

Sistemas Operativos

Introducción, historia y conceptos de los sistemas operativos

Clase 3 - Unidad I

Lic. Alexis Sostersich
sostersich.alexis@uader.edu.ar

Teoría – Sistemas Operativos

Introducción, historia y conceptos de los sistemas operativos

Unidad I – Clase 3

- Conceptos de los SO.
 - Procesos. Memoria y Archivos.
- Llamadas al sistema.
- Estructura de un sistema operativo.





Conceptos de los SO

Procesos

Memoria y Archivos



Procesos

Un proceso es en esencia un programa en ejecución.

Cada proceso tiene asociado un espacio de direcciones, una lista de ubicaciones de memoria que va desde algún mínimo (generalmente 0) hasta cierto valor máximo, donde el proceso puede leer y escribir información.

Podemos decir que un proceso es un recipiente que guarda toda la información necesaria para ejecutar un programa.

Procesos

Cada cierto tiempo, el sistema operativo decide detener la ejecución de un proceso y empezar a ejecutar otro;

Cuando un proceso se suspende en forma temporal como en el ejemplo anterior, debe reiniciarse después exactamente en el mismo estado que tenía cuando se detuvo.

Esto significa que toda la información acerca del proceso debe guardarse en forma explícita en alguna parte durante la suspensión.

Procesos

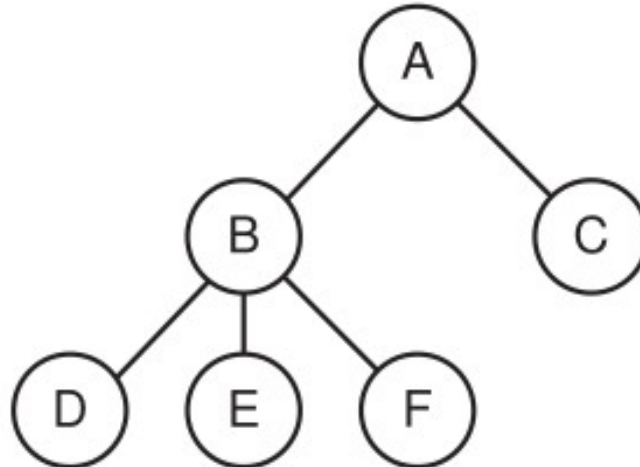
Cuando un proceso se suspende en forma temporal, todos estos apuntadores deben guardarse de manera que una llamada a read que se ejecute después de reiniciar el proceso lea los datos apropiados. En muchos sistemas operativos, toda la información acerca de cada proceso (además del contenido de su propio espacio de direcciones) se almacena en una tabla del sistema operativo, conocida como la tabla de procesos, la cual es un arreglo (o lista enlazada) de estructuras, una para cada proceso que se encuentre actualmente en existencia.

Un proceso (suspendido) consiste en su espacio de direcciones, que se conoce comúnmente como imagen de núcleo y su entrada en la tabla de procesos, que guarda el contenido de sus registros y muchos otros elementos necesarios para reiniciar el proceso más adelante.

Procesos hijos

Un proceso puede crear uno o más procesos aparte (conocidos como procesos hijos) y estos procesos a su vez pueden crear procesos hijos, llegamos rápidamente la estructura de árbol de procesos.

Los procesos relacionados que cooperan para realizar un cierto trabajo a menudo necesitan comunicarse entre sí y sincronizar sus actividades (comunicación entre procesos)



Espacios de direcciones (Memoria)

Cada computadora tiene cierta memoria principal que utiliza para mantener los programas en ejecución.

En un sistema operativo muy simple sólo hay un programa a la vez en la memoria. Para ejecutar un segundo programa se tiene que quitar el primero y colocar el segundo en la memoria.

Los sistemas operativos más sofisticados permiten colocar varios programas en memoria al mismo tiempo. Para evitar que interfieran unos con otros (y con el sistema operativo), se necesita cierto mecanismo de protección. Aunque este mecanismo tiene que estar en el hardware, es controlado por el sistema operativo.

Espacios de direcciones (Memoria)

Existe una técnica llamada memoria virtual, en la cual el sistema operativo mantiene una parte del espacio de direcciones en memoria principal y otra parte en el disco, moviendo pedazos de un lugar a otro según sea necesario.

En esencia, el sistema operativo crea la abstracción de un espacio de direcciones como el conjunto de direcciones al que puede hacer referencia un proceso.

El espacio de direcciones se desacopla de la memoria física de la máquina, pudiendo ser mayor o menor que la memoria física.

Archivos

Otro concepto clave de casi todos los sistemas operativos es el sistema de archivos. Una de las funciones principales del sistema operativo es ocultar las peculiaridades de los discos y demás dispositivos de E/S, presentando al programador un modelo abstracto limpio y agradable de archivos independientes del dispositivo.

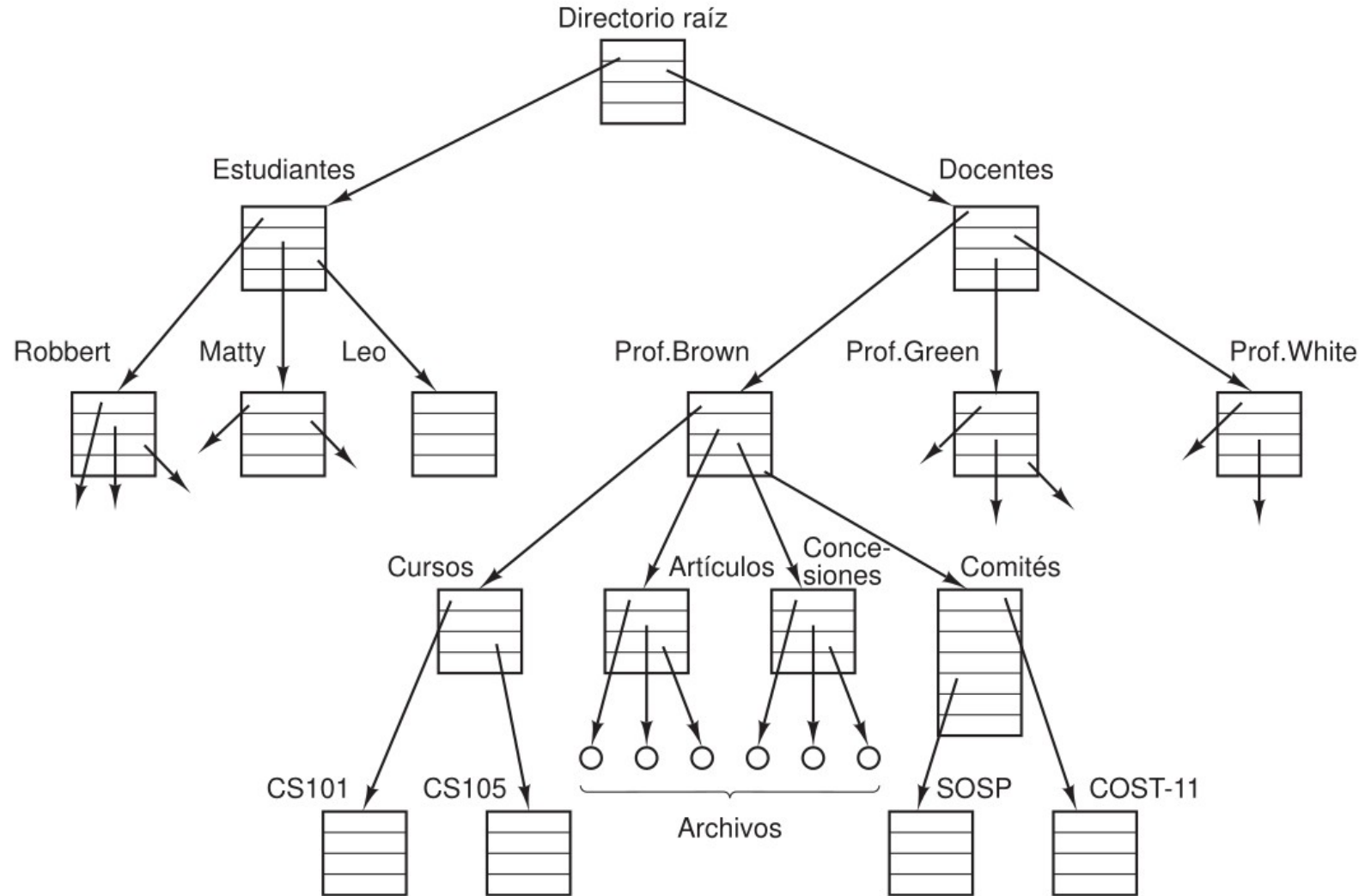
Sin duda se requieren las llamadas al sistema para crear los archivos, eliminarlos, leer y escribir en ellos. Antes de poder leer un archivo, debe localizarse en el disco para abrirse y una vez que se ha leído información del archivo debe cerrarse, por lo que se proporcionan llamadas para hacer estas cosas.

Archivos

Para proveer un lugar en donde se puedan mantener los archivos, la mayoría de los sistemas operativos tienen el concepto de un directorio como una manera de agrupar archivos.

Así, se necesitan llamadas al sistema para crear y eliminar directorios. También se proporcionan llamadas para poner un archivo existente en un directorio y para eliminar un archivo de un directorio. Las entradas de directorio pueden ser archivos u otros directorios. Este modelo también da surgimiento a una jerarquía (el sistema de archivos)

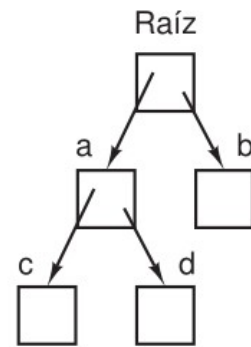
Sistema de archivos



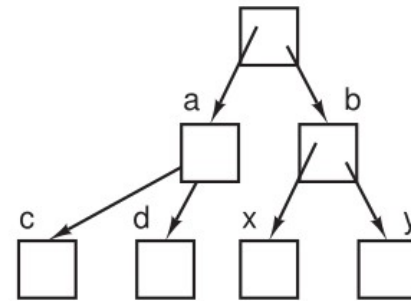
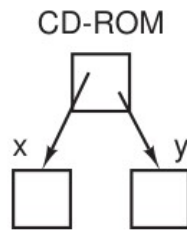
Montaje

En UNIX tenemos el concepto de sistema de archivos montado. Casi todas las computadoras personales tienen una o más unidades ópticas en las que se pueden insertar los CD-ROMs y los DVDs. También tienen puertos USB, a los que se pueden conectar memorias USB.

Para ofrecer una manera elegante de lidiar con estos medios removibles, UNIX permite adjuntar el sistema de archivos en un CD-ROM o DVD al árbol principal.



(a)



(b)

Archivo especial

Otro concepto importante en UNIX es el archivo especial. Los archivos especiales se proporcionan para poder hacer que los dispositivos de E/S se vean como archivos. De esta forma se puede leer y escribir en ellos utilizando las mismas llamadas al sistema que se utilizan para leer y escribir en archivos. Existen dos tipos de archivos especiales: archivos especiales de bloque y archivos especiales de carácter.

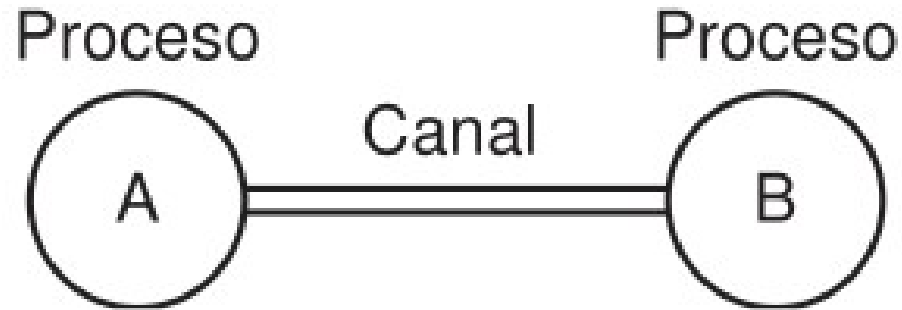
Los archivos especiales de bloque se utilizan para modelar dispositivos que consisten en una colección de bloques direccionables al azar, tales como los discos. Al abrir un archivo especial de bloque y leer, por decir, el bloque 4, un programa puede acceder de manera directa al cuarto bloque en el dispositivo sin importar la estructura del sistema de archivos que contenga.

De manera similar, los archivos especiales de carácter se utilizan para modelar impresoras, módems y otros dispositivos que aceptan o producen como salida un flujo de caracteres

Canales

La última característica que veremos en esta descripción general está relacionada con los procesos y los archivos: los canales.

Un canal (pipe) es un tipo de pseudoarchivo que puede utilizarse para conectar dos procesos



Entrada/salida

Todas las computadoras tienen dispositivos físicos para adquirir entrada y producir salida. Después de todo, ¿qué tendría de bueno una computadora si los usuarios no pudieran indicarle qué debe hacer y no pudieran obtener los resultados una vez que realizara el trabajo solicitado?

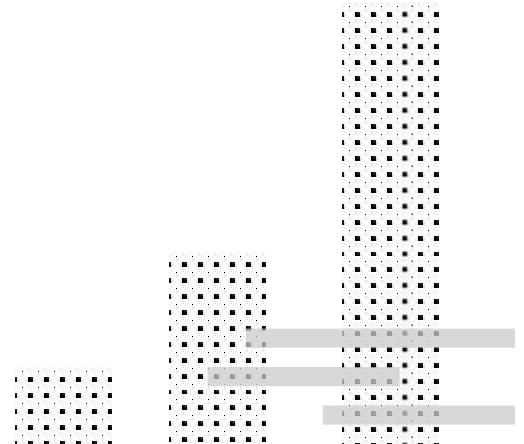
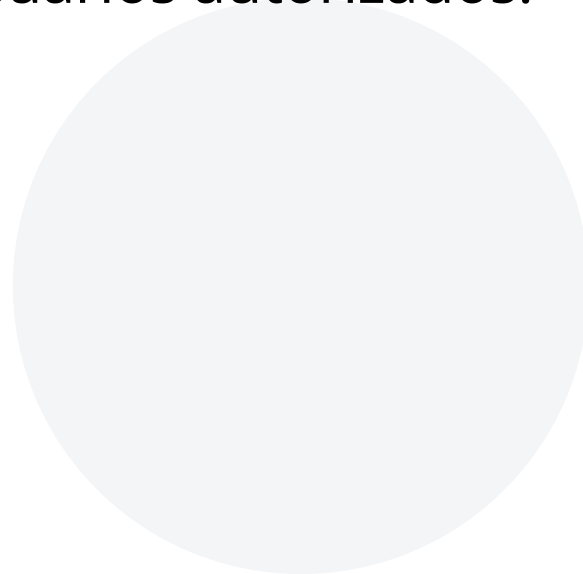
Existen muchos tipos de dispositivos de entrada y de salida, incluyendo teclados, monitores, impresoras, etcétera.

Es responsabilidad del sistema operativo administrar estos dispositivos. En consecuencia, cada sistema operativo tiene un subsistema de E/S para administrar sus dispositivos de E/S. Parte del software de E/S es independiente de los dispositivos, es decir, se aplica a muchos o a todos los dispositivos de E/S por igual. Otras partes del software, como los drivers de dispositivos, son específicas para ciertos dispositivos de E/S.

Protección

Las computadoras contienen grandes cantidades de información que los usuarios desean proteger y mantener de manera confidencial.

Es responsabilidad del sistema operativo administrar la seguridad del sistema de manera que los archivos, por ejemplo, sólo sean accesibles para los usuarios autorizados.



Protección

Como un ejemplo simple, sólo para tener una idea de cómo puede funcionar la seguridad, considere el sistema operativo UNIX. Los archivos en UNIX están protegidos debido a que cada uno recibe un código de protección binario de 9 bits. El código de protección consiste en tres campos de 3 bits, uno para el propietario, uno para los demás miembros del grupo del propietario (el administrador del sistema divide a los usuarios en grupos) y uno para todos los demás. Cada campo tiene un bit para el acceso de lectura, un bit para el acceso de escritura y un bit para el acceso de ejecución. Estos 3 bits se conocen como los bits rwx.

Protección

Por ejemplo, el código de protección `rwxr-x--x` indica que el propietario puede leer (r), escribir (w) o ejecutar (x) el archivo, otros miembros del grupo pueden leer o ejecutar (pero no escribir) el archivo y todos los demás pueden ejecutarlo(pero no leer ni escribir). Para un directorio, x indica el permiso de búsqueda. Un guión corto indica que no se tiene el permiso correspondiente.

El shell

El sistema operativo es el código que lleva a cabo las llamadas al sistema. Los editores, compiladores, ensambladores, enlazadores e intérpretes de comandos en definitiva no forman parte del sistema operativo, aun cuando son importantes y útiles.

Aunque el shell no forma parte del sistema operativo, utiliza con frecuencia muchas características del mismo y, por ende, sirve como un buen ejemplo de la forma en que se pueden utilizar las llamadas al sistema. También es la interfaz principal entre un usuario sentado en su terminal y el sistema operativo, a menos que el usuario esté usando una interfaz gráfica de usuario. Existen muchos shells, incluyendo sh, csh, ksh y bash. Todos ellos soportan la funcionalidad antes descrita, que se deriva del shell original (sh).



Llamadas al sistema



Llamadas al sistema

La interfaz entre los programas de usuario y el sistema operativo trata principalmente acerca de cómo lidiar con las abstracciones que presentan los SO.

Las llamadas al sistema disponibles en la interfaz varían de un sistema operativo a otro (aunque los conceptos subyacentes tienden a ser similares)

Cualquier computadora con una sola CPU puede ejecutar sólo una instrucción a la vez. Si un proceso está ejecutando un programa de usuario en modo usuario y necesita un servicio del sistema, como leer datos de un archivo, tiene que ejecutar una instrucción de trap para transferir el control al sistema operativo. Después, el sistema operativo averigua qué es lo que quiere el proceso llamador, para lo cual inspecciona los parámetros. Luego lleva a cabo la llamada al sistema y devuelve el control a la instrucción que va después de la llamada al sistema.

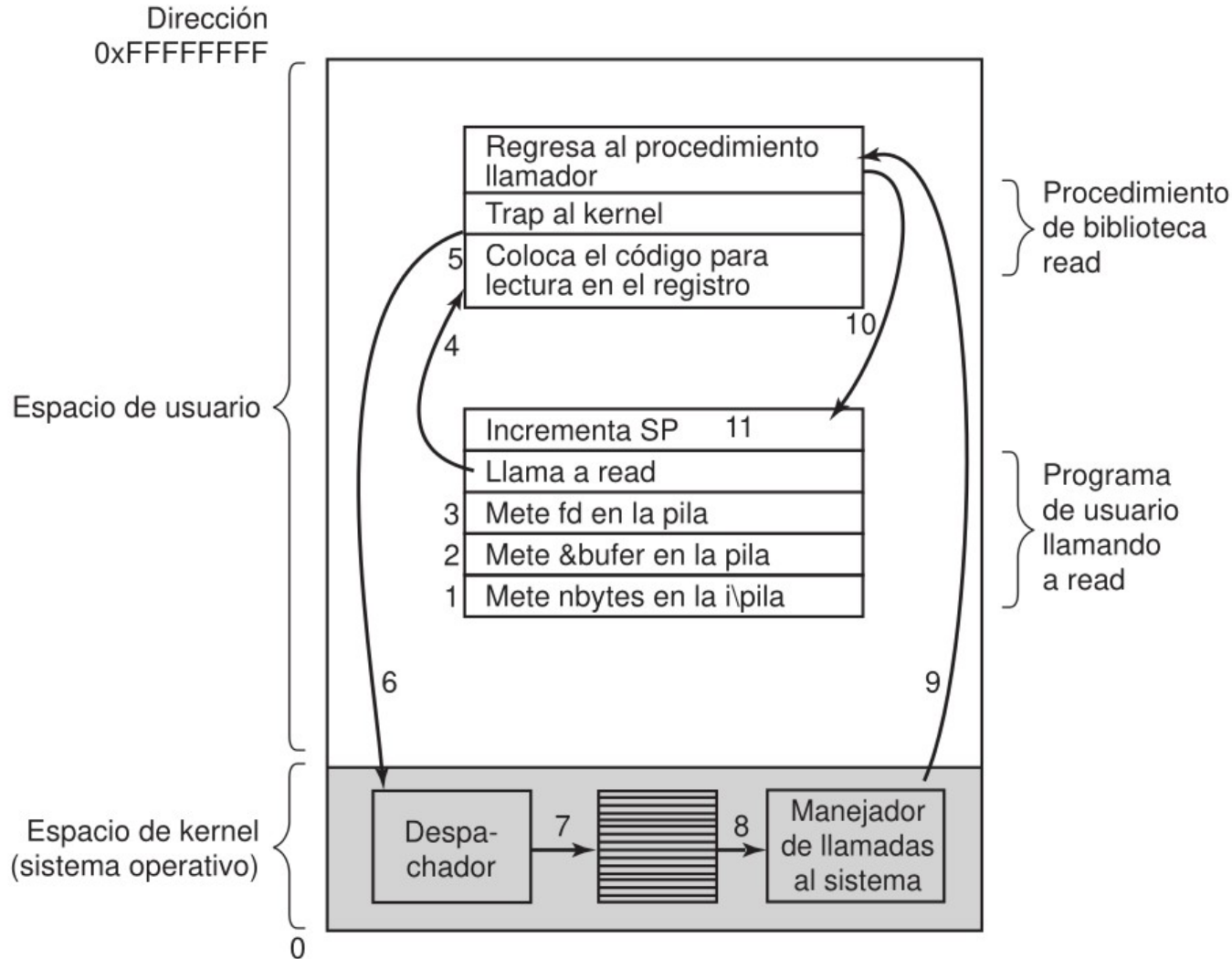
Llamada read

En C hacemos una llamada a un procedimiento de la biblioteca con el mismo nombre que la llamada al sistema: read. Una llamada desde un programa en C podría tener la siguiente apariencia:

```
cuenta=read(fd, bufer, nbytes);
```

La llamada al sistema (y el procedimiento de biblioteca) devuelve el número de bytes que se leen en cuenta. Por lo general este valor es el mismo que nbytes pero puede ser más pequeño si, por ejemplo, se encuentra el fin de archivo al estar leyendo.

Llamada read



Llamadas al sistema para la administración de procesos

fork es la única manera de crear un nuevo proceso en POSIX. Crea un duplicado exacto del proceso original, incluyendo todos los descriptores de archivos, registros y todo lo demás.

Después de fork, el proceso original y la copia (el padre y el hijo) se van por caminos separados. Todas las variables tienen valores idénticos al momento de la llamada a fork, pero como los datos del padre se copian para crear al hijo, los posteriores cambios en uno de ellos no afectarán al otro (el texto del programa, que no se puede modificar, se comparte entre el padre y el hijo).

La llamada a fork devuelve un valor, que es cero en el hijo e igual al identificador del proceso (PID) hijo en el padre. Mediante el uso del PID devuelto, los dos procesos pueden ver cuál es el proceso padre y cuál es el proceso hijo.

Llamadas al sistema para la administración de archivos

Para leer o escribir en un archivo, éste debe primero abrirse mediante `open`.

Esta llamada especifica el nombre del archivo que se va a abrir, ya sea como un nombre de ruta absoluto o relativo al directorio de trabajo, y un código de `O_RDONLY`, `O_WRONLY` o `O_RDWR`, que significa abrir para lectura, escritura o ambos. Para crear un nuevo archivo se utiliza el parámetro `O_CREAT`.

Después se puede utilizar el descriptor de archivo devuelto para leer o escribir. Al terminar, el archivo se puede cerrar mediante `close`, que hace que el descriptor de archivo esté disponible para reutilizarlo en una llamada a `open` posterior.

Llamadas al sistema para la administración de directorios

Veremos algunas llamadas al sistema que se relacionan más con los directorios o con el sistema de archivos como un todo, en vez de relacionarse sólo con un archivo específico.

Las primeras dos llamadas, mkdir y rmdir, crean y eliminan directorios vacíos, respectivamente.



Llamadas al sistema para la administración de directorios

link. Su propósito es permitir que aparezca el mismo archivo bajo dos o más nombres, a menudo en distintos directorios. Un uso común es para permitir que varios miembros del mismo equipo de programación compartan un archivo común, en donde cada uno de ellos puede ver el archivo en su propio directorio, posiblemente bajo distintos nombres.

Tener un archivo compartido significa que los cambios que realice cualquier miembro del equipo serán visibles de manera instantánea para los otros miembros; sólo hay un archivo.

/usr/ast		/usr/jim	
16	correo	31	bin
81	juegos	70	memo
40	prueba	59	f.c.
		38	prog1

/usr/ast		/usr/jim	
16	correo	31	bin
81	juegos	70	memo
40	prueba	59	f.c.
70	nota	38	prog1

Llamadas al sistema para la administración de directorios

La llamada al sistema **mount** permite combinar dos sistemas de archivos en uno.

Una situación común es hacer que el sistema de archivos raíz contenga las versiones binarias (ejecutables) de los comandos comunes y otros archivos de uso frecuente, en un disco duro.

Así, el usuario puede insertar un disco de CD-ROM con los archivos que se van a leer en la unidad de CD-ROM.



Estructura de un sistema operativo

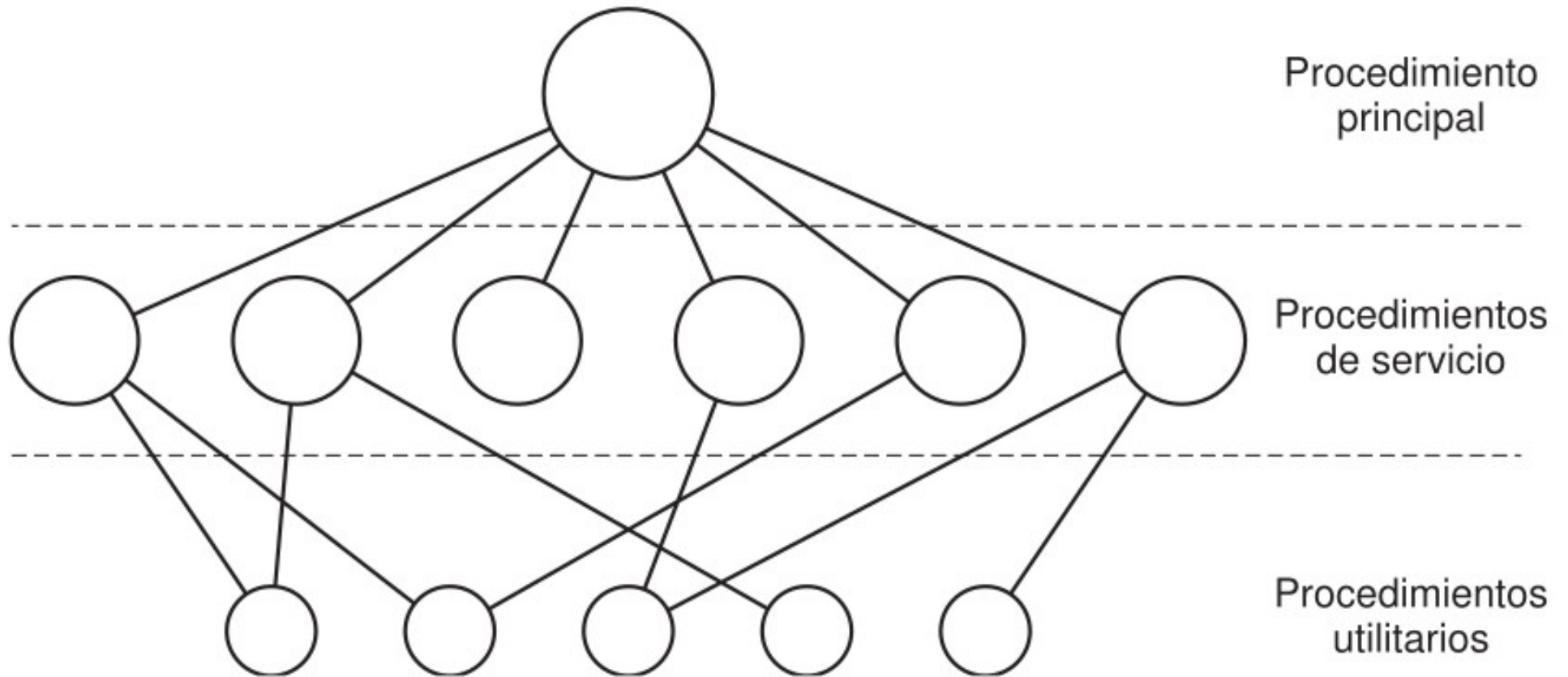


Sistemas monolíticos

En este diseño, que hasta ahora se considera como la organización más común, todo el sistema operativo se ejecuta como un solo programa en modo kernel. El sistema operativo se escribe como una colección de procedimientos, enlazados entre sí en un solo programa binario ejecutable extenso.

Cuando se utiliza esta técnica, cada procedimiento en el sistema tiene la libertad de llamar a cualquier otro, si éste proporciona cierto cómputo útil que el primero necesita. Al tener miles de procedimientos que se pueden llamar entre sí sin restricción, con frecuencia se produce un sistema poco manejable y difícil de comprender.

Sistemas monolíticos

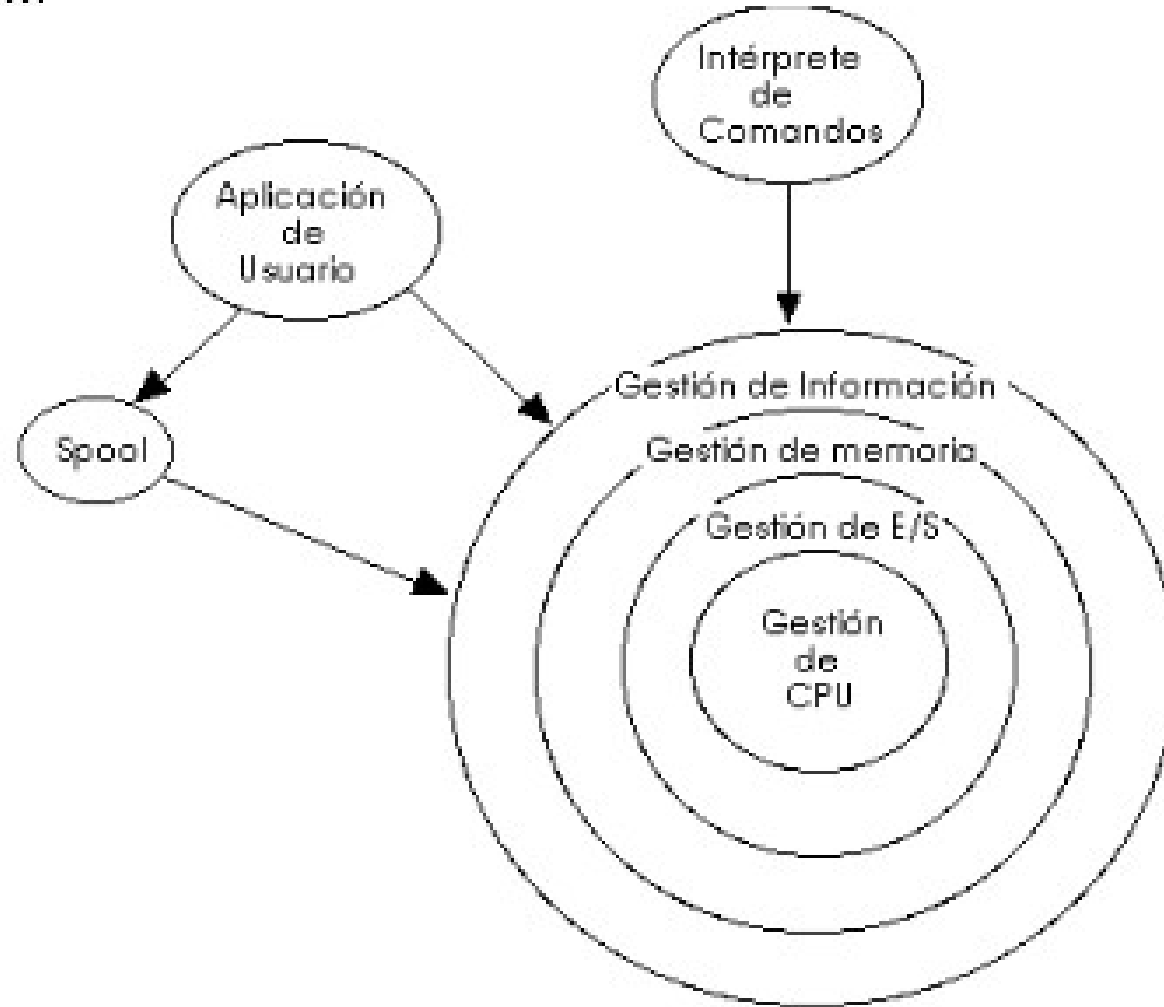


Sistemas de capas

Estos SO, se logran al organizar el sistema operativo como una jerarquía de capas, cada una construida encima de la que tiene abajo. El primer sistema construido de esta forma fue el sistema THE, construido en Technische Hogeschool Eindhoven en Holanda por E. W. Dijkstra (1968) y sus estudiantes. El sistema THE era un sistema simple de procesamiento por lotes para una computadora holandesa, la Electrologica X8, que tenía 32K de palabras de 27 bits (los bits eran costosos en aquel entonces).

Una mayor generalización del concepto de capas estaba presente en el sistema MULTICS. En vez de capa, MULTICS se describió como una serie de anillos concéntricos, en donde los interiores tenían más privilegios que los exteriores (que en efecto viene siendo lo mismo).

Sistemas de capas



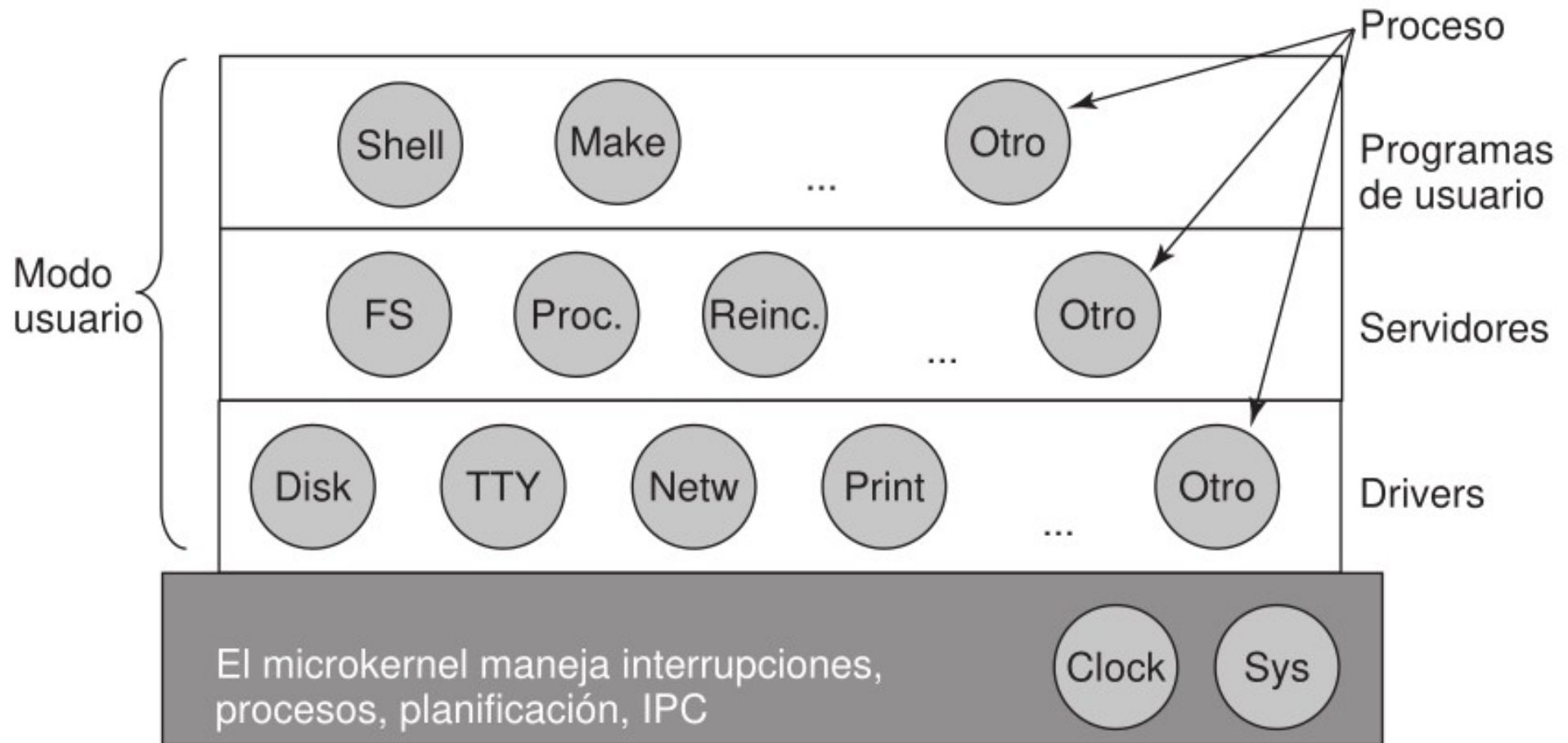
Microkernels

La idea básica detrás del diseño de microkernel es lograr una alta confiabilidad al dividir el sistema operativo en módulos pequeños y bien definidos, sólo uno de los cuales (el microkernel) se ejecuta en modo kernel y el resto se ejecuta como procesos de usuario ordinarios, sin poder relativamente.

En especial, al ejecutar cada driver de dispositivo y sistema de archivos como un proceso de usuario separado, un error en alguno de estos procesos puede hacer que falle ese componente, pero no puede hacer que falle todo el sistema. Así, un error en el driver del dispositivo de audio hará que el sonido sea confuso o se detenga, pero la computadora no fallará.

En contraste, en un sistema monolítico con todos los drivers en el kernel, un driver de audio con errores puede hacer fácilmente referencia a una dirección de memoria inválida y llevar a todo el sistema a un alto rotundo en un instante.

Estructura de MINIX 3



Modelo cliente-servidor

Una ligera variación de la idea del microkernel es diferenciar dos clases de procesos: los servidores, cada uno de los cuales proporciona cierto servicio, y los clientes, que utilizan estos servicios.

Este modelo se conoce como cliente-servidor. A menudo la capa inferior es un microkernel, pero eso no es requerido. La esencia es la presencia de procesos cliente y procesos servidor.

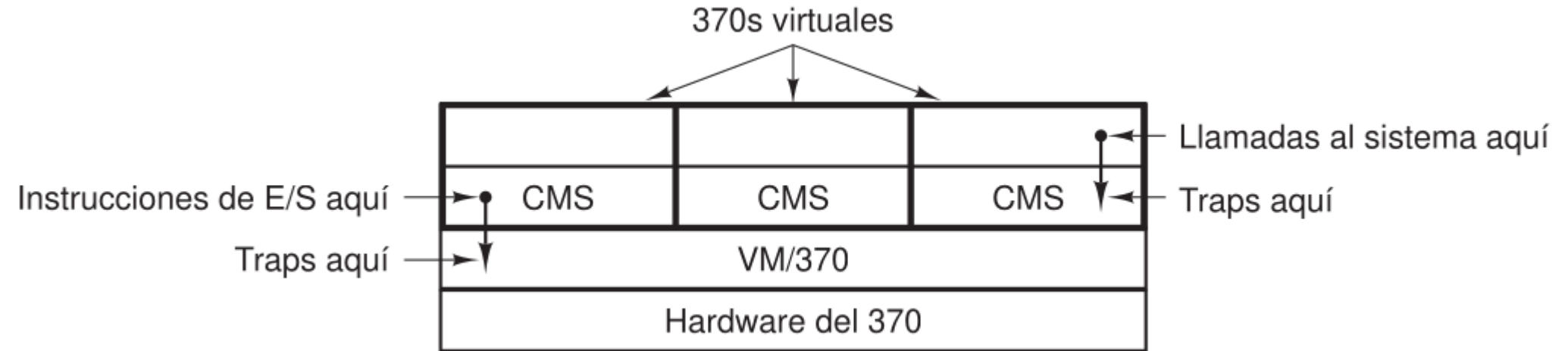
La comunicación entre clientes y servidores se lleva a cabo comúnmente mediante el paso de mensajes. Para obtener un servicio, un proceso cliente construye un mensaje indicando lo que desea y lo envía al servicio apropiado. Después el servicio hace el trabajo y envía de vuelta la respuesta. Si el cliente y el servidor se ejecutan en el mismo equipo se pueden hacer ciertas optimizaciones, pero en concepto estamos hablando sobre el paso de mensajes.

Máquinas virtuales

VM/370: Este sistema, que en un principio se llamó CP/CMS y posteriormente cambió su nombre a VM/370 (Seawright y MacKinnon, 1979), estaba basado en una astuta observación: un sistema de tiempo compartido proporciona (1) multiprogramación y (2) una máquina extendida con una interfaz más conveniente que el hardware por sí solo. La esencia de VM/370 es separar por completo estas dos funciones.

El corazón del sistema, que se conoce como monitor de máquina virtual, se ejecuta en el hardware solamente y realiza la multiprogramación, proporcionando no una, sino varias máquinas virtuales a la siguiente capa hacia arriba.

Máquinas virtuales



Exokernels

En vez de clonar la máquina actual, como se hace con las máquinas virtuales, otra estrategia es particionarla; en otras palabras, a cada usuario se le proporciona un subconjunto de los recursos.

Así, una máquina virtual podría obtener los bloques de disco del 0 al 1023, la siguiente podría obtener los bloques de disco del 1024 al 2047 y así sucesivamente.

Clasificación

Administración de tareas

Monotarea: Solamente permite ejecutar un proceso (aparte de los procesos del propio SO) en un momento dado. Una vez que empieza a ejecutar un proceso, continuará haciéndolo hasta su finalización y/o interrupción.

Multitarea: Es capaz de ejecutar varios procesos al mismo tiempo, habitualmente, mediante multiplexaje.

Administración de usuarios

Monousuario: Solo permite ejecutar los programas de un usuario al mismo tiempo.

Multiusuario: Permite que varios usuarios ejecuten simultáneamente sus programas, accediendo a la vez a los recursos de la computadora.

Clasificación

Administración de sesiones

Monosesión: Sistemas operativos que son capaces de ejecutar una sola sesión de usuario.

Multisesión: Sistemas operativos que son capaces de ejecutar varias sesiones de usuario a la vez.

Manejo de recursos

Centralizado: Permite usar los recursos de una sola computadora.

Distribuido: Permite utilizar los recursos (memoria, CPU, disco, periféricos...) de más de una computadora al mismo tiempo.

Bibliografía

- **Tanenbaum, A. S. (2009).** Sistemas operativos modernos (3a ed.) (pp. 1-18). México: Pearson Educación.
- **Stallings, W. (2005).** Sistemas operativos (5a ed.) (pp. 54-67). Madrid: Pearson Educación.

