio) y firm-real time (los datos fuera de tiempo no sirven, pero se puede solventar el sistema). El tiempo estipulado se puede clasificar según si son tareas periódicas o no periódicas.

Un sistema operativo en tiempo real suele tener rápidos cambios de contexto (planificación apropiativa con expulsión), mínima funcionalidad, alta capacidad de respuesta, multitarea con

compartición entre procesos y uso de guardados rápidos de archivos secuenciales. Los sistemas críticos cumplen con ser deterministas, sensibles, acceso total por parte de usuarios, fiables y con tolerancia a fallos. Los sistemas operativos en tiempo real suelen ser utilizados en sistemas empotrados que son sistemas con solo un propósito específico donde el software y hardware son desarrollados únicamente para ese fin.

Resumen general

Un sistema operativo es un conjunto de programas que son capaces de poder controlar y gestionar los recursos proporcionados por el hardware.

El arranque de un ordenador se ve cumplimentado por dos pasos: el arranque de hardware, el cual comprueba los dispositivos y agrega en la memoria principal el programa de arranque; y el arranque de software, el cual se realiza mediante el programa cargado en memoria anteriormente.

Un evento es una señal que indica que debe interrumpir el curso de la ejecución actual para solventarse. Existen dos tipos de eventos, los eventos software donde pertenecen las llamadas al sistema y las interrupciones software; y los eventos hardware de donde provienen las excepciones y las interrupciones hardware.

Los recursos del sistema operativo se gestionan mediante servicios, los cuales son:

- Gestión de procesos

Un proceso es un conjunto de instrucciones y datos que están en memoria principal para ser procesador por la CPU. Los procesos tienen recursos, accesos a estructuras de control, estado, jerarquías y permisos o protección. Un proceso se crea cuando el usuario o el sistema lo llama y se cierra de forma voluntaria mediante `exit` o `return`; o de forma involuntaria mediante la función `kill`. Puede ser duplicado mediante la llamada `fork`.

Las estructuras que tienen relación con la gestión de procesos son el bloque de procesos y la tabla de procesos.

El funcionamiento de la gestión de procesos es mediante el planificador. En sencillos pasos, un proceso sale de ejecución de forma voluntaria porque espera una interrupción o de forma involuntaria porque lo retira el planificador, se guarda todos los datos del proceso sacado, el planificador decide qué proceso es el siguiente mediante un algoritmo, se restaura el estado del proceso seleccionado y se establece como ejecutando.

- Gestión de periféricos

Un periférico es un dispositivo externo al sistema que está conectado a la CPU a través de módulos de entrada y salida. Los módulos de entrada/salida son los controladores del dispositivo. Disponen de tres registros: uno de control, otro de estado y uno último de datos.

Las estructuras que afectan al sistema son la tabla de dispositivos detectados, los descriptores de fichero, la tabla de drivers, el planificador de entrada/salida y otras estructuras que sirven para almacenar archivos u otras cosas.

El funcionamiento que tiene es el siguiente: el dispositivo se conecta y lo detecta el sistema, agregándolo a la tabla de dispositivos conectados; el ordenador carga el driver en memoria; el ordenador detecta que quieres acceder a un periférico y lo busca en la tabla de dispositivos; el módulo de entrada/salida envía una interrupción de software; cuando le dan el paso, añade el contenido en el buffer otorgado.

- Gestión del sistema de archivos

Un archivo es un conjunto de datos que tiene relación con un nombre y es almacenado en dispositivos de almacenamiento externo. El fichero es la unidad más pequeña de información que alberga una unidad lógica de almacenamiento. Suelen tener diversos atributos. Las operaciones que se pueden realizar con los archivos son leer, escribir, mover el puntero de lectura, abrir, cerrar, crear y borrar.

Un directorio es un elemento dentro del sistema de archivos que guarda la información de los ficheros. Existen varios tipos de directorios, los de un único nivel, los de dos, los de estructura en árbol y los que son en forma de grafo.

Las estructuras del sistema de archivos por parte del sistema operativo son: la tabla de montaje, la caché de directorios, la tabla global de archivos abiertos y la tabla de archivos abiertos por proceso.

Las estructuras por parte del almacenamiento son: el bloque de control de arranque, el bloque de control de volumen, las estructuras de directorios y los bloques de archivos.

Los sistemas de archivos pueden tener distintos métodos de asignación como la asignación continua, la asignación enlazada o la asignación indexada. El espacio libre en un sistema de archivos se puede gestionar mediante un vector de bits, una lista enlazada, agrupamiento o recuento.

- Gestión de memoria

La memoria principal es el lugar de donde un sistema operativo obtiene todos los datos, por lo que cualquier cosa que se quiera obtener, tiene que estar en memoria principal. En memoria principal están ubicados los códigos de los procesos, los datos, las pilas, las regiones de Heap, los ficheros proyectados, la memoria compartida y las pilas de los hilos.

En el caso de que la memoria principal sea insuficiente existe la memoria virtual. La memoria virtual es memoria ubicada en un almacenamiento secundario donde se guarda información de la memoria principal que no está en uso. Existen tres tipos de implementación de memoria virtual: la paginación, la segmentación y la segmentación paginada.

Para gestionar la memoria se utilizan las estructuras de reservar memoria, liberar espacio, cambiar tamaño, duplicar espacios de memoria y proyectar en memoria.

La memoria sigue una política según el espacio de trabajo, los parámetros de la memoria y los aspectos de la administración de páginas.

Los sistemas operativos necesitan disponer de una mínima seguridad. Seguridad significa que un objeto es confidencial, es íntegro y está disponible. Cuando la seguridad se ve afectada se dice que hay una amenaza. Para poder evitar las amenazas, existen varios principios que debe seguir todo sistema, además de llevar políticas y medidas correctas.

Los sistemas operativos no son capaces de poder librarse de todos los errores, por eso deben, en algunos casos, tolerarlos. Existen varios tipos de errores que pueden suceder en un sistema operativo.

En diversos campos de la actualidad, existen dos tipos de sistemas muy particulares los cuales son los sistemas distribuidos y los sistemas de tiempo real.

Los sistemas distribuidos son CPU interconectadas que funcionan a través de un único sistema operativo. Son sistemas muy sensibles que al más mínimo fallo puede hacer caer a toda la máquina.

Los sistemas en tiempo real son máquinas diseñadas expresamente para cumplir una función. Estas máquinas poseen un hardware y software específico. Son máquinas en tiempo real porque tienen un tiempo específico para poder realizar una tarea.

Referencias

Los materiales utilizados para el estudio y elaboración del proyecto son:

Apuntes y datos de aula global. (2020). Obtenido de <https://aulaglobal.uc3m.es/>.

Jesús Carretero, Félix García, Pedro de Miguel Ansagasti, & Fernando Pérez (2001). *SISTEMAS OPERATIVOS: Una visión aplicada*. Mc-Graw Hill.

Abraham Silberschatz, Peter Baer, Greg Gagne (2005). *Fundamentos de Sistemas Operativos*, Edición 7ma, . Mc-Graw Hill.