Tema 2. Introducción a los sistemas operativos

(PARTE 1: transparencias 1 a 27)

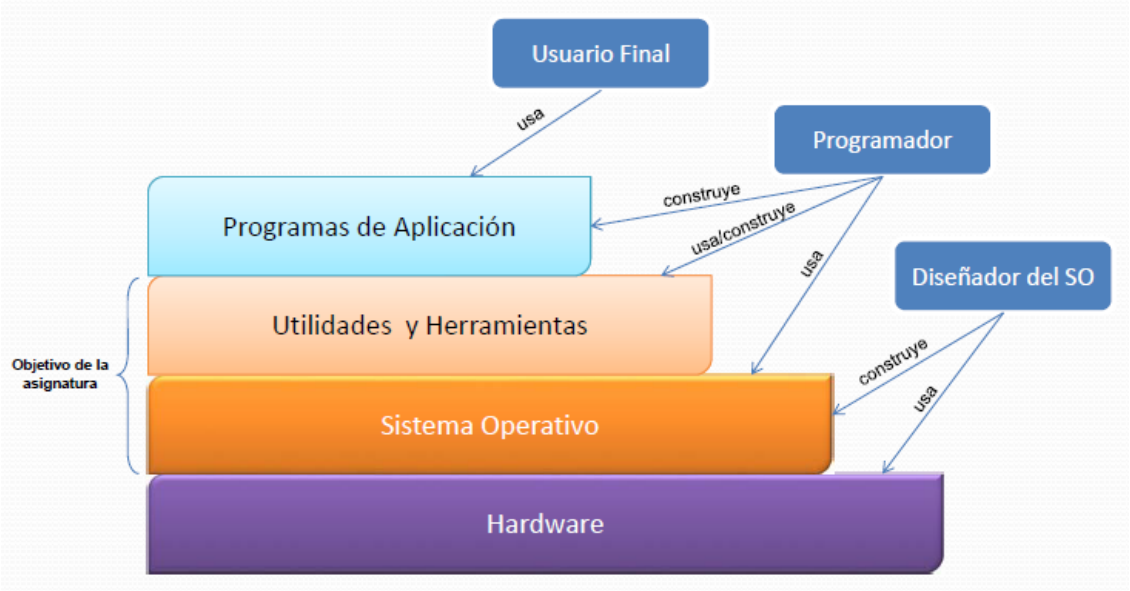
Introducción

En este tema nos vamos a centrar en el estudio de los sistemas operativos siguiendo los objetivos que tenéis en la primera transparencia del tema y que nos permitirán conocer:

- cuáles son sus principales tipos y cómo se implementa la multiprogramación,
- qué son los procesos y qué modelos se utilizan para realiza el control,
- qué son las hebras y cómo se modelan,
- cómo se gestiona la básica.

El sistema operativo facilita el uso del ordenador

En el Tema 1 vimos que el sistema operativo es el programa que controla la ejecución del resto de aplicaciones y programas y que es el interfaz entre las aplicaciones y que sirve de interfaz entre las aplicaciones y el hardware del computador, tal y como muestra la figura que tenéis en las transparencias del Tema 1:



El Sistema Operativo aparece como capa intermedia entre la capa de "utilidades y herramientas" y el hardware del ordenador:

- **los usuarios finales** no se suelen preocupar por los detalles del HW y ven el ordenador como un conjunto de aplicaciones.
- **los programadores** desarrollan los programas de aplicación utilizando utilidades y otros programas del sistema que le evitan tener que programar en código máquina (por ejemplo, los compiladores). Hay programadores que también se encargan de desarrollar estas utilidades. El Sistema Operativo oculta los detalles del hardware al programador y le sirve de interfaz al mismo a la hora de:
  - ayudarle a desarrollar programas proporcionándole editores, depuradores, y otras utilidades que se conocen como herramientas de desarrollo de programas de aplicación;
  - ejecutar programas cargando los datos e instrucciones en la memoria principal;

- acceder a los dispositivos de E/S de forma transparente a las instrucciones específicas que puedan utilizar cada uno de ellos;
- acceder a los ficheros de forma controlada por el conjunto de usuarios y dispositivos;
- acceder al sistema resolviendo posibles conflictos de acceso a los recursos,
- detectar los distintos tipos de errores durante la ejecución del sistema (errores de hardware, memoria, fallos de dispositivos, errores en los programas) y proporcionar una respuesta a los mismos;
- monitorizar el sistema y proporcionar estadísticas que pueden utilizarse para optimizar el sistema.

El sistema operativo facilita el uso del ordenador

El sistema operativo permite además que los recursos del sistema se puedan utilizar de forma eficiente. Para ello controla el transporte, almacenamiento y procesamiento de los datos funcionando como un programa (o conjunto de programas) más del ordenador, que controla el uso del resto de los recursos del sistema y la temporalización del resto de programas. Para ello, va solicitando y cediendo el control del procesador frecuentemente.

El Sistema operativo debe tener la capacidad de evolucionar

Los sistemas operativos se construyen de forma que se puedan desarrollar, escalar, probar e introducir mejoras sin que ello interfiera en su servicio. De esta forma, evoluciona en el tiempo mediante:

- las actualizaciones del software para el hardware existente y el soporte a nuevo hardware;
- la provisión de nuevos servicios para mejorar el rendimiento;
- la resolución de los fallos que se van detectando con el tiempo.

Como los cambios de sistemas operativos y de versiones suelen darse regularmente, es usual utilizar un desarrollo modular que facilite la capacidad de evolucionar.

Componentes de un sistema operativo multiprogramado

PROCESAMIENTO SERIE

En los primeros ordenadores (de finales de los años 40 y principios de los años 50, como el Manchester Baby que veis en la foto de abajo), el programador interactuaba directamente con el hardware del ordenador sin utilizar sistemas operativos: se cargaba el programa en código máquina a mano o utilizando tarjetas. Los errores debían examinarse también a mano viendo el contenido de los registros.

Los usuarios pedían cita a la hora de acceder al ordenador y reservaban un tiempo a la hora de utilizarlo, lo que implica las desventajas de:

- planificación (ineficacia en el uso de la CPU): si un usuario reserva una hora el ordenador y tarda menos, ese tiempo se perdía. Si no terminaba un usuario en su turno, debía de empezar de nuevo desde cero reservando otro turno
- tiempo de configuración muy alto a la hora de cargar el programa y volver a reiniciarlo si sucedía un error.

PROCESAMIENTO POR LOTES

A mediados de los años 50 se desarrollaron computadores que integraban un software denominador monitor. Este software agrupaba trabajos similares e ir cargando los diversos programas a la hora de reducir las dos desventajas principales del procesamiento serie.

Una parte del monitor está siempre cargada en memoria (monitor residente) y otras utilidades que se cargan como subrutinas en la parte de memoria de programas de usuario. El monitor lee un trabajo del lote, se carga en el área de memoria de programa de usuario, el procesador lo ejecuta hasta que hay una instrucción de finalización o error, el control vuelve de nuevo al monitor para que envíe los resultados y cargue el siguiente programa.

El principal problema que hace que el procesador esté ocioso radica en que los dispositivos de E/S son lentos con respecto al procesador. En la transparencia 3 podéis ver cómo son los tiempos de espera del procesador con monoprogramación y con un multiprogramador con tres programas en ejecución.

CONCEPTO DE MULTIPROGRAMACIÓN

En la multiprogramación o multitarea se trata de aprovechar el tiempo en que un trabajo está esperando a que termine una operación de E/S para que asignar el procesador a otro trabajo, albergando todos estos trabajos en la memoria.

En el ejemplo que tenéis en la transparencia 4 aparecen tres trabajos que tienen diferentes requerimientos de uso de la CPU y de los periféricos de E/S. En las transparencias 5 y 6 tenéis cómo se utilizarían los recursos del ordenador en el caso de mono o multiprogramación. En el caso de monoprogramación se requieren 30 minutos, con la infrautilización de los distintos dispositivos que podéis ver en la transparencia 5.

El sistema multiprogramado permite la ejecución de varios procesos de forma concurrente. El trabajo 1 se sigue ejecutando en los primeros 5 minutos, que se utilizan también para ejecutar

un tercio del trabajo 2 y la mitad del trabajo 3, de forma que pueden ejecutarse los tres procesos en 15 minutos con la optimización del uso de los recursos que podéis ver en la transparencia.

Gracias a esta filosofía pueden implementarse el tipo de interrupciones que vimos en el tema 1 para el mecanismo DMA (Direct Memory Address) de gestión de la E/S: mediante el uso de interrupciones el controlador del dispositivo puede encargarse de la gestión de la operación de E/S mientras el procesador realiza otro trabajo (hasta que el controlador del dispositivo active una interrupción para informarle que ha terminado).

Diseñar un sistema operativo multiprogramado requiere tener un algoritmo de planificación que decida el orden con el que se ejecutan los trabajos si varios listos para su ejecución

EJERCICIO 1
A partir de las definiciones que tenéis en la transparencia 7, responded a las preguntas de la transparencia 8.

Concepto de proceso

Se trata de un concepto fundamental dentro de los sistemas operativos, que se utiliza desde los 60 y que vamos a utilizar para referirnos a los trabajos de forma más general de acuerdo con el conjunto de definiciones que tenéis en la transparencia 9.

Estas definiciones están relacionadas con:

- la operación en lotes en sistemas multiprogramados (a partir de la idea original del uso simultáneo del procesador y los dispositivos de E/S, planificando el uso del procesador mediante las interrupciones)
- tiempo compartido (múltiples usuarios acceden de forma simultánea al sistema y el sistema operativo entrelaza la ejecución de los programas en pequeños intervalos de tiempo).
- sistemas de transacciones en tiempo real (usuarios realizando simultáneamente consultas a una base de datos).

El concepto de proceso está íntimamente ligado al de interrupción: los procesos suspenden su actividad a través de las interrupciones, se almacena su contexto actual (por ejemplo, el contador del programa actual y el contenido de otros registros), se ejecuta el código asociado a la interrupción y se continua a continuación con el proceso interrumpido u otro proceso diferente.

Esta forma de operar hace que puedan producirse errores ocasionados por:

- sincronizaciones inadecuadas (se interrumpe un proceso y luego no se reestablece de forma adecuada)
- violaciones de la exclusión mutua (hay recursos del sistema que solo pueden ser accedidos por uno de los procesos de forma simultánea para que no se produzcan errores)
- operaciones no deterministas de los programas (si hay programas que comparten memoria, pueden interferir y hacer que la sobreescritura haga que puedan comportarse de forma impredecible).

- Interbloqueos (que exista varios programas que se estén esperando entre si debido al uso que están haciendo de los recursos).

Para evitar estos errores, los procesos constan de:

- Un programa ejecutable,
- los datos necesarios que necesita el sistema operativo para ejecutarlo (incluyendo el contexto de ejecución). El conjunto de estos datos se enumera en la transparencia 10 (bloque de control de proceso) y lo estudiaremos con más detalle en el Tema 3.

La figura de la transparencia 11 muestra cómo se gestionan los procesos. En la figura se reflejan dos procesos (Proceso A y Proceso B) que se incluyen en una lista de procesos que almacena punteros de memoria a las direcciones de inicio de cada uno de ellos. De este modo, cada proceso es una estructura de datos que se tienen en cuenta a la hora de coordinar su ejecución. Como se puede ver en la figura, el registro contador de programa contiene la dirección de memoria donde se almacena la siguiente instrucción que se va a ejecutar del código del proceso B. En el registro índice de procesos se incluye el índice del proceso que se está ejecutando, el registro base contiene la dirección inicial del proceso y el registro límite su tamaño. Si previamente se estaba ejecutando el proceso A y no se concluyó, el contenido de los registros se almacenó en su contexto. La ejecución de este proceso continuará cuando se cargue en el PC una dirección de memoria asignada a una instrucción en su área de programa.

La transparencia 12 define otro concepto importe: el de traza de una ejecución. La traza de un proceso es la secuencia de instrucciones que se ejecutan para dicho proceso. El comportamiento del procesador puede caracterizarse mostrando la forma en que se intercalan las trazas de varios procesos. El despachador intercambia el procesador entre un proceso y otro.

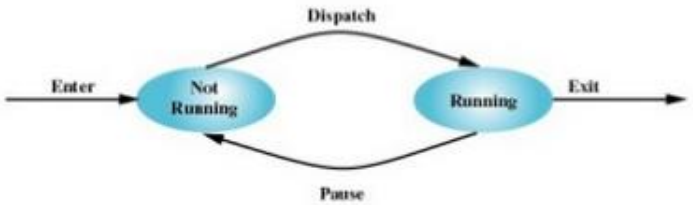
EJERCICIO 2
Responda a la pregunta al final de la transparencia 12

En el ejemplo de la transparencia 14 se asume que el sistema operativo solo permite que un proceso continúe durante 6 ciclos de instrucción, luego se le interrumpe y se ejecuta el código del dispatcher.

La transparencia 15 muestra cómo se comunican los procesos con el sistema operativo a través de distintos tipos de peticiones de servicio que se implementan mediante interrupciones software (trap).

Modelo de dos estados de los procesos

Se asume que hay dos estados para los procesos (en ejecución o no). El dispatcher hace que el estado de los procesos pueda cambiar de uno a otro. El sistema operativo controla el entrelazado de los procesos y la asignación de recursos a cada uno de ellos. El sistema operativo genera los procesos inicialmente en estado de no ejecución:



(a) State transition diagram

Modelo de cinco estados de los procesos

El modelo de dos estados es demasiado simple, en el modelo de cinco estados se contempla que los estados que no están en ejecución puedan estar preparados para ejecutarse, estén bloqueados (porque están esperando que termine una operación de E/S), sea un proceso nuevo que todavía no ha sido aceptado para ejecución, o un proceso terminado (ya se han extraído del conjunto de procesos aceptados para ejecución).

En las transparencias 18 y 19 tenéis representados los cinco estados y los eventos que tienen lugar para transitar entre cada uno de ellos. Las transparencias 20 a 26 describen cómo se realiza el control de los procesos:

- información que se utiliza para describir a los procesos (PCB),
- cómo se inicializa esta información cuando se crea el proceso,
- los modos de ejecución del procesador (usuario y kernel). Se representan también en la transparencia 16 e indican si el programa n ejecución tiene acceso a todos los recursos del sistema (normalmente solo el sistema operativo) o no (programas de usuario),
- cómo se cambia de un modo de ejecución a otro