



Introducción a los Sistemas Operativos

Arquitectura y Sistemas Operativos

Unidad 1



Sistema informático

- Hardware
 - Procesador y memoria
 - Dispositivos de E/S: almacenamiento, red, HCI, impresión, etc.
- Software
 - Software de sistema: compilador, GUI, shell, etc.
 - Aplicaciones
- Personas
 - Usuarios en general
 - Administradores del sistema
 - Desarrolladores / programadores



Computadora – Elementos básicos

- Al más alto nivel, una computadora consta del procesador, la memoria y los componentes de E/S, incluyendo uno o más módulos de cada tipo. Estos componentes se interconectan de manera que se pueda lograr la función principal de la computadora, que es ejecutar programas. Por tanto, hay cuatro elementos estructurales principales:
 - **Procesador:** Controla el funcionamiento del computador y realiza sus funciones de procesamiento de datos. Cuando sólo hay un procesador, se denomina usualmente unidad central de proceso (*Central Processing Unit, CPU*).
 - **Memoria principal:** Almacena datos y programas. Esta memoria es habitualmente volátil; es decir, cuando se apaga el computador, se pierde su contenido. En contraste, el contenido de la memoria del disco se mantiene incluso cuando se apaga el computador. A la memoria principal se le denomina también *memoria real o memoria primaria*.
 - **Módulos de E/S:** Transfieren los datos entre el computador y su entorno externo. El entorno externo está formado por diversos dispositivos, incluyendo dispositivos de memoria secundaria (por ejemplo, discos), equipos de comunicaciones y terminales.
 - **Bus del sistema:** Proporciona comunicación entre los procesadores, la memoria principal y los módulos de E/S.



Componentes de una computadora

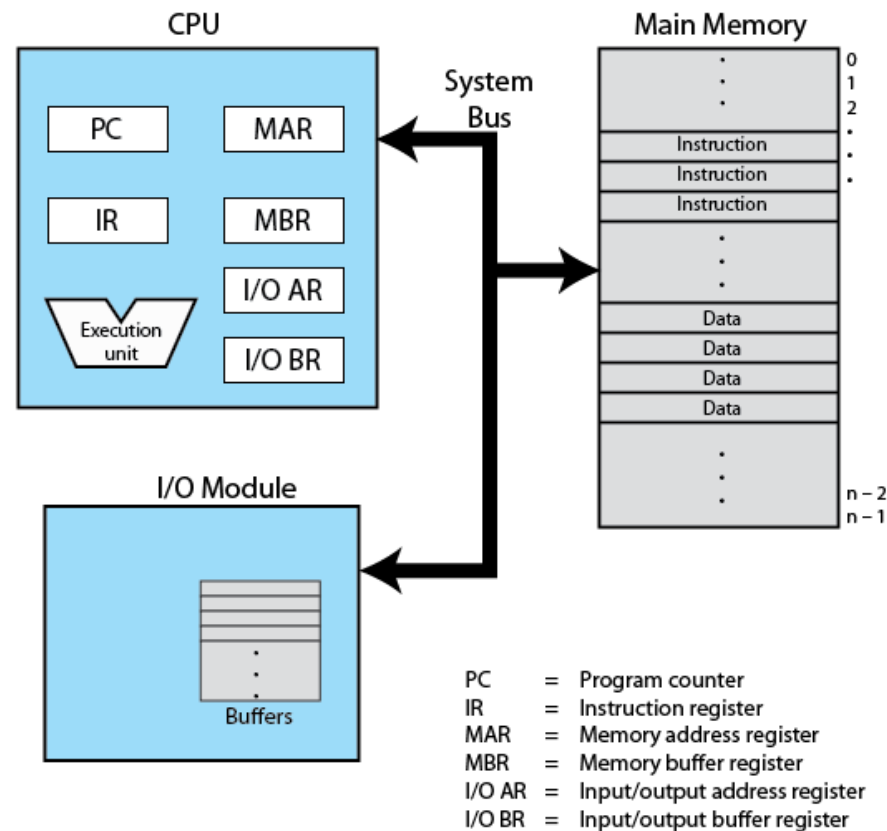
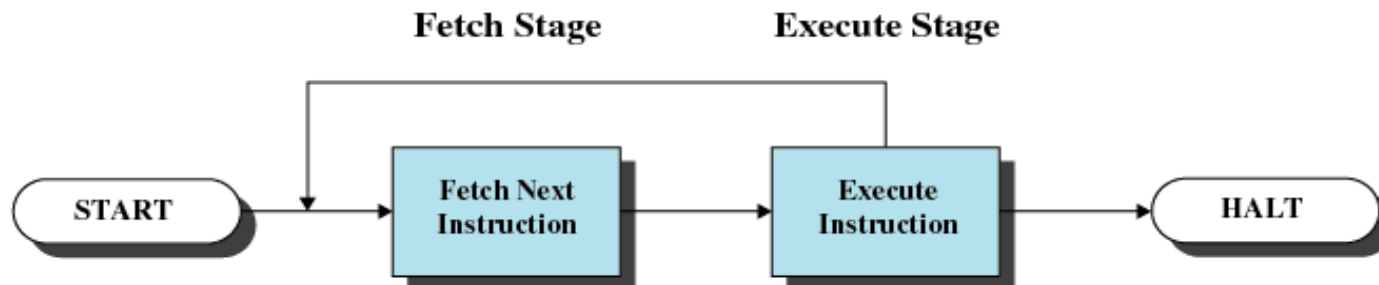


Figure 1.1 Computer Components: Top-Level View



Ejecución de instrucciones

- Un programa que se ejecuta en un procesador consta de un conjunto de instrucciones almacenado en memoria. En su forma más simple, el procesamiento de una instrucción consta de dos pasos:
 - El procesador lee (*busca*) *instrucciones de la memoria*, una cada vez, y ejecuta cada una de ellas. La ejecución del programa consiste en repetir el proceso de búsqueda y ejecución de instrucciones.
 - La ejecución de la instrucción puede involucrar varias operaciones dependiendo de la naturaleza de la misma.
- Se denomina *ciclo de instrucción al procesamiento requerido por una única instrucción*.
- *Fase de búsqueda y fase de ejecución*. La ejecución del programa se detiene sólo si se apaga la máquina, se produce algún tipo de error irreparable o se ejecuta una instrucción del programa que detiene el procesador.





Interrupciones

- Prácticamente todas las computadoras proporcionan un mecanismo por el cual otros módulos (memoria y E/S) pueden interrumpir la secuencia normal del procesador
- Básicamente, las interrupciones constituyen una manera de mejorar la utilización del procesador
- Una interrupción suspende la secuencia normal de ejecución
 - Cuando se completa el procesamiento de la interrupción, se reanuda la ejecución
 - El programa de usuario no tiene que contener ningún código especial para tratar las interrupciones; el procesador y el sistema operativo son responsables de suspender el programa de usuario y, posteriormente, reanudarlo en el mismo punto



Interrupciones

- Suspensión de la secuencia normal de ejecución

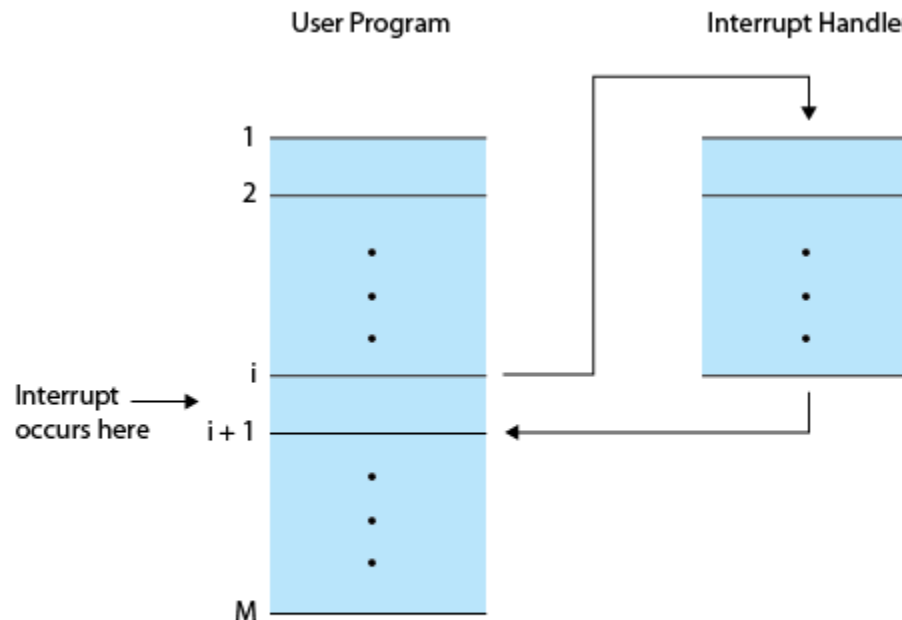
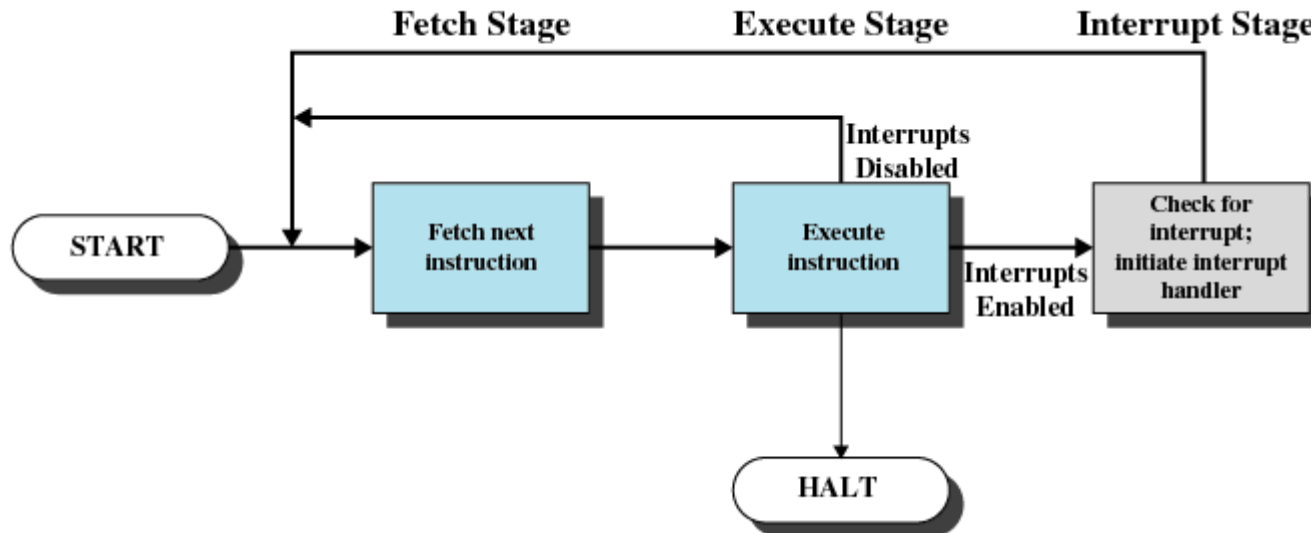


Figure 1.6 Transfer of Control via Interrupts



Ciclo de instrucción con interrupciones

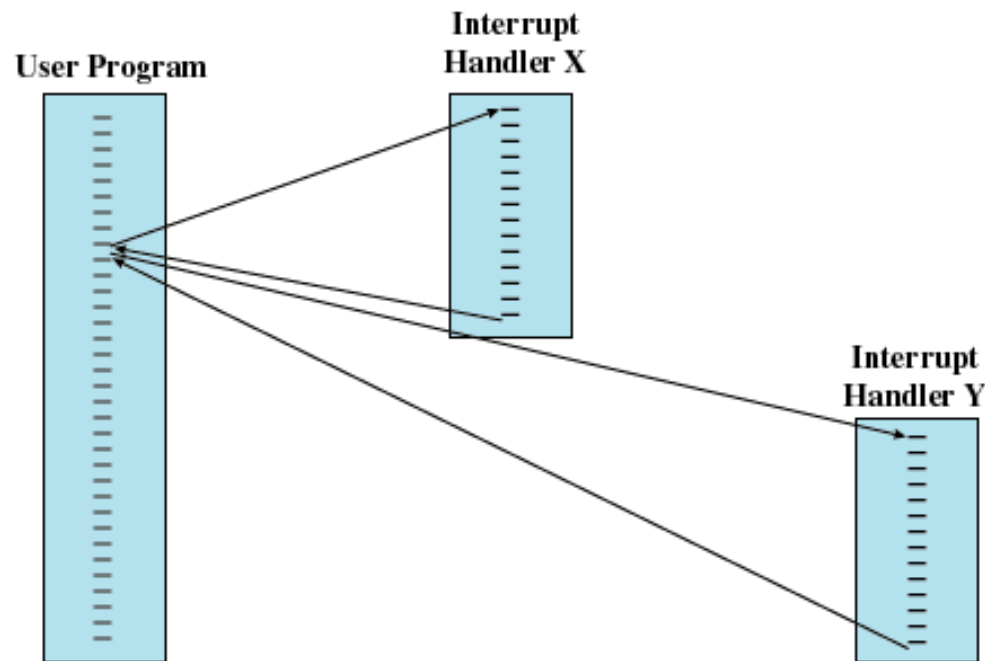
- El procesador comprueba si hay interrupciones
- Si no hay ninguna solicitud de interrupción, toma la siguiente instrucción para el programa actual
- Si una interrupción está pendiente, se suspende la ejecución del programa actual y se ejecuta la rutina del manejador de interrupciones





Interrupciones múltiples

- Las interrupciones se deshabilitan mientras se está procesando un pedido de interrupción

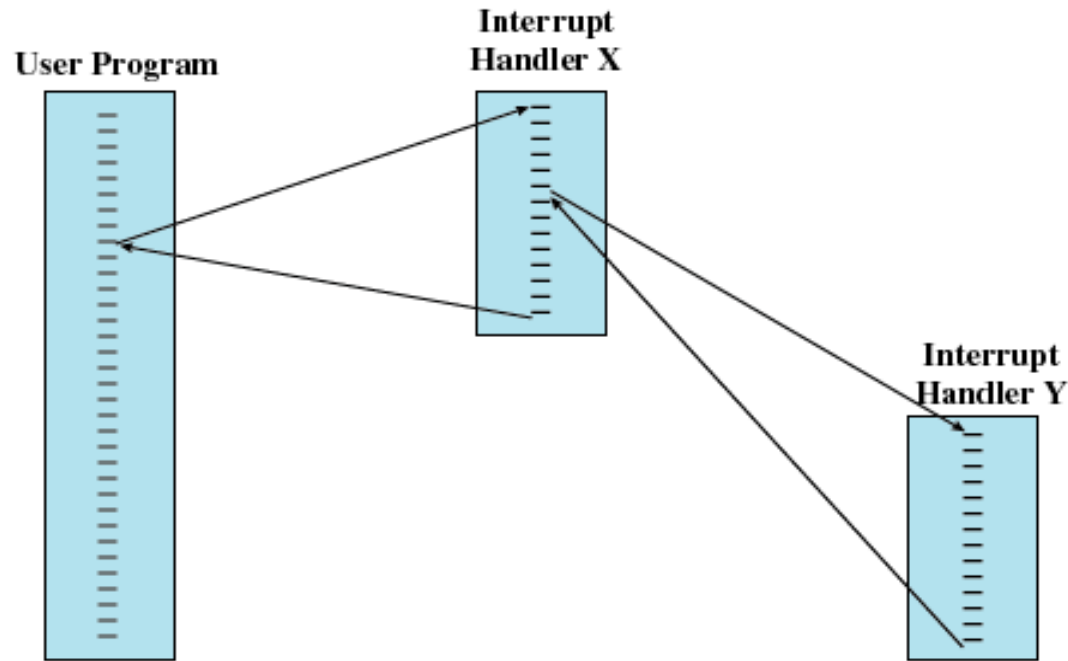


(a) Sequential interrupt processing



Interrupciones múltiples

- Se definen prioridades para la atención de los pedidos de interrupciones

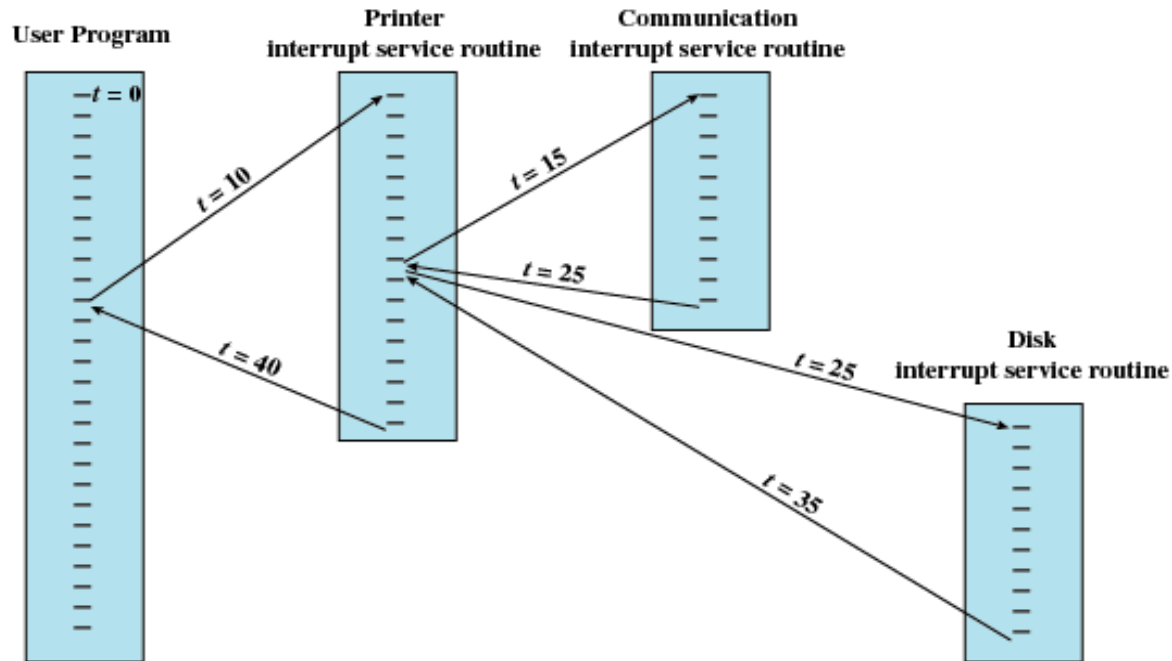


(b) Nested interrupt processing



Interrupciones múltiples

- Secuencia temporal con múltiples interrupciones anidadas





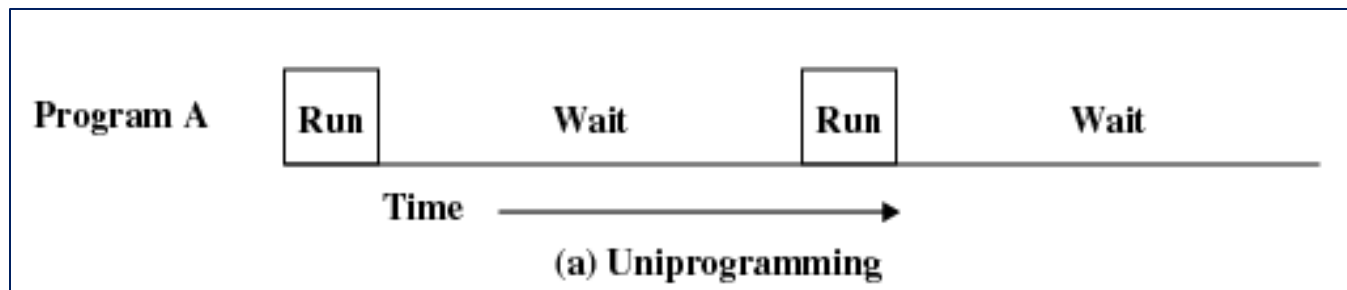
Clases de interrupciones

De programa	Generada por alguna condición que se produce como resultado de la ejecución de una instrucción, tales como un desborde aritmético, una división por cero, un intento de ejecutar una instrucción de máquina ilegal, y las referencias fuera del espacio de la memoria permitido para un usuario.
Por temporizador	Generada por un temporizador del procesador. Permite al sistema operativo realizar ciertas funciones de forma regular.
De E/S	Generada por un controlador de E/S para señalar la conclusión normal de una operación o para indicar diversas condiciones de error.
Por fallo del hardware	Generada por un fallo, como un fallo en el suministro de energía o un error de paridad en la memoria.



Mono programación

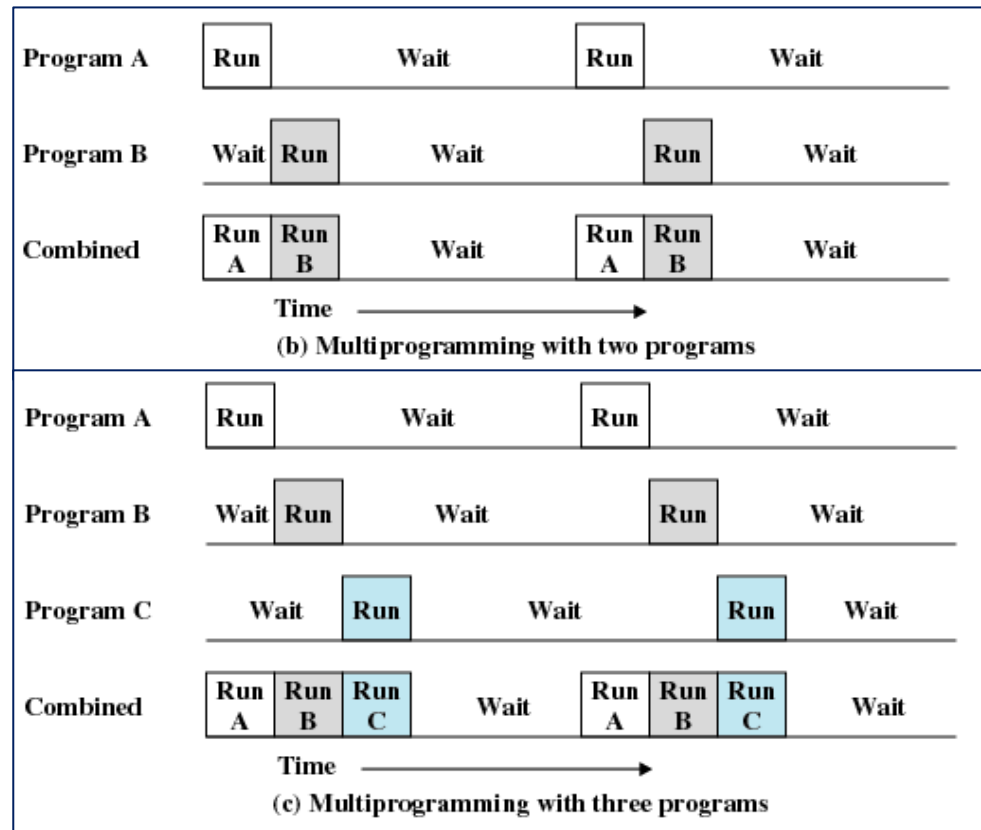
- El procesador debe esperar a que se complete la instrucción de E/S antes de continuar





Multiprogramación

- Cuando una tarea necesita esperar por la E/S, el procesador puede cambiar al otra tarea





Multiprogramación

- Incluso utilizando interrupciones, puede suceder que el procesador siga sin utilizarse eficientemente
 - Si el tiempo requerido para completar una operación de E/S es mucho mayor que el código de usuario entre las llamadas de E/S (una situación habitual), el procesador estará parado la mayor parte del tiempo
- Una solución a este problema es permitir que múltiples programas de usuario estén activos al mismo tiempo



Multiprogramación

- También llamado multitarea (multitasking)
- Cuando un proceso se bloquea al esperar por la E/S, se puede ejecutar en la CPU instrucciones de otro proceso
- Los procesos entrelazan su ejecución: concurrencia
- La CPU y la E/S trabajan a la misma vez
 - Se terminan más trabajos en menos tiempo



¿Qué es un Sistema Operativo?

- Un sistema operativo es un programa que administra el hardware de una computadora
- Interactúa entre el/los usuario/s y el hardware
- Un sistema informático tiene muchos recursos que pueden ser útiles para solucionar un problema: tiempo de CPU, espacio de memoria, almacenamiento de archivos, dispositivos de E/S, etc.
 - El sistema operativo actúa como intermediario entre estos recursos



Objetivos y funciones de los SO

- Un sistema operativo es un programa que controla la ejecución de aplicaciones y programas y que actúa como interfaz entre las aplicaciones y el hardware de la computadora. Un sistema operativo tiene los siguientes tres objetivos:
 - Facilidad de uso
 - Un sistema operativo facilita el uso de una computadora
 - Eficiencia
 - Un sistema operativo permite que los recursos de un sistema de computación se puedan utilizar de una manera eficiente.
 - Capacidad para evolucionar
 - Un sistema operativo se debe construir de tal forma que se puedan desarrollar, probar e introducir nuevas funciones en el sistema sin interferir con su servicio.

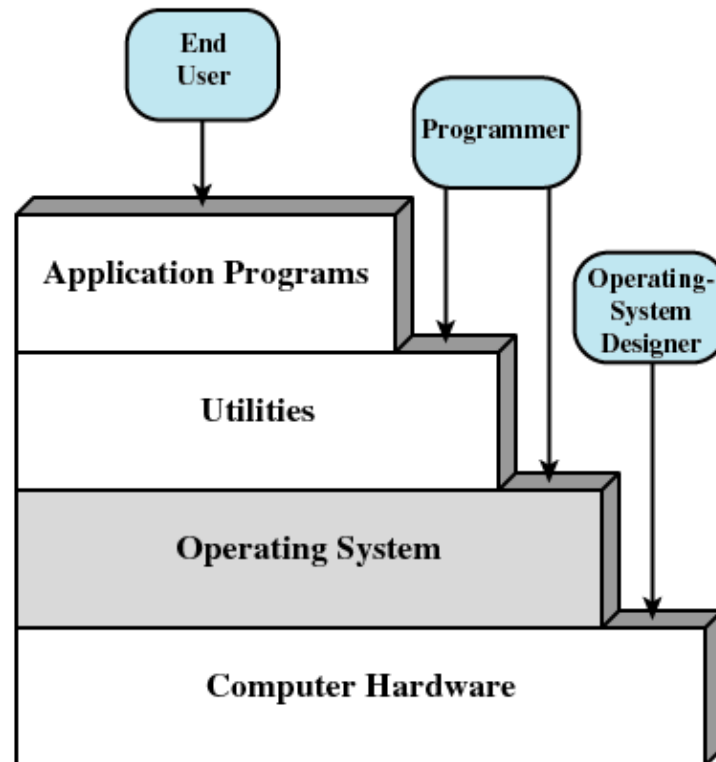


Objetivos y funciones de los SO

- El sistema operativo es el único programa que interactúa directamente con el hardware de la computadora. Sus funciones primarias son:
 - **Abstracción:** En los programas no preocupan los detalles de acceso al hardware, o de la configuración de la computadora. El SO se encarga de proporcionar una serie de abstracciones para que los programadores puedan enfocarse en resolver las necesidades particulares de los usuarios.
 - **Administración de recursos:** Una computadora tiene a su disposición una gran cantidad de recursos (memoria, espacio de almacenamiento, tiempo de procesamiento, etc.). Los diferentes procesos que se ejecuten compiten por sus recursos. Al gestionar la asignación de recursos, el SO implementa políticas que los asigne de forma efectiva y acorde a las necesidades establecidas.
 - **Aislamiento:** En un sistema multiusuario y multitarea cada proceso y cada usuario no tendrá que preocuparse por otros que estén usando el mismo sistema. Idealmente, su experiencia será la misma que si el sistema estuviera exclusivamente dedicado a su atención. Para implementar correctamente las funciones de aislamiento, el SO utiliza hardware específico para dicha protección.



Capas de un sistema de computación





SO – Elementos básicos I

- **Drivers:** son programas que forman parte del sistema operativo y manejan los detalles de bajo nivel relacionados con la operación de los distintos dispositivos periféricos.
- **Núcleo o Kernel:** es el SO propiamente dicho, su parte central. Se encarga de las tareas fundamentales y contiene diversos subsistemas.
- **Interprete de comandos (shell):** un programa mas, que muchas veces es ejecutado automáticamente cuando comienza el SO, que le permite al usuario interactuar con el SO. Puede ser gráfico o de línea de comandos.
- **Proceso:** un programa en ejecución mas su espacio de memoria asociado y otros atributos.



SO – Elementos básicos II

- **Archivo:** secuencia de bits con un nombre y una serie de atributos que indican permisos.
- **Directorio:** colección de archivos y directorios que contiene un nombre y se organiza jerárquicamente.
- **Dispositivo virtual:** una abstracción de un dispositivo físico bajo la forma, en general, de un archivo, de manera tal que se pueda abrir, leer, escribir, etc.
- **Sistema de archivos:** es la forma de organizar los datos en el disco para gestionar su acceso, permisos, etc.



SO – Elementos básicos III

- **Directorios del sistema:** son directorios donde el propio SO guarda archivos que necesita para su funcionamiento, por ejemplo, /boot, /devices o C:\Windows\system32.
- **Binario del sistema:** son archivos, que viven en los directorios del sistema. Si bien no forman parte del kernel, suelen llevar a cabo tareas muy importantes o proveer las utilidades básicas del sistema. Ejemplos:
 - /usr/sbin/syslogd: es el encargado de guardar los eventos del sistema en un archivo.
 - /usr/bin/who: indica que usuarios están en sesión en el sistema.
- **Archivo de configuración:** es un archivo mas, excepto porque el SO saca de este información que necesita para funcionar. Por ejemplo, /etc/passwd o C:\Windows\system32\user.dat



SO – Elementos básicos IV

- **Usuario:** la representación, dentro del propio SO, de las personas o entidades que pueden usarlo. Principalmente sirve como una forma de aislar información entre estos y establecer limitaciones y reglas.
- **Grupo:** un conjunto de usuarios.



SO - Interfaces

- ¿Para quién es la interfaz?
 - Usuarios en general: entorno de ejecución
 - Administradores: entorno de administración
 - Desarrolladores: interfaz de programación
- ¿Qué aspecto tiene la interfaz?
 - Texto (CLI: Command Line Interface)
 - Gráfica (GUI: Graphical User Interface)
 - Servicios de programación (API)



CLI

- Incorpora un lenguaje sencillo para dar instrucciones al SO:
 - Cargar programas, trabajar con archivos, etc.
 - Se le llama shell, intérprete de órdenes, consola...

```
MINGW32:/c/Users

Gustavo@Gustavo-nbook MINGW32 /c/Users
$

Gustavo@Gustavo-nbook MINGW32 /c/Users
$ ls
'All Users'@ 'Default User'@ DefaultAppPool/ Gustavo/
Default/     Default.migrated/ desktop.ini Public/

Gustavo@Gustavo-nbook MINGW32 /c/Users
$ ll
total 33
lrwxrwxrwx 1 Gustavo 197121 14 jul. 16 2016 'All Users' -> /c/ProgramData/
drwxr-xr-x 1 Gustavo 197121 0 sep. 17 2016 Default/
lrwxrwxrwx 1 Gustavo 197121 16 jul. 16 2016 'Default User' -> /c/Users/Default
/
drwxr-xr-x 1 Gustavo 197121 0 sep. 17 2016 Default.migrated/
drwxr-xr-x 1 Gustavo 197121 0 nov. 2 2016 DefaultAppPool/
-rw-r--r-- 1 Gustavo 197121 174 jul. 16 2016 desktop.ini
drwxr-xr-x 1 Gustavo 197121 0 may. 25 20:35 Gustavo/
drwxr-xr-x 1 Gustavo 197121 0 sep. 17 2016 Public/

Gustavo@Gustavo-nbook MINGW32 /c/Users
$ |
```



GUI





API del SO – Llamadas al sistema

- API: Application Programming Interface
- El SO ofrece a los desarrolladores y a los procesos un conjunto de servicios públicos, accesibles mediante una API: llamadas al sistema (system calls)
- Llamada `write()` de UNIX. Escribe un bloque de datos en un archivo o en un dispositivo de E/S.

```
int write(int fd, void* buffer, size_t size);
```



SO – Servicios proporcionados

- **Desarrollo de programas:** El sistema operativo proporciona una variedad de utilidades y servicios, tales como editores y depuradores, para asistir al programador en la creación de los programas.
- **Ejecución de programas:** Las instrucciones y los datos se deben cargar en memoria principal. Los dispositivos de E/S y archivos se deben inicializar, y otros recursos deben prepararse. Los sistemas operativos realizan estas tareas de planificación en nombre del usuario.
- **Acceso a dispositivos de E/S:** Cada dispositivo de E/S requiere su propio conjunto de instrucciones o señales de control para cada operación. El SO proporciona una interfaz uniforme que esconde esos detalles de forma que los programadores puedan acceder a dichos dispositivos utilizando lecturas y escrituras sencillas.
- **Acceso controlado a los archivos:** Para acceder a los archivos, el SO debe conocer la naturaleza del dispositivo de E/S (disco, cinta), y la estructura de los datos contenidos en los archivos. En un sistema con múltiples usuarios, el SO proporciona mecanismos de protección para controlar el acceso a los archivos.
- **Acceso al sistema:** Protección a recursos y datos. Evita el uso de usuarios no autorizados y resuelve conflictos en el uso de recursos.
- **Detección y respuesta a errores:** Durante la ejecución de un programa pueden ocurrir errores. Errores de hardware internos y externos: error de memoria, fallo en un dispositivo; y errores software: división por cero, el intento de acceso a una posición de memoria prohibida o la incapacidad del SO para conceder la solicitud de una aplicación). El SO debe proporcionar una respuesta que elimine la condición de error, con el menor impacto en las aplicaciones que están en ejecución.
- **Auditoría:** Un buen sistema operativo recoge estadísticas de uso de los diferentes recursos y supervisa parámetros de rendimiento tales como el tiempo de respuesta. Esta información es útil para anticipar las necesidades de mejoras y optimizar el sistema a fin de mejorar su rendimiento.



SO como administrador de recursos

- Procesos y recursos
 - Proceso: programa en ejecución
 - Recurso: dispositivo físico o virtual que necesita un proceso para ejecutarse
- Los recursos son escasos
- Los procesos compiten por ellos
- El SO actúa como árbitro/mediador, que asigna recursos de forma justa y eficiente



SO como administrador de recursos

- Una computadora es un conjunto de recursos que se utilizan para el transporte, almacenamiento y procesamiento de los datos, así como para llevar a cabo el control de estas funciones. El sistema operativo se encarga de gestionar estos recursos
- Gestionando los recursos de la computadora, el sistema operativo tiene el control de las funciones básicas del mismo
- El SO debe determinar a quién se le entregan los recursos, qué cantidad, en qué momento y por cuánto tiempo
 - Política de gestión de recursos



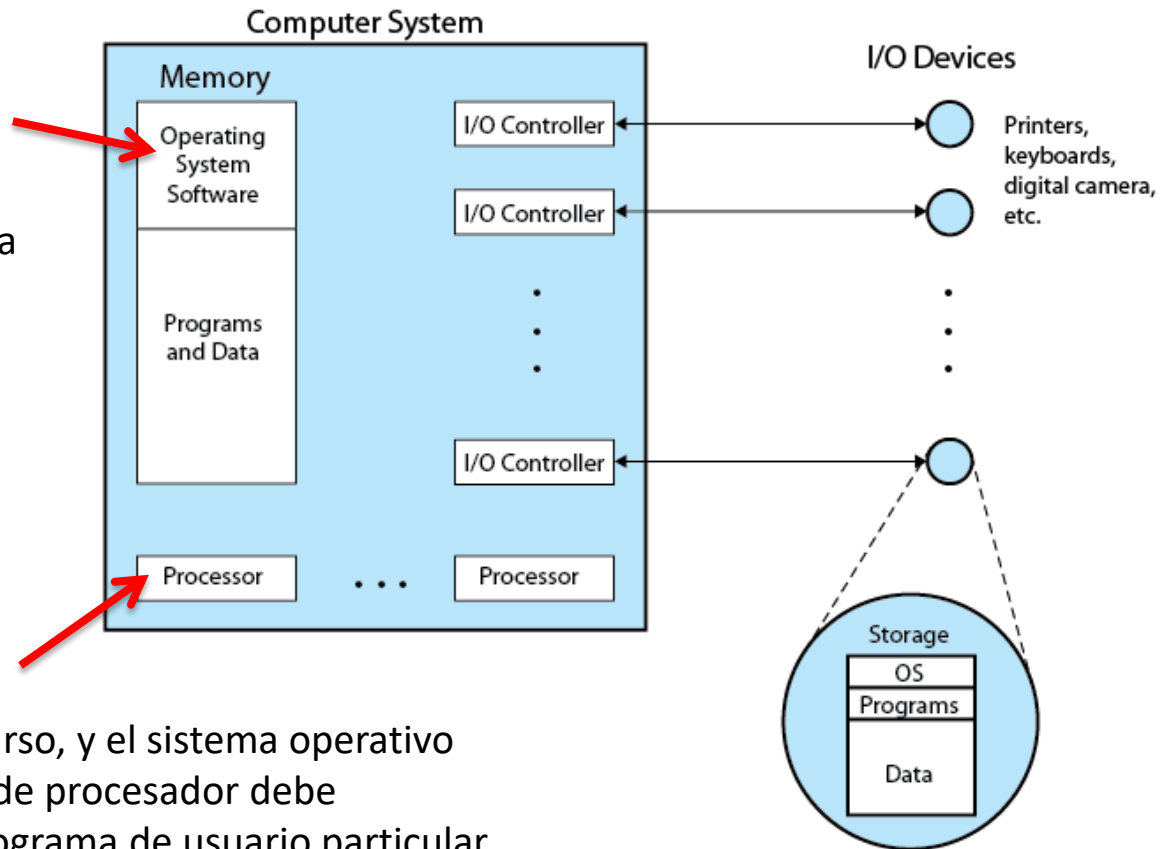
SO como administrador de recursos

- Criterios que deben cumplir las políticas del SO
 - Optimizar el rendimiento del sistema
 - Reparto equitativo para evitar acaparamientos e inanición de los procesos perjudicados
 - Garantizar la seguridad del sistema (confidencialidad, integridad, disponibilidad)
- Estos criterios entran en conflicto
 - Ejemplo: no se puede dar el máximo rendimiento y al mismo tiempo dar un reparto justo



SO como administrador de recursos

Una porción del SO se encuentra en la memoria principal. Esto incluye el **kernel (núcleo)** que contiene las funciones del sistema operativo más frecuentemente utilizadas y, en cierto momento, otras porciones del sistema operativo actualmente en uso.



El procesador es también un recurso, y el sistema operativo debe determinar cuánto tiempo de procesador debe asignarse a la ejecución de un programa de usuario particular.



SO como administrador de procesos

- Un proceso es una unidad de trabajo en un sistema
- Cada sistema tiene un conjunto de procesos, algunos de ellos del SO (ejecutan código del sistema) y el resto procesos de usuario.
- Todos estos procesos pueden ser ejecutados de forma concurrente, compartiendo en el tiempo la CPU (si se dispone solo de una)
- El SO es responsable de las siguientes actividades respecto a la administración de procesos
 - Crear y borrar los procesos de usuarios y del sistema
 - Suspende y reanuda procesos
 - Proporcionar mecanismos para la sincronización de procesos
 - Proporcionar mecanismos para la comunicación de procesos
 - Proporcionar mecanismos para el tratamiento de interbloqueos



SO como administrador de memoria

- Para que un programa pueda ejecutarse, debe estar asignado a direcciones absolutas y cargados en memoria principal (MP)
- Mientras se está ejecutando, accede a las instrucciones y datos de la memoria
- Cuando el programa termina, su espacio de memoria queda disponible para que otro programa pueda ser cargado y ejecutado
- El SO es responsable de las siguientes actividades para la gestión de memoria
 - Controlar que partes de la MP se está usando y quien lo está haciendo
 - Decidir que datos y procesos agregar o eliminar de la MP
 - Asignar y liberar la asignación del espacio de memoria según sea necesario



Paginación y memoria virtual

- Paginación
 - División de un proceso en “páginas” que se pueden colocar en zonas diferentes de la memoria principal
- Memoria virtual
 - El programa no necesita estar cargado totalmente en memoria
 - El almacenamiento secundario se usa como extensión de la memoria principal
- Se resuelve automáticamente, sin que el usuario ni el programador tengan que intervenir



SO como administrador de archivos

- La administración del sistema de archivos es uno de los componentes mas visibles de un sistema operativo
- El SO implementa el concepto abstracto de archivo, gestionando los medios de almacenamiento masivo y los dispositivos que los controlan
- Cuando varios usuarios tienen acceso a los archivos, es deseable controlar quien y en que forma (r-w-x) accede a los archivos
- El SO es responsable de las siguientes actividades referidas a la administración de archivos
 - Creación y borrado de archivos
 - Creación y borrado de directorios para organizar archivos
 - Soporte de primitivas para manipular archivos y directorios
 - Asignación de archivos a los dispositivos de almacenamiento secundario
 - Copias de seguridad



SO – Facilidades para la evolución

- Un sistema operativo importante debe evolucionar en el tiempo por las siguientes razones:
 - Actualizaciones de hardware y aparición de nuevos tipos de hardware
 - Nuevos servicios
 - Resolución de fallos



Protección de la información y seguridad

- Protección es cualquier mecanismo que controle el acceso a los procesos y recursos
 - Disponibilidad
 - Proteger el sistema contra la interrupción
 - Confidencialidad
 - Asegurar que los usuarios no pueden acceder a datos para cuales el acceso no está autorizado
- Los mecanismos de seguridad deben defender al sistema frente a ataques internos y externos
 - Virus
 - Denegación de servicio
 - Usuarios ilegítimos



Entornos de computación

- Computadoras personales
- Dispositivos de mano (*mobile computing*): móviles y tabletas
- Sistemas embebidos (*embedded systems*)
- Servidores + multiprocesadores
- *Clusters* de servidores
- Supercomputadoras
- Sistemas distribuidos
- Sistemas virtualizados
- Sistemas en la nube (*clouds*)



Algunos tipos de sistemas operativos

- Procesamiento por lotes (*batch processing*)
- Tiempo compartido (*time sharing*)
- Tiempo real (*real time*)
- Sistemas multiusuario
- Máquinas virtuales



Sistemas de procesamiento por lotes

- Primera generación de SO (principios de los 1950)
- Objetivo:
 - Automatizar la ejecución de trabajos y aumentar la utilización y ocupación del procesador
- Los trabajos se agrupaban en **lotes** que se iban ejecutando en **secuencia**
 - Los más primitivos eran secuenciales; la multiprogramación se incorporó a finales de los '50
- Primer lenguaje para dictar tareas al SO: JCL (Job Control Language)



Sistemas de tiempo compartido

- Posteriores a los sistemas por lotes, comercializados en los '60
- Mejoras sobre los sistemas por lotes, para conseguir interactividad
- Cada proceso dispone de una pequeña rodaja (*slice*) de tiempo periódica. Los procesos se van turnando en el uso de la CPU
- Si la rodaja de tiempo es bastante pequeña (milisegundos), el usuario no percibe las pausas periódicas de su sesión



Sistemas de tiempo real (RTOS)

- Diseñados para cumplir tareas que deben completarse en un plazo prefijado (sistemas de control industrial, sistemas multimedia, ...)
- Usan algoritmos especiales para la planificación del procesador
- STR crítico
 - Para industria y sistemas embebidos en los que el cumplimiento de plazos es crítico
 - Suelen prescindir de servicios que afectan a los tiempos (ejemplo: memoria virtual)



Sistemas multiusuario

- Un sistema multiusuario reconoce que hay varios perfiles de acceso, con privilegios distintos:
 - Permisos de acceso a archivos y aplicaciones
 - Cuotas de espacio o de tiempo de procesador
 - Prioridad en el acceso a los recursos
- Multiusuario \neq multitarea
 - Puede haber sistemas multitarea que no son multiusuario



Sistemas multiprocesadores

- Desde unos pocos hasta miles de procesadores que se comunican entre sí, compartiendo buses y en ocasiones, el reloj, memoria y periféricos
- Varios modelos de acceso a memoria
 - UMA (Uniform Memory Access) La memoria física es compartida uniformemente por todo procesadores
 - Los procesadores tipo NUMA (Non Uniform Memory Access) tienen tiempos de acceso a la memoria compartida que dependen de la ubicación del proceso y la memoria
- Varios modelos de ejecución de procesos:
 - SMP multiprocesamiento simétrico: una tarea se puede ejecutar en cualquier procesador
 - AMP multiprocesamiento asimétrico: hay especialización de tareas (ejemplo: un procesador ejecuta el SO y otro los procesos de usuario)



Sistemas distribuidos

- Un sistema distribuido es un conjunto de computadoras conectadas en red y que se utiliza como si fuera un único sistema con múltiples procesadores + una gran memoria compartida + un gran almacenamiento secundario
- No existe un “sistema distribuido universal”, pero sí hay servicios con características de sistema distribuido
 - Servicios en la nube (Azure, Amazon, Google...)
 - World Wide Web



Sistemas virtualizados

- En un sistema virtualizado los servidores centrales (uno o más) soportan la carga de trabajo de todos los usuarios
- Los servidores tienen un software específico que permiten tener en ejecución, en la misma máquina, varias instancias del mismo sistema operativo, o incluso de diferentes
- Para utilizar estos sistemas centralizados, se necesitan terminales, denominados *ThinClients* o terminales ligeros, que se encargan de mostrar la interfaz de los sistemas operativos remotos
 - Los terminales se conectan a los servidores centrales, quienes están ejecutando los sistemas operativos demandados y las aplicaciones de cada cliente
 - Los terminales son simples consolas para poder visualizar la interfaz del sistema operativo que corre en remoto



Bibliografía complementaria

- Sistemas Operativos: aspectos internos y principios de diseño (Stallings, 2005, 5ª edición)
 - Capítulos 1 y 2
- Fundamentos de los Sistemas Operativos (Silberschatz, Galvin, Gagne. 2006, 7ª edición)
 - Capítulos 1 y 2