

Sistemas Operativos 1

Procesos

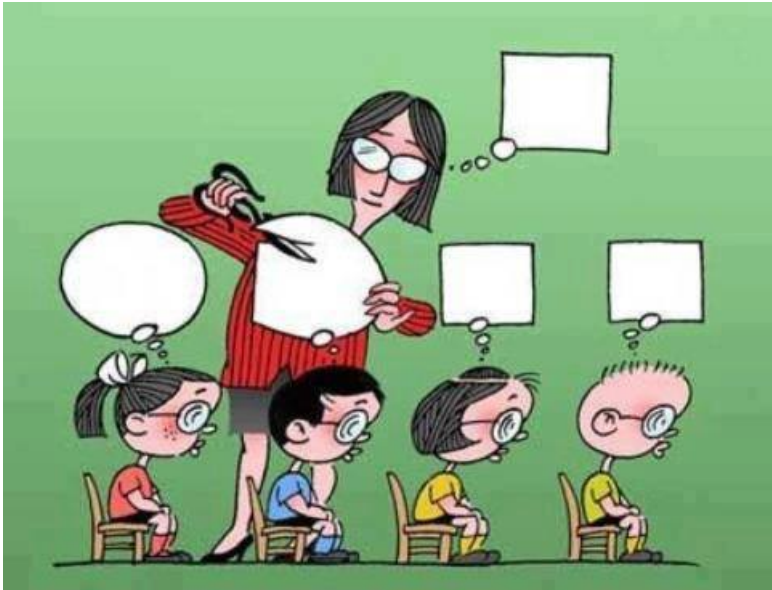
Edwin Salvador

15 de octubre de 2015

Sesión 3

¿En que consiste la buena educación?

Esto debe detenerse!!



Contenido I

- 1 Conceptos básicos
- 2 Núcleo del Sistema operativo
 - Tipos
- 3 El intérprete de comandos
- 4 Características de un proceso
 - El bloque de control de proceso
 - Traza de un proceso
- 5 Ejecución de instrucciones
 - Interrupciones
 - Trap
 - Las llamadas al sistema
- 6 Planificadores
- 7 Creación y Terminación de procesos
- 8 Estados
 - Modelos de estados de procesos
 - Swapping

Contenido II

9 Procesos en Windows y Ubuntu

Elementos de un SO



Definición

Un proceso para un SO en un conjunto de **instrucciones** que atraviesa dinámicamente un conjunto de **estados** y le solicita al sistema los **recursos** que le son necesarios para funcionar.

Definición

Un proceso para un SO en un conjunto de **instrucciones** que atraviesa dinámicamente un conjunto de **estados** y le solicita al sistema los **recursos** que le son necesarios para funcionar.

- Son los programas en ejecución de cualquier SO.

Definición

Un proceso para un SO en un conjunto de **instrucciones** que atraviesa dinámicamente un conjunto de **estados** y le solicita al sistema los **recursos** que le son necesarios para funcionar.

- Son los programas en ejecución de cualquier SO.
- Una acción u orden específica le indica al SO que debe empezar a ejecutar un programa.

Definición

Un proceso para un SO en un conjunto de **instrucciones** que atraviesa dinámicamente un conjunto de **estados** y le solicita al sistema los **recursos** que le son necesarios para funcionar.

- Son los programas en ejecución de cualquier SO.
- Una acción u orden específica le indica al SO que debe empezar a ejecutar un programa.
- Un proceso tiene períodos de ejecución activa y de espera.

Definición

Un proceso para un SO en un conjunto de **instrucciones** que atraviesa dinámicamente un conjunto de **estados** y le solicita al sistema los **recursos** que le son necesarios para funcionar.

- Son los programas en ejecución de cualquier SO.
- Una acción u orden específica le indica al SO que debe empezar a ejecutar un programa.
- Un proceso tiene períodos de ejecución activa y de espera.
- ¿Por qué debe esperar un proceso generalmente?

Definición

Un proceso para un SO en un conjunto de **instrucciones** que atraviesa dinámicamente un conjunto de **estados** y le solicita al sistema los **recursos** que le son necesarios para funcionar.

- Son los programas en ejecución de cualquier SO.
- Una acción u orden específica le indica al SO que debe empezar a ejecutar un programa.
- Un proceso tiene períodos de ejecución activa y de espera.
- ¿Por qué debe esperar un proceso generalmente? por la terminación de actividades de E/S.

Definición

Un proceso para un SO en un conjunto de **instrucciones** que atraviesa dinámicamente un conjunto de **estados** y le solicita al sistema los **recursos** que le son necesarios para funcionar.

- Son los programas en ejecución de cualquier SO.
- Una acción u orden específica le indica al SO que debe empezar a ejecutar un programa.
- Un proceso tiene períodos de ejecución activa y de espera.
- ¿Por qué debe esperar un proceso generalmente? por la terminación de actividades de E/S.
- Cuando un proceso está inactivo o en espera, el SO puede planificar la ejecución de otros procesos.

- Un proceso puede ser asignado y ejecutado por el procesador, pero el SO será quien controle su evolución y registrará los cambios que se produzcan en estos.

- Un proceso puede ser asignado y ejecutado por el procesador, pero el SO será quien controle su evolución y registrará los cambios que se produzcan en estos.
- Cada proceso dispone de atributos como: estado actual, unidad de planificación, derechos de acceso, prioridad, etc.

- Un proceso puede ser asignado y ejecutado por el procesador, pero el SO será quien controle su evolución y registrará los cambios que se produzcan en estos.
- Cada proceso dispone de atributos como: estado actual, unidad de planificación, derechos de acceso, prioridad, etc.
- El SO planifica y gestiona los procesos para conseguir que su utilización sea lo más óptima posible.

- Un proceso puede ser asignado y ejecutado por el procesador, pero el SO será quien controle su evolución y registrará los cambios que se produzcan en estos.
- Cada proceso dispone de atributos como: estado actual, unidad de planificación, derechos de acceso, prioridad, etc.
- El SO planifica y gestiona los procesos para conseguir que su utilización sea lo más óptima posible.
- Tenemos varios tipos de procesos y los podemos clasificar en: procesos de usuario y procesos de sistema.

- Un proceso puede ser asignado y ejecutado por el procesador, pero el SO será quien controle su evolución y registrará los cambios que se produzcan en estos.
- Cada proceso dispone de atributos como: estado actual, unidad de planificación, derechos de acceso, prioridad, etc.
- El SO planifica y gestiona los procesos para conseguir que su utilización sea lo más óptima posible.
- Tenemos varios tipos de procesos y los podemos clasificar en: procesos de usuario y procesos de sistema.
- **Procesos de usuario** aquel creado por el sistema operativo como respuesta a una acción del usuario o de una aplicación ejecutada por este.

- Un proceso puede ser asignado y ejecutado por el procesador, pero el SO será quien controle su evolución y registrará los cambios que se produzcan en estos.
- Cada proceso dispone de atributos como: estado actual, unidad de planificación, derechos de acceso, prioridad, etc.
- El SO planifica y gestiona los procesos para conseguir que su utilización sea lo más óptima posible.
- Tenemos varios tipos de procesos y los podemos clasificar en: procesos de usuario y procesos de sistema.
- **Procesos de usuario** aquel creado por el sistema operativo como respuesta a una acción del usuario o de una aplicación ejecutada por este.
- **Procesos de sistema** forman parte del propio SO y desempeña alguna de sus características. Ej. la elección del siguiente proceso a ejecutar o bien acceder a un recurso de E/S.

Requisitos mínimos de un SO

- Todos los SO actuales están basados en procesos.

Requisitos mínimos de un SO

- Todos los SO actuales están basados en procesos.
- Un SO debe cumplir algunos **requisitos mínimos para trabajar con procesos**:

Requisitos mínimos de un SO

- Todos los SO actuales están basados en procesos.
- Un SO debe cumplir algunos **requisitos mínimos para trabajar con procesos**:
 - debe ser capaz de intercalar la ejecución de procesos (mayor utilización de CPU mejor tiempo de respuesta)

Requisitos mínimos de un SO

- Todos los SO actuales están basados en procesos.
- Un SO debe cumplir algunos **requisitos mínimos para trabajar con procesos**:
 - debe ser capaz de intercalar la ejecución de procesos (mayor utilización de CPU mejor tiempo de respuesta)
 - debe asignar los recursos disponibles a los procesos de manera óptima (evitar **interbloqueos**).

Requisitos mínimos de un SO

- Todos los SO actuales están basados en procesos.
- Un SO debe cumplir algunos **requisitos mínimos para trabajar con procesos**:
 - debe ser capaz de intercalar la ejecución de procesos (mayor utilización de CPU mejor tiempo de respuesta)
 - debe asignar los recursos disponibles a los procesos de manera óptima (evitar **interbloqueos**).
 - debe dar soporte a la comunicación entre procesos y ofrecer mecanismos para su creación.

Proceso vs Programa

Programa	Proceso
Estático.	Dinámico.
No tiene contador de programa.	Tiene un contador de programa.
Existe desde que se instala hasta que se borra.	Su ciclo de vida comprende desde que se activa hasta que termina.

Un proceso se puede considerar como una instancia de un programa. Un programa puede estar compuesto de varios procesos.

Contenido I

- 1 Conceptos básicos
- 2 Núcleo del Sistema operativo
 - Tipos
- 3 El intérprete de comandos
- 4 Características de un proceso
 - El bloque de control de proceso
 - Traza de un proceso
- 5 Ejecución de instrucciones
 - Interrupciones
 - Trap
 - Las llamadas al sistema
- 6 Planificadores
- 7 Creación y Terminación de procesos
- 8 Estados
 - Modelos de estados de procesos
 - Swapping

Contenido II

9 Procesos en Windows y Ubuntu

Núcleo del Sistema operativo

- Conocido como *kernel*.

Núcleo del Sistema operativo

- Conocido como *kernel*.
- Su función principal es interactuar directamente con el hardware para gestionar el procesador, la memoria, la E/S y el resto de recursos disponibles.

Núcleo del Sistema operativo

- Conocido como *kernel*.
- Su función principal es interactuar directamente con el hardware para gestionar el procesador, la memoria, la E/S y el resto de recursos disponibles.
- Controla todas las operaciones de los procesos.

Núcleo del Sistema operativo

- Conocido como *kernel*.
- Su función principal es interactuar directamente con el hardware para gestionar el procesador, la memoria, la E/S y el resto de recursos disponibles.
- Controla todas las operaciones de los procesos.
- Generalmente, es solo una pequeña parte del SO pero es la más utilizada.

Núcleo del Sistema operativo

- Conocido como *kernel*.
- Su función principal es interactuar directamente con el hardware para gestionar el procesador, la memoria, la E/S y el resto de recursos disponibles.
- Controla todas las operaciones de los procesos.
- Generalmente, es solo una pequeña parte del SO pero es la más utilizada.
- Son diseñados para realizar el mínimo posible de procesamiento en cada interrupción y dejar que el resto lo realice proceso apropiado, de esta manera el núcleo queda libre para realizar otras operaciones.

Funciones del núcleo

- Gestionar la memoria

Funciones del núcleo

- Gestionar la memoria
- Administrar el sistema de archivos

Funciones del núcleo

- Gestionar la memoria
- Administrar el sistema de archivos
- Asignar recursos entre los procesos de usuarios

Funciones del núcleo

- Gestionar la memoria
- Administrar el sistema de archivos
- Asignar recursos entre los procesos de usuarios
- Administrar los servicios de E/S

Funciones del núcleo

- Gestionar la memoria
- Administrar el sistema de archivos
- Asignar recursos entre los procesos de usuarios
- Administrar los servicios de E/S
- Cambiar el estado de procesos

Funciones del núcleo

- Gestionar la memoria
- Administrar el sistema de archivos
- Asignar recursos entre los procesos de usuarios
- Administrar los servicios de E/S
- Cambiar el estado de procesos
- Apoyar ciertas funciones de contabilidad del sistema.

- **Nivel 1. Gestión de memoria:** proporciona facilidades a bajo nivel para la gestión de memoria secundaria para ejecutar procesos.

Capas de un núcleo

- **Nivel 1. Gestión de memoria:** proporciona facilidades a bajo nivel para la gestión de memoria secundaria para ejecutar procesos.
- **Nivel 2. Procesos:** activa los contadores de tiempo para cada uno de los procesos, creando interrupciones de hardware cuando no son respetadas.

Capas de un núcleo

- **Nivel 1. Gestión de memoria:** proporciona facilidades a bajo nivel para la gestión de memoria secundaria para ejecutar procesos.
- **Nivel 2. Procesos:** activa los contadores de tiempo para cada uno de los procesos, creando interrupciones de hardware cuando no son respetadas.
- **Nivel 3. E/S:** proporciona las facilidades para poder utilizar dispositivos E/S requeridos por los procesos.

Capas de un núcleo

- **Nivel 1. Gestión de memoria:** proporciona facilidades a bajo nivel para la gestión de memoria secundaria para ejecutar procesos.
- **Nivel 2. Procesos:** activa los contadores de tiempo para cada uno de los procesos, creando interrupciones de hardware cuando no son respetadas.
- **Nivel 3. E/S:** proporciona las facilidades para poder utilizar dispositivos E/S requeridos por los procesos.
- **Nivel 4. Información, aplicación o interprete de lenguajes:** facilita la comunicación con los lenguajes y el SO para aceptar las órdenes en cada una de las aplicaciones.

Capas de un núcleo

- **Nivel 1. Gestión de memoria:** proporciona facilidades a bajo nivel para la gestión de memoria secundaria para ejecutar procesos.
- **Nivel 2. Procesos:** activa los contadores de tiempo para cada uno de los procesos, creando interrupciones de hardware cuando no son respetadas.
- **Nivel 3. E/S:** proporciona las facilidades para poder utilizar dispositivos E/S requeridos por los procesos.
- **Nivel 4. Información, aplicación o interprete de lenguajes:** facilita la comunicación con los lenguajes y el SO para aceptar las órdenes en cada una de las aplicaciones.
- **Nivel 5. Control de archivos:** proporciona la facilidad para el almacenamiento a largo plazo y la manipulación de archivos con nombre. Asigna espacio y acceso de datos en la memoria.

Contenido I

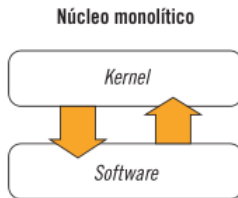
- 1 Conceptos básicos
- 2 Núcleo del Sistema operativo
 - Tipos
- 3 El intérprete de comandos
- 4 Características de un proceso
 - El bloque de control de proceso
 - Traza de un proceso
- 5 Ejecución de instrucciones
 - Interrupciones
 - Trap
 - Las llamadas al sistema
- 6 Planificadores
- 7 Creación y Terminación de procesos
- 8 Estados
 - Modelos de estados de procesos
 - Swapping
- 9 Procesos en Windows y Ubuntu

Monolíticos

- Abarcan todos los servicios del sistema (sistema de archivos, controladores de dispositivos, redes, planificación, gestión de memoria, etc).

Monolíticos

- Abarcan todos los servicios del sistema (sistema de archivos, controladores de dispositivos, redes, planificación, gestión de memoria, etc).
- Ej: Unix, Linux



Micronúcleos (microkernel)

- Rendimiento inferior al del monolítico.

Micronúcleos (microkernel)

- Rendimiento inferior al del monolítico.
- Menos complejos.

Micronúcleos (microkernel)

- Rendimiento inferior al del monolítico.
- Menos complejos.
- Descentralización de fallos.

Micronúcleos (microkernel)

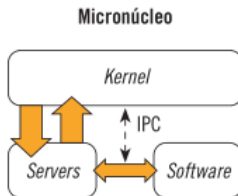
- Rendimiento inferior al del monolítico.
- Menos complejos.
- Descentralización de fallos.
- Facilidad de crear y depurar controladores de dispositivos.

Micronúcleos (microkernel)

- Rendimiento inferior al del monolítico.
- Menos complejos.
- Descentralización de fallos.
- Facilidad de crear y depurar controladores de dispositivos.
- Proporcionan conjunto mínimo de llamadas para implementar servicios básicos (planificación, comunicación de procesos, etc)

Micronúcleos (microkernel)

- Rendimiento inferior al del monolítico.
- Menos complejos.
- Descentralización de fallos.
- Facilidad de crear y depurar controladores de dispositivos.
- Proporcionan conjunto mínimo de llamadas para implementar servicios básicos (planificación, comunicación de procesos, etc)
- Los demás procesos se ejecutan como procesos de servidores en espacio de usuario.



Híbridos

- Basados en los micronúcleos.

Híbridos

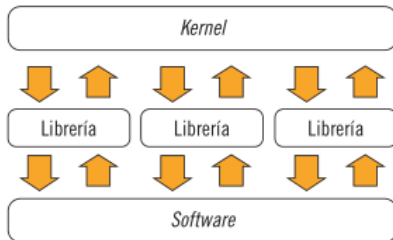
- Basados en los micronúcleos.
- Incluyen código adicional en el espacio del núcleo para que su ejecución sea más rápida que si estuviera en espacio de usuario.

Híbridos

- Basados en los micronúcleos.
- Incluyen código adicional en el espacio del núcleo para que su ejecución sea más rápida que si estuviera en espacio de usuario.
- La mayoría de SO actuales como Windows o Mac OS X tienen este tipo de núcleos.

Exonúcleos

- Permiten el uso de bibliotecas lo que incrementa su funcionalidad al tener acceso casi directo al hardware.



Contenido I

- 1 Conceptos básicos
- 2 Núcleo del Sistema operativo
 - Tipos
- 3 El intérprete de comandos
- 4 Características de un proceso
 - El bloque de control de proceso
 - Traza de un proceso
- 5 Ejecución de instrucciones
 - Interrupciones
 - Trap
 - Las llamadas al sistema
- 6 Planificadores
- 7 Creación y Terminación de procesos
- 8 Estados
 - Modelos de estados de procesos
 - Swapping

Contenido II

9 Procesos en Windows y Ubuntu

El intérprete de comandos

- Conocido como *shell* .

El intérprete de comandos

- Conocido como *shell* .
- Un programa que interpreta las órdenes del usuario y las convierte en llamadas al sistema.

El intérprete de comandos

- Conocido como *shell* .
- Un programa que interpreta las órdenes del usuario y las convierte en llamadas al sistema.
- No es parte del SO.

El intérprete de comandos

- Conocido como *shell* .
- Un programa que interpreta las órdenes del usuario y las convierte en llamadas al sistema.
- No es parte del SO.
- Es fundamental para el funcionamiento del SO ya que este lo utiliza para ejecutar órdenes básicas para el manejo.

Ejemplos de shell

- **Command.com** de MS-DOS, Win 95/98/Me. Puede ejecutarse en modo interactivo o por lotes (secuencia de comandos).

Ejemplos de shell

- **Command.com** de MS-DOS, Win 95/98/Me. Puede ejecutarse en modo interactivo o por lotes (secuencia de comandos).
- **cmd.exe** Evolución del Command.com. XP, Server 2003, Vista, 7, 8 y ¿10?).

Ejemplos de shell

- **Command.com** de MS-DOS, Win 95/98/Me. Puede ejecutarse en modo interactivo o por lotes (secuencia de comandos).
- **cmd.exe** Evolución del Command.com. XP, Server 2003, Vista, 7, 8 y ¿10?).
- **Bash** Por defecto de la mayoría de distribuciones Linux. Consiste en un intérprete de órdenes de Unix escrito para el proyecto GNU.

Ejemplos de shell

- **Command.com** de MS-DOS, Win 95/98/Me. Puede ejecutarse en modo interactivo o por lotes (secuencia de comandos).
- **cmd.exe** Evolución del Command.com. XP, Server 2003, Vista, 7, 8 y ¿10?).
- **Bash** Por defecto de la mayoría de distribuciones Linux. Consiste en un intérprete de órdenes de Unix escrito para el proyecto GNU.
- **Ksh** *korn shell* de Unix. Interpreta órdenes por línea. En algunas distribuciones de GNU/Linux. Bajo licencia GPL.

Ejemplos de shell

- **Command.com** de MS-DOS, Win 95/98/Me. Puede ejecutarse en modo interactivo o por lotes (secuencia de comandos).
- **cmd.exe** Evolución del Command.com. XP, Server 2003, Vista, 7, 8 y ¿10?).
- **Bash** Por defecto de la mayoría de distribuciones Linux. Consiste en un intérprete de órdenes de Unix escrito para el proyecto GNU.
- **Ksh** *korn shell* de Unix. Interpreta órdenes por línea. En algunas distribuciones de GNU/Linux. Bajo licencia GPL.
- **Bourne Shell** usado en primeras versiones Unix.

Ejemplo del uso de Shell

- En Windows

Ejemplo del uso de Shell

- En Windows
- En Ubuntu

Contenido I

- 1 Conceptos básicos
- 2 Núcleo del Sistema operativo
 - Tipos
- 3 El intérprete de comandos
- 4 Características de un proceso
 - El bloque de control de proceso
 - Traza de un proceso
- 5 Ejecución de instrucciones
 - Interrupciones
 - Trap
 - Las llamadas al sistema
- 6 Planificadores
- 7 Creación y Terminación de procesos
- 8 Estados
 - Modelos de estados de procesos
 - Swapping

Contenido II

9 Procesos en Windows y Ubuntu

Elementos que caracterizan a un proceso

Un proceso tiene algunos elementos que lo caracterizan:

- **Identificador:** un identificador único asociado al proceso para distinguirlo del resto.

Elementos que caracterizan a un proceso

Un proceso tiene algunos elementos que lo caracterizan:

- **Identificador:** un identificador único asociado al proceso para distinguirlo del resto.
- **Estado:** Puede estar en diferentes estados. Si está actualmente corriendo entonces será *en ejecución*. Veremos los posibles estados más adelante.

Elementos que caracterizan a un proceso

Un proceso tiene algunos elementos que lo caracterizan:

- **Identificador:** un identificador único asociado al proceso para distinguirlo del resto.
- **Estado:** Puede estar en diferentes estados. Si está actualmente corriendo entonces será *en ejecución*. Veremos los posibles estados más adelante.
- **Prioridad:** Una prioridad relativa al resto de procesos para determinar cuando debe ser ejecutado.

Elementos que caracterizan a un proceso

Un proceso tiene algunos elementos que lo caracterizan:

- **Identificador:** un identificador único asociado al proceso para distinguirlo del resto.
- **Estado:** Puede estar en diferentes estados. Si está actualmente corriendo entonces será *en ejecución*. Veremos los posibles estados más adelante.
- **Prioridad:** Una prioridad relativa al resto de procesos para determinar cuando debe ser ejecutado.
- **Contador de programa (PC):** Indica la dirección de la siguiente instrucción que debe ejecutarse.

Elementos que caracterizan a un proceso

Un proceso tiene algunos elementos que lo caracterizan:

- **Identificador:** un identificador único asociado al proceso para distinguirlo del resto.
- **Estado:** Puede estar en diferentes estados. Si está actualmente corriendo entonces será *en ejecución*. Veremos los posibles estados más adelante.
- **Prioridad:** Una prioridad relativa al resto de procesos para determinar cuando debe ser ejecutado.
- **Contador de programa (PC):** Indica la dirección de la siguiente instrucción que debe ejecutarse.
- **Punteros a memoria:** Los punteros al código de programa y los datos asociados al proceso además de los bloques de memoria compartidos con otros procesos.

Elementos que caracterizan a un proceso

- **Datos de contexto:** los datos presentes en los registros del procesador cuando el proceso está corriendo.

Elementos que caracterizan a un proceso

- **Datos de contexto:** los datos presentes en los registros del procesador cuando el proceso está corriendo.
- **Información de estado de E/S:** incluye los dispositivos E/S asignados al proceso, las peticiones de E/S pendientes, lista de ficheros en uso por el proceso, etc.

Elementos que caracterizan a un proceso

- **Datos de contexto:** los datos presentes en los registros del procesador cuando el proceso está corriendo.
- **Información de estado de E/S:** incluye los dispositivos E/S asignados al proceso, las peticiones de E/S pendientes, lista de ficheros en uso por el proceso, etc.
- **Información de auditoría:** la cantidad de tiempo de procesador y tiempo de reloj utilizados por el proceso.

Contenido I

- 1 Conceptos básicos
- 2 Núcleo del Sistema operativo
 - Tipos
- 3 El intérprete de comandos
- 4 Características de un proceso
 - El bloque de control de proceso
 - Traza de un proceso
- 5 Ejecución de instrucciones
 - Interrupciones
 - Trap
 - Las llamadas al sistema
- 6 Planificadores
- 7 Creación y Terminación de procesos
- 8 Estados
 - Modelos de estados de procesos
 - Swapping
- 9 Procesos en Windows y Ubuntu

El Bloque de control de proceso (BCP)

- Todos los elementos de la lista anterior son almacenados en una estructura de datos llamada **bloque de control de proceso (BCP)**.
- El BCP es creado y gestionado por el SO y contiene información suficiente para que el proceso pueda ser interrumpido y luego se lo pueda restaurar como si no hubiera existido ninguna interrupción.
- Es gracias al BCP que el SO puede soportar la **multiprogramación**.
- En un computador monoprocesador solo un proceso puede estar en estado *en ejecución* en un instante determinado.

Identificador
Estado
Prioridad
Contador de programa
Punteros de memoria
Datos de contexto
Información de estado de E/S
Información de auditoría
⋮

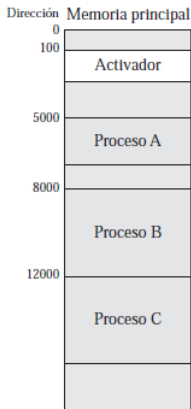
Contenido I

- 1 Conceptos básicos
- 2 Núcleo del Sistema operativo
 - Tipos
- 3 El intérprete de comandos
- 4 Características de un proceso
 - El bloque de control de proceso
 - Traza de un proceso
- 5 Ejecución de instrucciones
 - Interrupciones
 - Trap
 - Las llamadas al sistema
- 6 Planificadores
- 7 Creación y Terminación de procesos
- 8 Estados
 - Modelos de estados de procesos
 - Swapping
- 9 Procesos en Windows y Ubuntu

- Se puede caracterizar el comportamiento de un proceso, listando su secuencia de instrucciones ejecutadas. A esta lista se la conoce como **traza** del proceso.

- Se puede caracterizar el comportamiento de un proceso, listando su secuencia de instrucciones ejecutadas. A esta lista se la conoce como **traza** del proceso.
- Se puede observar el comportamiento de un procesador mostrando como las trazas de varios procesos se entrelazan.

Ilustración de trazas y trazas combinadas



5000	8000	12000
5001	8001	12001
5002	8002	12002
5003	8003	12003
5004		12004
5005		12005
5006		12006
5007		12007
5008		12008
5009		12009
5010		12010
5011		12011
(a) Trazas del Proceso A	(b) Trazas del Proceso B	(c) Trazas del Proceso C

5000 = Dirección de comienzo del programa del Proceso A.
8000 = Dirección de comienzo del programa del Proceso B.
12000 = Dirección de comienzo del programa del Proceso C.

Activador: Intercambia el procesador de un proceso a otro.

Contador de programa: indica la dirección de la siguiente instrucción a ejecutar.

Ilustración de trazas y trazas combinadas

1	5000		27	12004	
2	5001		28	12005	
3	5002				Temporización
4	5003		29	100	
5	5004		30	101	
6	5005		31	102	
			32	103	
7	100	Temporización	33	104	
8	101		34	105	
9	102		35	5006	
10	103		36	5007	
11	104		37	5008	
12	105		38	5009	
13	8000		39	5010	
14	8001		40	5011	
15	8002				Temporización
16	8003		41	100	
		Petición de E/S	42	101	
17	100		43	102	
18	101		44	103	
19	102		45	104	
20	103		46	105	
21	104		47	12006	
22	105		48	12007	
23	12000		49	12008	
24	12001		50	12009	
25	12002		51	12010	
26	12003		52	12011	
					Temporización

100 = Dirección de comienzo del programa activador.

Las zonas sombreadas indican la ejecución del proceso de activación:

la primera y la tercera columna cuentan ciclos de instrucciones:

la segunda y la cuarta columna las direcciones de las instrucciones que se ejecutan

Contenido I

- 1 Conceptos básicos
- 2 Núcleo del Sistema operativo
 - Tipos
- 3 El intérprete de comandos
- 4 Características de un proceso
 - El bloque de control de proceso
 - Traza de un proceso
- 5 **Ejecución de instrucciones**
 - Interrupciones
 - Trap
 - Las llamadas al sistema
- 6 Planificadores
- 7 Creación y Terminación de procesos
- 8 Estados
 - Modelos de estados de procesos
 - Swapping

Contenido II

9 Procesos en Windows y Ubuntu

Ejecución de instrucciones

Ciclos de instrucción

- El procesamiento de una instrucción se denomina **ciclo de instrucción**

Ejecución de instrucciones

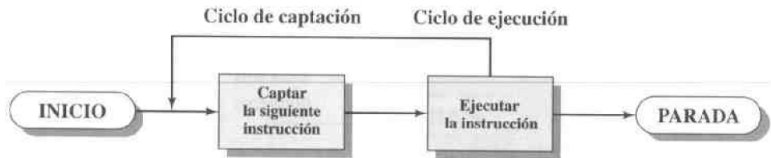
Ciclos de instrucción

- El procesamiento de una instrucción se denomina **ciclo de instrucción**
- Este consta de dos etapas: **ciclo de captación** y **ciclo de ejecución**.

Ejecución de instrucciones

Ciclos de instrucción

- El procesamiento de una instrucción se denomina **ciclo de instrucción**
- Este consta de dos etapas: **ciclo de captación** y **ciclo de ejecución**.
- La ejecución de un programa solo se detiene si la máquina se desconecta, se produce algún error *irrecuperable* o ejecuta una instrucción del programa que lo detiene.



Ejecución de instrucciones

- El procesador capta la instrucción y el contador de programa (PC) aumenta secuencialmente. (instrucción en posición 300 seguirá 301, 302)

Ejecución de instrucciones

- El procesador capta la instrucción y el contador de programa (PC) aumenta secuencialmente. (instrucción en posición 300 seguirá 301, 302)
- La instrucción captada se almacena en el registro de instrucción (IR).

Ejecución de instrucciones

- El procesador capta la instrucción y el contador de programa (PC) aumenta secuencialmente. (instrucción en posición 300 seguirá 301, 302)
- La instrucción captada se almacena en el registro de instrucción (IR).
- El procesador interpreta instrucción y puede llevar a cabo una o varias de las siguientes acciones:

Ejecución de instrucciones

- El procesador capta la instrucción y el contador de programa (PC) aumenta secuencialmente. (instrucción en posición 300 seguirá 301, 302)
- La instrucción captada se almacena en el registro de instrucción (IR).
- El procesador interpreta instrucción y puede llevar a cabo una o varias de las siguientes acciones:
 - **Procesador-memoria** datos desde procesador a memoria o viceversa.

Ejecución de instrucciones

- El procesador capta la instrucción y el contador de programa (PC) aumenta secuencialmente. (instrucción en posición 300 seguirá 301, 302)
- La instrucción captada se almacena en el registro de instrucción (IR).
- El procesador interpreta instrucción y puede llevar a cabo una o varias de las siguientes acciones:
 - **Procesador-memoria** datos desde procesador a memoria o viceversa.
 - **Procesador-E/S** datos desde E/S o viceversa.

Ejecución de instrucciones

- El procesador capta la instrucción y el contador de programa (PC) aumenta secuencialmente. (instrucción en posición 300 seguirá 301, 302)
- La instrucción captada se almacena en el registro de instrucción (IR).
- El procesador interpreta instrucción y puede llevar a cabo una o varias de las siguientes acciones:
 - **Procesador-memoria** datos desde procesador a memoria o viceversa.
 - **Procesador-E/S** datos desde E/S o viceversa.
 - **Procesamiento de datos** operaciones aritméticas o lógicas.

Ejecución de instrucciones

- El procesador capta la instrucción y el contador de programa (PC) aumenta secuencialmente. (instrucción en posición 300 seguirá 301, 302)
- La instrucción captada se almacena en el registro de instrucción (IR).
- El procesador interpreta instrucción y puede llevar a cabo una o varias de las siguientes acciones:
 - **Procesador-memoria** datos desde procesador a memoria o viceversa.
 - **Procesador-E/S** datos desde E/S o viceversa.
 - **Procesamiento de datos** operaciones aritméticas o lógicas.
 - **Control** Controla las alteraciones en las secuencias de ejecución de instrucciones.

Contenido I

- 1 Conceptos básicos
- 2 Núcleo del Sistema operativo
 - Tipos
- 3 El intérprete de comandos
- 4 Características de un proceso
 - El bloque de control de proceso
 - Traza de un proceso
- 5 Ejecución de instrucciones
 - Interrupciones
 - Trap
 - Las llamadas al sistema
- 6 Planificadores
- 7 Creación y Terminación de procesos
- 8 Estados
 - Modelos de estados de procesos
 - Swapping
- 9 Procesos en Windows y Ubuntu

Interrupciones

- Mecanismo que permite interrumpir la secuencia normal de procesamiento de la CPU ante condiciones particulares.

Interrupciones

- Mecanismo que permite interrumpir la secuencia normal de procesamiento de la CPU ante condiciones particulares.
- Mejoran el eficiencia del procesador.

Interrupciones

- Mecanismo que permite interrumpir la secuencia normal de procesamiento de la CPU ante condiciones particulares.
- Mejoran el eficiencia del procesador.
- Permiten al procesador ejecutar otras instrucciones mientras otra operación con un dispositivo más lento está en curso.

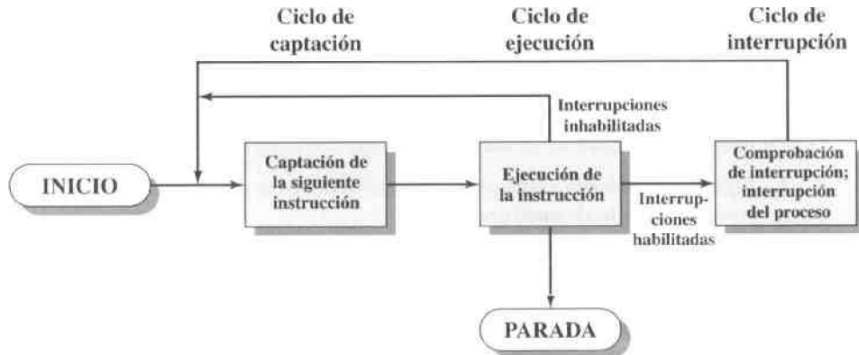
Interrupciones

- Mecanismo que permite interrumpir la secuencia normal de procesamiento de la CPU ante condiciones particulares.
- Mejoran la eficiencia del procesador.
- Permiten al procesador ejecutar otras instrucciones mientras otra operación con un dispositivo más lento está en curso.
- Ejemplo: Los dispositivos de E/S son más lentos por lo tanto el procesador tendría que esperar hasta que el dispositivo de E/S (impresora) complete su operación de escritura. Lo cual podría tomar miles de ciclos de instrucción que sería un desperdicio del uso del procesador.

Interrupciones

- Mecanismo que permite interrumpir la secuencia normal de procesamiento de la CPU ante condiciones particulares.
- Mejoran el eficiencia del procesador.
- Permiten al procesador ejecutar otras instrucciones mientras otra operación con un dispositivo más lento está en curso.
- Ejemplo: Los dispositivos de E/S son más lentos por lo tanto el procesador tendría que esperar hasta que el dispositivo de E/S (impresora) complete su operación de escritura. Lo cual podría tomar miles de ciclos de instrucción que sería un derroche del uso del procesador.
- El procesador y el SO son los encargados de detener el programa de usuario y después permitir que siga en el mismo punto.

Interrupciones



Contenido I

- 1 Conceptos básicos
- 2 Núcleo del Sistema operativo
 - Tipos
- 3 El intérprete de comandos
- 4 Características de un proceso
 - El bloque de control de proceso
 - Traza de un proceso
- 5 Ejecución de instrucciones
 - Interrupciones
 - **Trap**
 - Las llamadas al sistema
- 6 Planificadores
- 7 Creación y Terminación de procesos
- 8 Estados
 - Modelos de estados de procesos
 - Swapping
- 9 Procesos en Windows y Ubuntu

- Similar a una interrupción pero asociada a la ejecución de la instrucción actual (condición de error o excepción, operación que implica una llamada de E/S como abrir un archivo).

Contenido I

- 1 Conceptos básicos
- 2 Núcleo del Sistema operativo
 - Tipos
- 3 El intérprete de comandos
- 4 Características de un proceso
 - El bloque de control de proceso
 - Traza de un proceso
- 5 Ejecución de instrucciones
 - Interrupciones
 - Trap
 - Las llamadas al sistema
- 6 Planificadores
- 7 Creación y Terminación de procesos
- 8 Estados
 - Modelos de estados de procesos
 - Swapping
- 9 Procesos en Windows y Ubuntu

Las llamadas al sistema

Definición

Las llamadas al sistema constituyen la interfaz entre el sistema operativo y los procesos.

- Los programas utilizan las llamadas del sistema para solicitar servicios del SO.

Las llamadas al sistema

Definición

Las llamadas al sistema constituyen la interfaz entre el sistema operativo y los procesos.

- Los programas utilizan las llamadas del sistema para solicitar servicios del SO.
- A cada llamada del sistema le corresponde un procedimiento de la biblioteca que puede llamar a los programas del usuario.

Definición

Las llamadas al sistema constituyen la interfaz entre el sistema operativo y los procesos.

- Los programas utilizan las llamadas del sistema para solicitar servicios del SO.
- A cada llamada del sistema le corresponde un procedimiento de la biblioteca que puede llamar a los programas del usuario.
- Se la hace por medio de lenguajes ensambladores, aunque en algunos casos existen facilidades que permiten que se realicen desde lenguajes de alto nivel.

Llamadas al sistema

- Se las realiza por medio de instrucciones de máquina *Trap*.

Llamadas al sistema

- Se las realiza por medio de instrucciones de máquina *Trap*.
- Se guarda el contador de programa y la palabra estado del procesador (PSW).

Llamadas al sistema

- Se las realiza por medio de instrucciones de máquina *Trap*.
- Se guarda el contador de programa y la palabra estado del procesador (PSW).
- Se carga un nuevo contador de programa y una nueva PSW.

Llamadas al sistema

- Se las realiza por medio de instrucciones de máquina *Trap*.
- Se guarda el contador de programa y la palabra estado del procesador (PSW).
- Se carga un nuevo contador de programa y una nueva PSW.
- El nuevo contador de programa contiene una dirección de memoria donde reside un programa de SO que ejecuta el servicio solicitado.

Llamadas al sistema

- Se las realiza por medio de instrucciones de máquina *Trap*.
- Se guarda el contador de programa y la palabra estado del procesador (PSW).
- Se carga un nuevo contador de programa y una nueva PSW.
- El nuevo contador de programa contiene una dirección de memoria donde reside un programa de SO que ejecuta el servicio solicitado.
- Al finalizar el servicio se indica si hubo éxito o fracaso y se ejecuta una instrucción *return from trap*.

Llamadas al sistema

- Se las realiza por medio de instrucciones de máquina *Trap*.
- Se guarda el contador de programa y la palabra estado del procesador (PSW).
- Se carga un nuevo contador de programa y una nueva PSW.
- El nuevo contador de programa contiene una dirección de memoria donde reside un programa de SO que ejecuta el servicio solicitado.
- Al finalizar el servicio se indica si hubo éxito o fracaso y se ejecuta una instrucción *return from trap*.
- Se restaura el contador de programa y la PSW original y continúa la ejecución.

Llamadas al sistema

- Se las realiza por medio de instrucciones de máquina *Trap*.
- Se guarda el contador de programa y la palabra estado del procesador (PSW).
- Se carga un nuevo contador de programa y una nueva PSW.
- El nuevo contador de programa contiene una dirección de memoria donde reside un programa de SO que ejecuta el servicio solicitado.
- Al finalizar el servicio se indica si hubo éxito o fracaso y se ejecuta una instrucción *return from trap*.
- Se restaura el contador de programa y la PSW original y continúa la ejecución.
- Las llamadas al sistema se ejecutan de manera similar en distintos SO.

Categorías de llamadas al sistema

- **Control de procesos** finalizar, abortar, cargar, ejecutar, crear, terminar, establecer y obtener atributos del proceso, esperar un tiempo, señalar y esperar evento, asignar y liberar memoria.

Categorías de llamadas al sistema

- **Control de procesos** finalizar, abortar, cargar, ejecutar, crear, terminar, establecer y obtener atributos del proceso, esperar un tiempo, señalar y esperar evento, asignar y liberar memoria.
- **Manipulación de archivos** crear, eliminar, abrir, cerrar, leer, escribir, reposicionar, obtener y establecer atributos de archivo.

Categorías de llamadas al sistema

- **Control de procesos** finalizar, abortar, cargar, ejecutar, crear, terminar, establecer y obtener atributos del proceso, esperar un tiempo, señalar y esperar evento, asignar y liberar memoria.
- **Manipulación de archivos** crear, eliminar, abrir, cerrar, leer, escribir, reposicionar, obtener y establecer atributos de archivo.
- **Manipulación de dispositivos** solicitar, liberar, leer, escribir, reposicionar, obtener y establecer atributos del dispositivo.

Categorías de llamadas al sistema

- **Control de procesos** finalizar, abortar, cargar, ejecutar, crear, terminar, establecer y obtener atributos del proceso, esperar un tiempo, señalar y esperar evento, asignar y liberar memoria.
- **Manipulación de archivos** crear, eliminar, abrir, cerrar, leer, escribir, reposicionar, obtener y establecer atributos de archivo.
- **Manipulación de dispositivos** solicitar, liberar, leer, escribir, reposicionar, obtener y establecer atributos del dispositivo.
- **Mantenimiento de información** obtener fecha y hora, datos del sistema y atributos.

Categorías de llamadas al sistema

- **Control de procesos** finalizar, abortar, cargar, ejecutar, crear, terminar, establecer y obtener atributos del proceso, esperar un tiempo, señalar y esperar evento, asignar y liberar memoria.
- **Manipulación de archivos** crear, eliminar, abrir, cerrar, leer, escribir, reposicionar, obtener y establecer atributos de archivo.
- **Manipulación de dispositivos** solicitar, liberar, leer, escribir, reposicionar, obtener y establecer atributos del dispositivo.
- **Mantenimiento de información** obtener fecha y hora, datos del sistema y atributos.
- **Comunicaciones** crear, eliminar conexión de comunicación, enviar y recibir mensajes, transferir información de estado, etc.

Llamadas al sistema para gestión de archivos

- Muchas de las llamadas al sistema tienen relación con el sistema de archivos.

Llamadas al sistema para gestión de archivos

- Muchas de las llamadas al sistema tienen relación con el sistema de archivos.
- Al abrir un archivo se genera una llamada: *O_RDONLY*, *O_WRONLY*, *O_RDWR*.

Llamadas al sistema para gestión de archivos

- Muchas de las llamadas al sistema tienen relación con el sistema de archivos.
- Al abrir un archivo se genera una llamada: *O_RDONLY*, *O_WRONLY*, *O_RDWR*.
- Las más utilizadas son *read* y *write*.

Llamadas al sistema para gestión de archivos

- Muchas de las llamadas al sistema tienen relación con el sistema de archivos.
- Al abrir un archivo se genera una llamada: *O_RDONLY*, *O_WRONLY*, *O_RDWR*.
- Las más utilizadas son *read* y *write*.
- *O_CREAT* para crear un archivo.

Llamadas al sistema para gestión de archivos

- Muchas de las llamadas al sistema tienen relación con el sistema de archivos.
- Al abrir un archivo se genera una llamada: *O_RDONLY*, *O_WRONLY*, *O_RDWR*.
- Las más utilizadas son *read* y *write*.
- *O_CREAT* para crear un archivo.
- *close* cerrar archivo.

Llamadas al sistema para gestión de archivos

- Muchas de las llamadas al sistema tienen relación con el sistema de archivos.
- Al abrir un archivo se genera una llamada: *O_RDONLY*, *O_WRONLY*, *O_RDWR*.
- Las más utilizadas son *read* y *write*.
- *O_CREAT* para crear un archivo.
- *close* cerrar archivo.
- La mayoría de programas leen y escriben en forma secuencial. Pero la llamada *Lseek* modifica el valor del apuntador y permite que las llamadas *read* o *write* lean o escriban en cualquier punto del archivo.

Contenido I

- 1 Conceptos básicos
- 2 Núcleo del Sistema operativo
 - Tipos
- 3 El intérprete de comandos
- 4 Características de un proceso
 - El bloque de control de proceso
 - Traza de un proceso
- 5 Ejecución de instrucciones
 - Interrupciones
 - Trap
 - Las llamadas al sistema
- 6 Planificadores**
- 7 Creación y Terminación de procesos
- 8 Estados
 - Modelos de estados de procesos
 - Swapping

Contenido II

9 Procesos en Windows y Ubuntu

- El planificador se encarga de establecer el orden de ejecución de los procesos.

Planificadores

- El planificador se encarga de establecer el orden de ejecución de los procesos.
- Utiliza un algoritmo de planificación.

Planificadores

- El planificador se encarga de establecer el orden de ejecución de los procesos.
- Utiliza un algoritmo de planificación.
- Un proceso necesita recursos para ejecutarse (tiempo de CPU, memoria, archivos, dispositivos E/S).

- El planificador se encarga de establecer el orden de ejecución de los procesos.
- Utiliza un algoritmo de planificación.
- Un proceso necesita recursos para ejecutarse (tiempo de CPU, memoria, archivos, dispositivos E/S).
- Estos recursos son asignados cuando se crea el proceso o durante su ejecución.

Tipos de planificadores

- **A largo plazo** entran en funcionamiento cuando un proceso finaliza o si no finaliza en un cierto tiempo. Trabajan con la cola de procesos por lote y admiten estos al sistema por lote.

Tipos de planificadores

- **A largo plazo** entran en funcionamiento cuando un proceso finaliza o si no finaliza en un cierto tiempo. Trabajan con la cola de procesos por lote y admiten estos al sistema por lote.
- **A medio plazo** se encargan de ordenar los procesos que están bloqueados y de insertarlos y quitarlos de la memoria para ponerlos en la cola de ejecutables.

Tipos de planificadores

- **A largo plazo** entran en funcionamiento cuando un proceso finaliza o si no finaliza en un cierto tiempo. Trabajan con la cola de procesos por lote y admiten estos al sistema por lote.
- **A medio plazo** se encargan de ordenar los procesos que están bloqueados y de insertarlos y quitarlos de la memoria para ponerlos en la cola de ejecutables.
- **A corto plazo** eligen que proceso va a ser asignado a la cola de CPU.

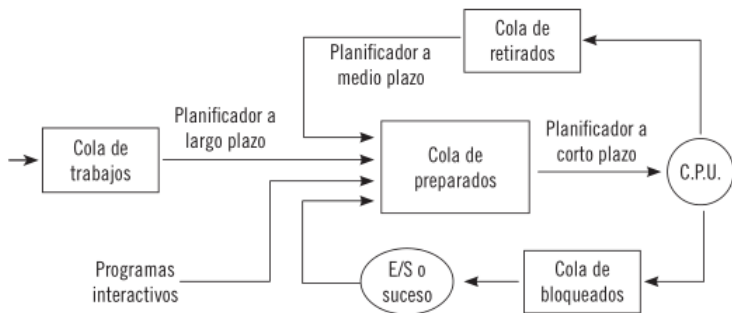
Tipos de planificadores

- **A largo plazo** entran en funcionamiento cuando un proceso finaliza o si no finaliza en un cierto tiempo. Trabajan con la cola de procesos por lote y admiten estos al sistema por lote.
- **A medio plazo** se encargan de ordenar los procesos que están bloqueados y de insertarlos y quitarlos de la memoria para ponerlos en la cola de ejecutables.
- **A corto plazo** eligen que proceso va a ser asignado a la cola de CPU.

Los SO antiguos no tenían planificadores a largo y medio plazo.

Los planificadores a largo y medio plazo regulan la carga del sistema.

Diagrama de planificadores



- **FIFO** atiende al primer proceso que llega y el resto entra en cola de espera. **Desventajas** Penaliza procesos cortos, tiempo de respuesta impredecible.

Algoritmos de planificación

- **FIFO** atiende al primer proceso que llega y el resto entra en cola de espera. **Desventajas** Penaliza procesos cortos, tiempo de respuesta impredecible.
- **Round-Robin** Se define un tiempo de ejecución para los procesos y se los atiende de manera circular por turnos. Si el proceso no ha finalizado en el intervalo se coloca al final de la lista.

Algoritmos de planificación

- **FIFO** atiende al primer proceso que llega y el resto entra en cola de espera. **Desventajas** Penaliza procesos cortos, tiempo de respuesta impredecible.
- **Round-Robin** Se define un tiempo de ejecución para los procesos y se los atiende de manera circular por turnos. Si el proceso no ha finalizado en el intervalo se coloca al final de la lista.
- **Short job first** (*Trabajo más corto primero*) atiende al proceso más corto de la cola. La CPU no dejará al proceso hasta que finalice.

Algoritmos de planificación

- **FIFO** atiende al primer proceso que llega y el resto entra en cola de espera. **Desventajas** Penaliza procesos cortos, tiempo de respuesta impredecible.
- **Round-Robin** Se define un tiempo de ejecución para los procesos y se los atiende de manera circular por turnos. Si el proceso no ha finalizado en el intervalo se coloca al final de la lista.
- **Short job first** (*Trabajo más corto primero*) atiende al proceso más corto de la cola. La CPU no dejará al proceso hasta que finalice.
- **Prioridad multinivel** Los procesos se asignan por prioridad en diferentes colas. El SO aplicará un determinado algoritmo a cada cola.

Algoritmos de planificación

- **FIFO** atiende al primer proceso que llega y el resto entra en cola de espera. **Desventajas** Penaliza procesos cortos, tiempo de respuesta impredecible.
- **Round-Robin** Se define un tiempo de ejecución para los procesos y se los atiende de manera circular por turnos. Si el proceso no ha finalizado en el intervalo se coloca al final de la lista.
- **Short job first** (*Trabajo más corto primero*) atiende al proceso más corto de la cola. La CPU no dejará al proceso hasta que finalice.
- **Prioridad multinivel** Los procesos se asignan por prioridad en diferentes colas. El SO aplicará un determinado algoritmo a cada cola.
- **High remainder next** la prioridad es marcada por el tiempo que está esperando en la cola y el tiempo necesario para su finalización.

Contenido I

- 1 Conceptos básicos
- 2 Núcleo del Sistema operativo
 - Tipos
- 3 El intérprete de comandos
- 4 Características de un proceso
 - El bloque de control de proceso
 - Traza de un proceso
- 5 Ejecución de instrucciones
 - Interrupciones
 - Trap
 - Las llamadas al sistema
- 6 Planificadores
- 7 Creación y Terminación de procesos
- 8 Estados
 - Modelos de estados de procesos
 - Swapping

Contenido II

9 Procesos en Windows y Ubuntu

Creación de procesos

- Cuando un nuevo proceso se va a añadir, el SO construye las estructuras de datos para manejar el proceso (BCP) y reserva el espacio de direcciones de memoria principal para el proceso.
- Existen cuatro eventos que llevan a la creación de un proceso:
 - Recepción y admisión de un nuevo trabajo.

Creación de procesos

- Cuando un nuevo proceso se va a añadir, el SO construye las estructuras de datos para manejar el proceso (BCP) y reserva el espacio de direcciones de memoria principal para el proceso.
- Existen cuatro eventos que llevan a la creación de un proceso:
 - Recepción y admisión de un nuevo trabajo.
 - La conexión del usuario provoca la creación de un proceso que ejecute el shell.

Creación de procesos

- Cuando un nuevo proceso se va a añadir, el SO construye las estructuras de datos para manejar el proceso (BCP) y reserva el espacio de direcciones de memoria principal para el proceso.
- Existen cuatro eventos que llevan a la creación de un proceso:
 - Recepción y admisión de un nuevo trabajo.
 - La conexión del usuario provoca la creación de un proceso que ejecute el shell.
 - Respuesta a una petición del usuario.

Creación de procesos

- Cuando un nuevo proceso se va a añadir, el SO construye las estructuras de datos para manejar el proceso (BCP) y reserva el espacio de direcciones de memoria principal para el proceso.
- Existen cuatro eventos que llevan a la creación de un proceso:
 - Recepción y admisión de un nuevo trabajo.
 - La conexión del usuario provoca la creación de un proceso que ejecute el shell.
 - Respuesta a una petición del usuario.
 - Un proceso solicita la creación de otro proceso.



Terminación de procesos

De igual manera existen varias razones que pueden llevar a la terminación de un proceso.

- Finalización normal.
- Límite de tiempo excedido (en ejecución).
- Límite de tiempo (en espera)
- Memoria no disponible.
- Violaciones de frontera: acceder a direcciones de memoria no permitidas.
- Error de protección: acceder a recursos no permitidos o utilizarlos de maneras no apropiadas (archivos de solo lectura).
- Error aritmético: división por 0, representación de números mayores a la capacidad del hardware.
- Fallo de E/S: fichero no encontrado.
- Instrucción no válida.
- Instrucción privilegiada: ejecución de instrucciones exclusivas al SO.

Terminación de procesos

- Uso inapropiado de datos.
- Intervención del operador por el SO.
- Terminación del proceso padre.
- Solicitud del proceso padre.

Contenido I

- 1 Conceptos básicos
- 2 Núcleo del Sistema operativo
 - Tipos
- 3 El intérprete de comandos
- 4 Características de un proceso
 - El bloque de control de proceso
 - Traza de un proceso
- 5 Ejecución de instrucciones
 - Interrupciones
 - Trap
 - Las llamadas al sistema
- 6 Planificadores
- 7 Creación y Terminación de procesos
- 8 Estados
 - Modelos de estados de procesos
 - Swapping

Contenido II

9 Procesos en Windows y Ubuntu

Estados de un proceso

Los procesos evolucionan de manera cíclica entre diferentes estados. Estos estados pueden ser (los más comunes):

- **Nuevo** procesos que acaban de ser incluidos y el SO aún no los admite para ser ejecutados.

Estados de un proceso

Los procesos evolucionan de manera cíclica entre diferentes estados. Estos estados pueden ser (los más comunes):

- **Nuevo** procesos que acaban de ser incluidos y el SO aún no los admite para ser ejecutados.
- **Listo** procesos que cuentan con todos los recursos para comenzar o seguir su ejecución.

Estados de un proceso

Los procesos evolucionan de manera cíclica entre diferentes estados. Estos estados pueden ser (los más comunes):

- **Nuevo** procesos que acaban de ser incluidos y el SO aún no los admite para ser ejecutados.
- **Listo** procesos que cuentan con todos los recursos para comenzar o seguir su ejecución.
- **Ejecución** proceso que tiene el control del procesador. En sistemas con un solo procesador solo un proceso puede estar en este estado.

Estados de un proceso

Los procesos evolucionan de manera cíclica entre diferentes estados. Estos estados pueden ser (los más comunes):

- **Nuevo** procesos que acaban de ser incluidos y el SO aún no los admite para ser ejecutados.
- **Listo** procesos que cuentan con todos los recursos para comenzar o seguir su ejecución.
- **Ejecución** proceso que tiene el control del procesador. En sistemas con un solo procesador solo un proceso puede estar en este estado.
- **Bloqueado** procesos en espera de algún recurso o evento para continuar su ejecución.

Estados de un proceso

Los procesos evolucionan de manera cíclica entre diferentes estados. Estos estados pueden ser (los más comunes):

- **Nuevo** procesos que acaban de ser incluidos y el SO aún no los admite para ser ejecutados.
- **Listo** procesos que cuentan con todos los recursos para comenzar o seguir su ejecución.
- **Ejecución** proceso que tiene el control del procesador. En sistemas con un solo procesador solo un proceso puede estar en este estado.
- **Bloqueado** procesos en espera de algún recurso o evento para continuar su ejecución.
- **Terminado** procesos excluidos por el SO del grupo de procesos ejecutables. Un proceso pasa a terminado:

Estados de un proceso

Los procesos evolucionan de manera cíclica entre diferentes estados. Estos estados pueden ser (los más comunes):

- **Nuevo** procesos que acaban de ser incluidos y el SO aún no los admite para ser ejecutados.
- **Listo** procesos que cuentan con todos los recursos para comenzar o seguir su ejecución.
- **Ejecución** proceso que tiene el control del procesador. En sistemas con un solo procesador solo un proceso puede estar en este estado.
- **Bloqueado** procesos en espera de algún recurso o evento para continuar su ejecución.
- **Terminado** procesos excluidos por el SO del grupo de procesos ejecutables. Un proceso pasa a terminado:
 - terminación normal

Estados de un proceso

Los procesos evolucionan de manera cíclica entre diferentes estados. Estos estados pueden ser (los más comunes):

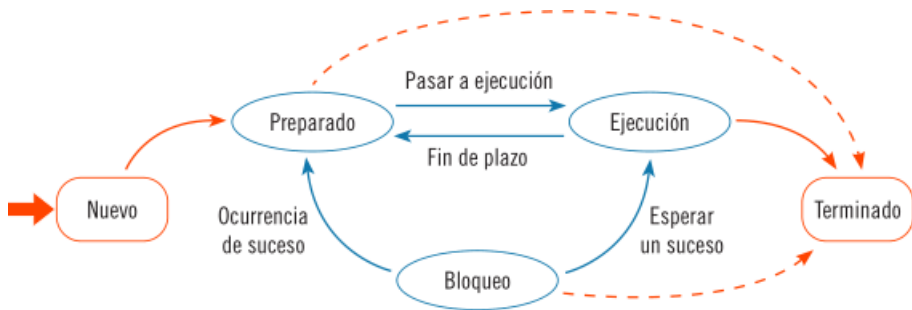
- **Nuevo** procesos que acaban de ser incluidos y el SO aún no los admite para ser ejecutados.
- **Listo** procesos que cuentan con todos los recursos para comenzar o seguir su ejecución.
- **Ejecución** proceso que tiene el control del procesador. En sistemas con un solo procesador solo un proceso puede estar en este estado.
- **Bloqueado** procesos en espera de algún recurso o evento para continuar su ejecución.
- **Terminado** procesos excluidos por el SO del grupo de procesos ejecutables. Un proceso pasa a terminado:
 - terminación normal
 - error irrecuperable

Estados de un proceso

Los procesos evolucionan de manera cíclica entre diferentes estados. Estos estados pueden ser (los más comunes):

- **Nuevo** procesos que acaban de ser incluidos y el SO aún no los admite para ser ejecutados.
- **Listo** procesos que cuentan con todos los recursos para comenzar o seguir su ejecución.
- **Ejecución** proceso que tiene el control del procesador. En sistemas con un solo procesador solo un proceso puede estar en este estado.
- **Bloqueado** procesos en espera de algún recurso o evento para continuar su ejecución.
- **Terminado** procesos excluidos por el SO del grupo de procesos ejecutables. Un proceso pasa a terminado:
 - terminación normal
 - error irrecuperable
 - terminación por otro proceso autorizado.

Estados de un proceso



Contenido I

- 1 Conceptos básicos
- 2 Núcleo del Sistema operativo
 - Tipos
- 3 El intérprete de comandos
- 4 Características de un proceso
 - El bloque de control de proceso
 - Traza de un proceso
- 5 Ejecución de instrucciones
 - Interrupciones
 - Trap
 - Las llamadas al sistema
- 6 Planificadores
- 7 Creación y Terminación de procesos
- 8 Estados
 - Modelos de estados de procesos
 - Swapping
- 9 Procesos en Windows y Ubuntu

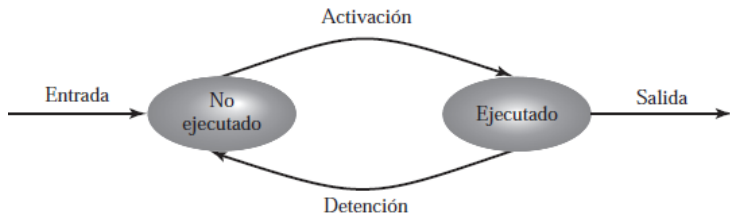
Modelos de estados de procesos

- La responsabilidad principal del SO es controlar la ejecución de los procesos, por lo tanto debe determinar el patrón de entrelazado para la ejecución y asignación de recursos a estos procesos.
- El primer paso para el diseño de un SO es describir el comportamiento que se desea que tengan los procesos.
- A continuación veremos dos modelos de estados de procesos: un modelo de dos estados y uno de cinco estados.

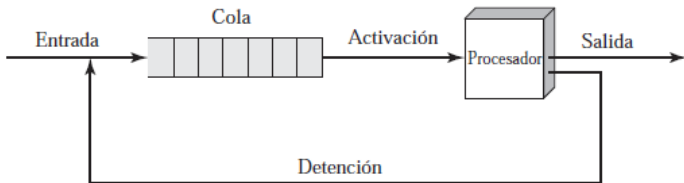
Modelo de procesos de dos estados

- Si tenemos en cuenta que en un instante dado un proceso está siendo **ejecutado o no**, entonces podemos decir que un procesos estará en dos posibles estados **Ejecutando** y **No Ejecutando**.
- Cuando el SO, crea un proceso, crea un BCP para este nuevo proceso e inserta el proceso en el sistema en estado *No Ejecutando*. Entonces, en este punto el proceso ya existe, es conocido por el SO pero está esperando su oportunidad de ser ejecutado.
- Cada cierto tiempo el proceso actualmente en ejecución se interrumpirá y pasará al estado *No Ejecutando*.
- El **activador** seleccionará otro proceso para ejecutar y este pasará al estado *Ejecutando*.
- Los procesos en estado *No Ejecutando* deben ser almacenados en una especie de cola que contiene punteros a los BCP de cada proceso en este estado.

Modelo de procesos de dos estados



(a) Diagrama de transiciones de estados



(b) Modelos de colas

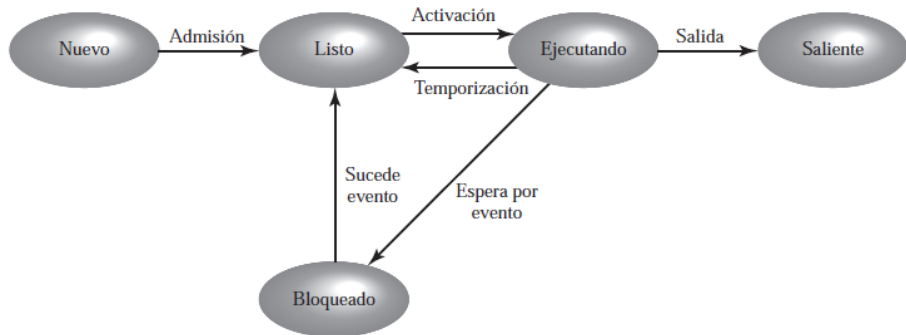
Modelo de procesos de dos estados

- Este modelo de gestión de colas sería efectiva si todos los procesos estuviesen siempre preparados para ejecutar.
- Esta cola es de tipo FIFO por lo tanto no toma en cuenta que procesos llevan más tiempo en la cola o que procesos están bloqueados esperando alguna operación de E/S.
- Por esta razón este modelo de dos estados no es eficiente.

Modelo de procesos de cinco estados

- Un modelo más eficiente podría dividir el estado *No Ejecutando* en dos: *Listo* y *Bloqueado*.
- Este modelo añade también un estado *Nuevo* y otro *Saliente*.
- Entonces los cinco estados son:
 - Ejecutando
 - Listo
 - Bloqueado (en espera de evento u operación E/S)
 - Nuevo (BCP creado pero aún no cargado en memoria principal).
 - Saliente (proceso detenido o abortado)

Modelo de procesos de cinco estados



Modelo de procesos de cinco estados

Posibles transiciones entre estados

Las posibles transiciones entre los 5 estados son:

- Null \rightarrow Nuevo

Modelo de procesos de cinco estados

Posibles transiciones entre estados

Las posibles transiciones entre los 5 estados son:

- Null \rightarrow Nuevo
- Nuevo \rightarrow Listo

Modelo de procesos de cinco estados

Posibles transiciones entre estados

Las posibles transiciones entre los 5 estados son:

- Null \rightarrow Nuevo
- Nuevo \rightarrow Listo
- Listo \rightarrow Ejecutando

Modelo de procesos de cinco estados

Posibles transiciones entre estados

Las posibles transiciones entre los 5 estados son:

- Null \rightarrow Nuevo
- Nuevo \rightarrow Listo
- Listo \rightarrow Ejecutando
- Ejecutando \rightarrow Saliente

Modelo de procesos de cinco estados

Posibles transiciones entre estados

Las posibles transiciones entre los 5 estados son:

- Null \rightarrow Nuevo
- Nuevo \rightarrow Listo
- Listo \rightarrow Ejecutando
- Ejecutando \rightarrow Saliente
- Ejecutando \rightarrow Listo

Modelo de procesos de cinco estados

Posibles transiciones entre estados

Las posibles transiciones entre los 5 estados son:

- Null \rightarrow Nuevo
- Nuevo \rightarrow Listo
- Listo \rightarrow Ejecutando
- Ejecutando \rightarrow Saliente
- Ejecutando \rightarrow Listo
- Ejecutando \rightarrow Bloqueado

Modelo de procesos de cinco estados

Posibles transiciones entre estados

Las posibles transiciones entre los 5 estados son:

- Null \rightarrow Nuevo
- Nuevo \rightarrow Listo
- Listo \rightarrow Ejecutando
- Ejecutando \rightarrow Saliente
- Ejecutando \rightarrow Listo
- Ejecutando \rightarrow Bloqueado
- Bloqueado Listo \rightarrow Nuevo

Modelo de procesos de cinco estados

Posibles transiciones entre estados

Las posibles transiciones entre los 5 estados son:

- Null \rightarrow Nuevo
- Nuevo \rightarrow Listo
- Listo \rightarrow Ejecutando
- Ejecutando \rightarrow Saliente
- Ejecutando \rightarrow Listo
- Ejecutando \rightarrow Bloqueado
- Bloqueado Listo \rightarrow Nuevo
- Listo \rightarrow Saliente

Modelo de procesos de cinco estados

Posibles transiciones entre estados

Las posibles transiciones entre los 5 estados son:

- Null \rightarrow Nuevo
- Nuevo \rightarrow Listo
- Listo \rightarrow Ejecutando
- Ejecutando \rightarrow Saliente
- Ejecutando \rightarrow Listo
- Ejecutando \rightarrow Bloqueado
- Bloqueado Listo \rightarrow Nuevo
- Listo \rightarrow Saliente
- Bloqueado \rightarrow Saliente

Modelo de procesos de cinco estados

- En este modelo los estados *Listo* y *Bloqueado* son manejados por colas independientes.

Modelo de procesos de cinco estados

- En este modelo los estados *Listo* y *Bloqueado* son manejados por colas independientes.
- Cuando el SO debe seleccionar el siguiente proceso a ejecutar busca en la cola de *Listos*. Podría existir un esquema de prioridades o simplemente un esquema FIFO.

Modelo de procesos de cinco estados

- En este modelo los estados *Listo* y *Bloqueado* son manejados por colas independientes.
- Cuando el SO debe seleccionar el siguiente proceso a ejecutar busca en la cola de *Listos*. Podría existir un esquema de prioridades o simplemente un esquema FIFO.
- Si se maneja un esquema de prioridades, sería conveniente tener varias colas de procesos listos, una por cada prioridad. Así el SO podría seleccionar el procesos listo de mayor prioridad más rápidamente.

Modelo de procesos de cinco estados

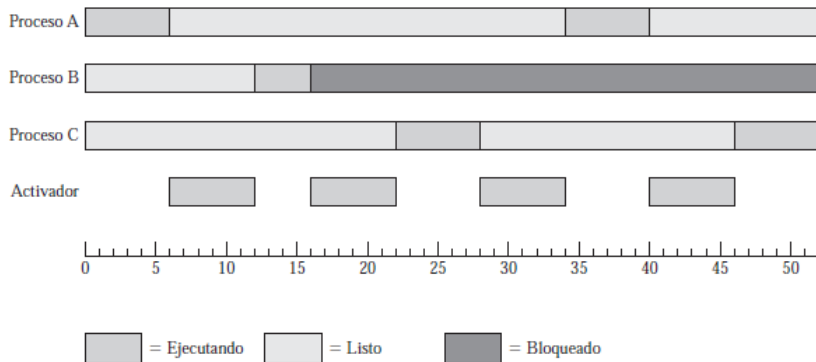
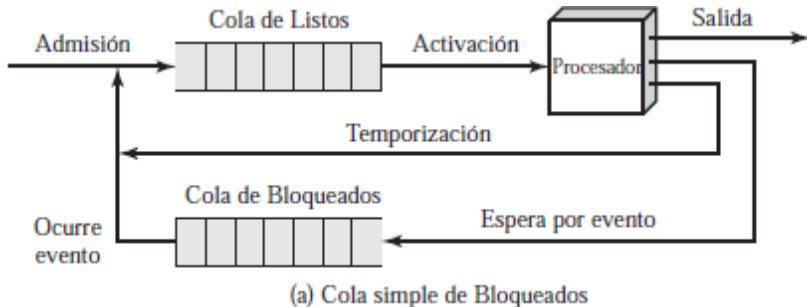


Figura: Estados de los procesos A, B y C

Modelos de procesos de cinco estados

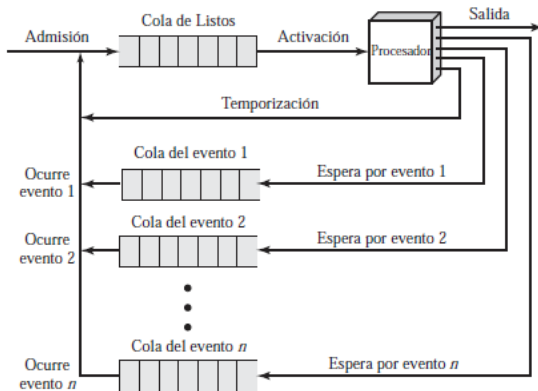
Modelo de cola simple



Cuando sucede algún evento el SO debe recorrer la cola de *Bloqueados* en busca de los procesos que estén esperando por este evento.

Modelos de procesos de cinco estados

Modelo de múltiples colas de bloqueados



(b) Múltiples colas de Bloqueados

En algunos SO las colas de procesos *Bloqueados* podría contener miles de procesos por lo que sería más eficiente manejar una cola por cada evento.

Contenido I

- 1 Conceptos básicos
- 2 Núcleo del Sistema operativo
 - Tipos
- 3 El intérprete de comandos
- 4 Características de un proceso
 - El bloque de control de proceso
 - Traza de un proceso
- 5 Ejecución de instrucciones
 - Interrupciones
 - Trap
 - Las llamadas al sistema
- 6 Planificadores
- 7 Creación y Terminación de procesos
- 8 Estados
 - Modelos de estados de procesos
 - Swapping
- 9 Procesos en Windows y Ubuntu

Swapping (procesos suspendidos)

- El modelo de múltiples colas de procesos *Bloqueados* aumenta la eficiencia del sistema, sin embargo la diferencia entre la velocidad del procesador y los dispositivos E/S es tan grande que es muy probable que todos los procesos lleguen a estar en un estado *Bloqueado*.

Swapping (procesos suspendidos)

- El modelo de múltiples colas de procesos *Bloqueados* aumenta la eficiencia del sistema, sin embargo la diferencia entre la velocidad del procesador y los dispositivos E/S es tan grande que es muy probable que todos los procesos lleguen a estar en un estado *Bloqueado*.
- Si tenemos miles de procesos en estado *Bloqueado*, estos podrían estar ocupando todo el espacio en memoria principal y el procesador estaría ocioso debido a que no tendríamos capacidad para aceptar nuevos procesos.

Swapping (procesos suspendidos)

- El modelo de múltiples colas de procesos *Bloqueados* aumenta la eficiencia del sistema, sin embargo la diferencia entre la velocidad del procesador y los dispositivos E/S es tan grande que es muy probable que todos los procesos lleguen a estar en un estado *Bloqueado*.
- Si tenemos miles de procesos en estado *Bloqueado*, estos podrían estar ocupando todo el espacio en memoria principal y el procesador estaría ocioso debido a que no tendríamos capacidad para aceptar nuevos procesos.
- Es cierto que podríamos aumentar la memoria pero esto también aumentaría el costo del sistema. Y es importante notar que debido a las grandes memorias disponibles, la tendencia es a ejecutar **procesos más grandes y no más procesos**.

Swapping (procesos suspendidos)

- Una solución para lograr aceptar más procesos y disminuir el tiempo ocioso del procesador es utilizar una técnica llamada *swapping* (memoria de intercambio).

Swapping (procesos suspendidos)

- Una solución para lograr aceptar más procesos y disminuir el tiempo ocioso del procesador es utilizar una técnica llamada *swapping* (memoria de intercambio).
- El swapping consiste en intercambiar parte o todo el proceso a de memoria principal a disco.

Swapping (procesos suspendidos)

- Una solución para lograr aceptar más procesos y disminuir el tiempo ocioso del procesador es utilizar una técnica llamada *swapping* (memoria de intercambio).
- El swapping consiste en intercambiar parte o todo el proceso a de memoria principal a disco.
- Si no existen procesos en memoria principal en estado *Listo*, el SO mueve uno de estos procesos *Bloqueados* a disco (a la cola de procesos *Suspendidos*).

Swapping (procesos suspendidos)

- Una solución para lograr aceptar más procesos y disminuir el tiempo ocioso del procesador es utilizar una técnica llamada *swapping* (memoria de intercambio).
- El swapping consiste en intercambiar parte o todo el proceso a de memoria principal a disco.
- Si no existen procesos en memoria principal en estado *Listo*, el SO mueve uno de estos procesos *Bloqueados* a disco (a la cola de procesos *Suspendidos*).
- El swapping añade el estado *Suspendido*

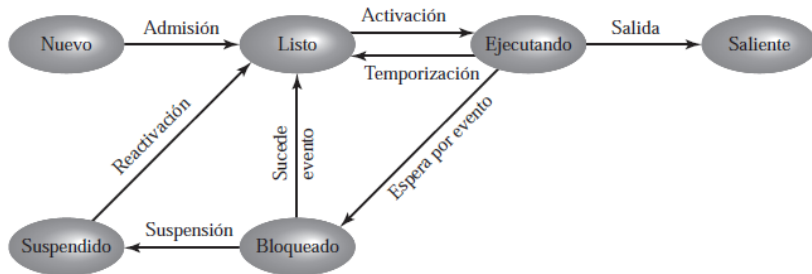
Swapping (procesos suspendidos)

- Una solución para lograr aceptar más procesos y disminuir el tiempo ocioso del procesador es utilizar una técnica llamada *swapping* (memoria de intercambio).
- El swapping consiste en intercambiar parte o todo el proceso a de memoria principal a disco.
- Si no existen procesos en memoria principal en estado *Listo*, el SO mueve uno de estos procesos *Bloqueados* a disco (a la cola de procesos *Suspendidos*).
- El swapping añade el estado *Suspendido*
- Así el SO puede continuar con la creación de nuevos procesos o traer procesos de la cola de *Suspendidos* que puedan continuar siendo ejecutados.

Swapping (procesos suspendidos)

- Una solución para lograr aceptar más procesos y disminuir el tiempo ocioso del procesador es utilizar una técnica llamada *swapping* (memoria de intercambio).
- El swapping consiste en intercambiar parte o todo el proceso a de memoria principal a disco.
- Si no existen procesos en memoria principal en estado *Listo*, el SO mueve uno de estos procesos *Bloqueados* a disco (a la cola de procesos *Suspendidos*).
- El swapping añade el estado *Suspendido*
- Así el SO puede continuar con la creación de nuevos procesos o traer procesos de la cola de *Suspendidos* que puedan continuar siendo ejecutados.
- Esta técnica generalmente mejora el rendimiento del sistema.

Swapping (procesos suspendidos)



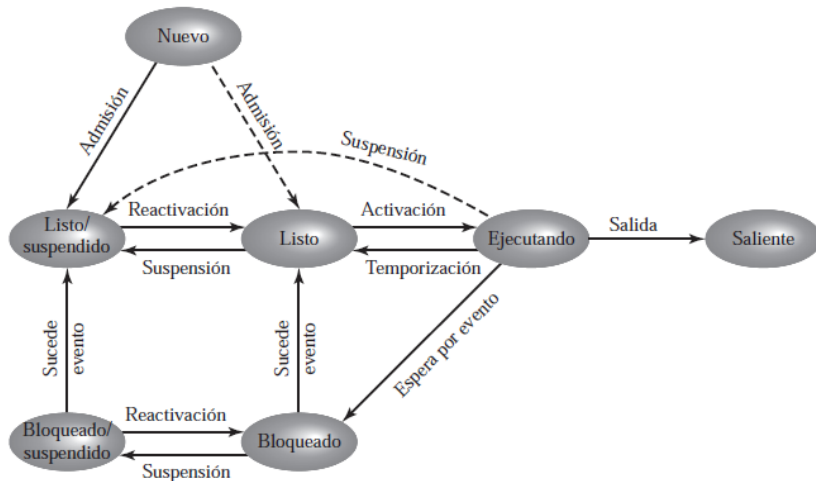
Modelo de estados con swapping

Para que el SO pueda determinar más fácilmente que proceso debe traer de la cola de *Suspendidos* se necesitan los siguientes estados:

- Listo
- Bloqueado
- Bloqueado/Suspendido
- Listo/Suspendido

Modelo de estados con swapping

Utilizando dos estados suspendido



Transiciones con swapping (dos estados suspendidos)

La figura anterior nos muestra las posibles transiciones entre estados cuando se utilizan dos estados suspendidos. Estas transiciones son:

- **Bloqueado** → **Bloqueado/Suspendido**
- **Bloqueado/Suspendido** → **Listo/Suspendido**
- **Listo/Suspendido** → **Listo**
- **Listo** → **Listo/Suspendido** (liberar memoria o prioridad baja)
- **Nuevo** → **Listo/Suspendido** o **Listo** (depende espacio en memoria)
- **Bloqueado/Suspendido** → **Bloqueado** (prioridad alta o evento pronto a ocurrir)
- **Ejecutando** → **Listo/Suspendido** (liberar memoria)
- **Cualquier estado** → **Saliente**

Contenido I

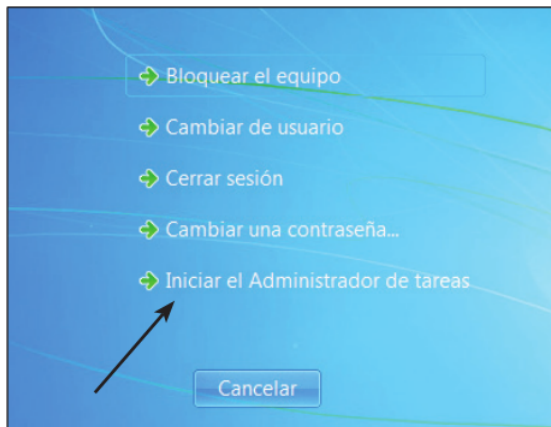
- 1 Conceptos básicos
- 2 Núcleo del Sistema operativo
 - Tipos
- 3 El intérprete de comandos
- 4 Características de un proceso
 - El bloque de control de proceso
 - Traza de un proceso
- 5 Ejecución de instrucciones
 - Interrupciones
 - Trap
 - Las llamadas al sistema
- 6 Planificadores
- 7 Creación y Terminación de procesos
- 8 Estados
 - Modelos de estados de procesos
 - Swapping

Contenido II

9 Procesos en Windows y Ubuntu

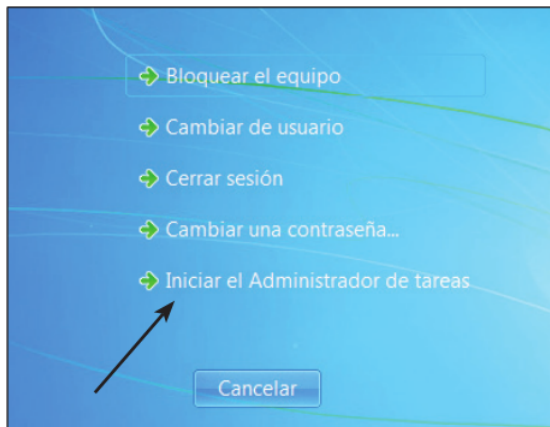
Acceder a procesos en Windows 7

- ¿Cómo accedemos a ver los procesos ejecutándose en Windows 7?



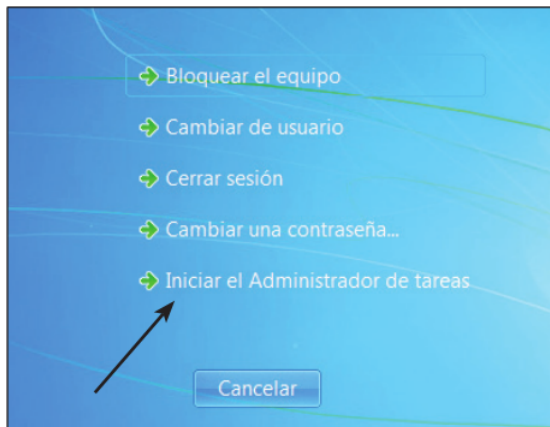
Acceder a procesos en Windows 7

- ¿Cómo accedemos a ver los procesos ejecutándose en Windows 7?
ctrl + alt + supr **Administrador de tareas**.



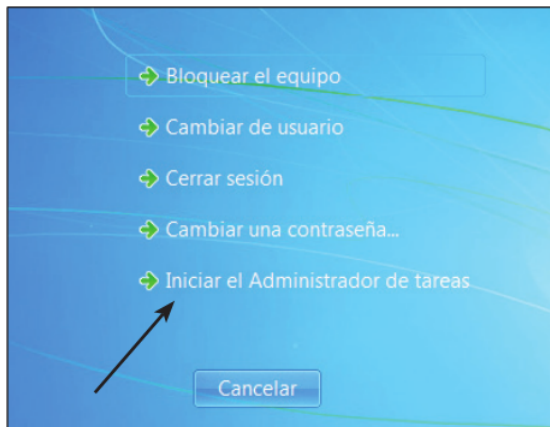
Acceder a procesos en Windows 7

- ¿Cómo accedemos a ver los procesos ejecutándose en Windows 7?
ctrl + alt + supr **Administrador de tareas**.
- ¿Otras opciones en Windows 7?



Acceder a procesos en Windows 7

- ¿Cómo accedemos a ver los procesos ejecutándose en Windows 7?
ctrl + alt + supr **Administrador de tareas**.
- ¿Otras opciones en Windows 7?
- ¿Opciones en Windows 8?



Procesos en Windows

Administrador de tareas de Windows

Archivo Opciones Ver Ayuda

Aplicaciones **Procesos** Servicios Rendimiento Funciones de red Usuarios

Nombre de imagen	Nombre de usuario	CPU	Memoria ...	Descripción
firefox.exe	yo mismo	02	276.556 KB	Firefox
AcroRd32.exe	yo mismo	00	93.956 KB	Adobe Reader
chrome.exe	yo mismo	00	38.688 KB	Google Chrome
chrome.exe	yo mismo	00	36.980 KB	Google Chrome
wlmail.exe	yo mismo	00	34.076 KB	Windows Live Mail
explorer.exe	yo mismo	00	32.392 KB	Explorador de Windows
plugin-container.exe	yo mismo	00	29.764 KB	Plugin Container for Firefox
WINWORD.EXE	yo mismo	05	18.732 KB	Microsoft Office Word
wlcomm.exe	yo mismo	00	14.684 KB	Windows Live Communications Platform
soffice.bin	yo mismo	00	7.228 KB	OpenOffice.org 3.3
mspaint.exe	yo mismo	00	5.212 KB	Paint
sidebar.exe	yo mismo	00	4.836 KB	Gadgets de escritorio de Windows
AcroRd32.exe	yo mismo	00	2.732 KB	Adobe Reader
msnmsgr.exe	yo mismo	00	1.604 KB	Windows Live Messenger
taskmgr.exe	yo mismo	16	1.560 KB	Administrador de tareas de Windows
taskhost.exe	yo mismo	00	1.236 KB	Proceso de host para tareas de Windows
SweetIM.exe	yo mismo	00	1.232 KB	SweetIM Instant Messenger Enhancer
jucheck.exe	yo mismo	00	1.136 KB	Java(TM) Update Checker
mssecos.exe	yo mismo	00	1.052 KB	Microsoft Security Client User Interface
csrss.exe	SYSTEM	06	992 KB	Proceso en tiempo de ejecución del cliente-servidor
wuauclt.exe	yo mismo	00	620 KB	Windows Update
winlogon.exe	SYSTEM	00	620 KB	Aplicación de inicio de sesión de Windows
jucheck.exe	yo mismo	00	604 KB	Java(TM) Update Scheduler

☐ Mostrar procesos de todos los usuarios

Finalizar proceso

Procesos: 64 Uso de CPU: 30% Memoria física: 58%

- La pestaña de procesos nos mostrará:

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria
 - descripción

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria
 - descripción
 - clic derecho en la cabecera permite seleccionar que aspectos mostrar.

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria
 - descripción
 - clic derecho en la cabecera permite seleccionar que aspectos mostrar.
 - Total de procesos activos (inferior izquierda)

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria
 - descripción
 - clic derecho en la cabecera permite seleccionar que aspectos mostrar.
 - Total de procesos activos (inferior izquierda)
 - clic derecho sobre proceso mostrará:

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria
 - descripción
 - clic derecho en la cabecera permite seleccionar que aspectos mostrar.
 - Total de procesos activos (inferior izquierda)
 - clic derecho sobre proceso mostrará:
 - Terminar proceso

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria
 - descripción
 - clic derecho en la cabecera permite seleccionar que aspectos mostrar.
 - Total de procesos activos (inferior izquierda)
 - clic derecho sobre proceso mostrará:
 - Terminar proceso
 - Establecer prioridad

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria
 - descripción
 - clic derecho en la cabecera permite seleccionar que aspectos mostrar.
 - Total de procesos activos (inferior izquierda)
 - clic derecho sobre proceso mostrará:
 - Terminar proceso
 - Establecer prioridad
 - Buscar en línea

- La pestaña de procesos nos mostrará:
 - los procesos en ejecución.
 - el usuario propietario
 - el trabajo del procesador
 - la cantidad de memoria
 - descripción
 - clic derecho en la cabecera permite seleccionar que aspectos mostrar.
 - Total de procesos activos (inferior izquierda)
 - clic derecho sobre proceso mostrará:
 - Terminar proceso
 - Establecer prioridad
 - Buscar en línea
 - Propiedades

- ¿Cómo abrimos el administrador de procesos en Ubuntu?

- ¿Cómo abrimos el administrador de procesos en Ubuntu? Varias opciones: Información similar que en Win.

- ¿Cómo abrimos el administrador de procesos en Ubuntu? Varias opciones: Información similar que en Win.
 - Buscar: Monitor del sistema

- ¿Cómo abrimos el administrador de procesos en Ubuntu? Varias opciones: Información similar que en Win.
 - Buscar: Monitor del sistema
 - Buscar/Icono de aplicaciones/monitor del sistema

Procesos en Ubuntu

- ¿Cómo abrimos el administrador de procesos en Ubuntu? Varias opciones: Información similar que en Win.
 - Buscar: Monitor del sistema
 - Buscar/Icono de aplicaciones/monitor del sistema
 - Instalar “indicador de carga del sistema”



Monitor del sistema

Procesos Recursos Sistemas de archivos

Carga media para los últimos 1, 5 y 15 minutos: 0,63, 0,69, 0,62

Actualizar Ver ▾

Nombre del proceso	Usuario	% CPU ▲	ID	Memoria	Prioridad
compiz	chalo	22	2006	170,3 MiB	Normal
gnome-system-monitor	chalo	10	3706	13,2 MiB	Normal
indicator-multiloader	chalo	1	3677	6,1 MiB	Normal
unity-scope-loader	chalo	0	3648	4,9 MiB	Normal
unity_picasa_daemon.py	chalo	0	3594	7,5 MiB	Normal
unity_shotwell_daemon.py	chalo	0	3593	5,2 MiB	Normal
unity_facebook_daemon.py	chalo	0	3590	7,1 MiB	Normal
unity_flickr_daemon.py	chalo	0	3589	7,4 MiB	Normal
unity-video-lens-daemon	chalo	0	3575	2,8 MiB	Normal
bash	chalo	0	3392	1,6 MiB	Normal
ubuntu-geoip-provider	chalo	0	2621	1,1 MiB	Normal
geoclue-master	chalo	0	2617	920,0 KiB	Normal
gvfsd-metadata	chalo	0	2570	392,0 KiB	Normal
deia-dup-monitor	chalo	0	2430	3,1 MiB	Normal

Finalizar proceso