

**SC- 203 Fundamentos de Sistemas  
Operativos  
Clase 02**

---



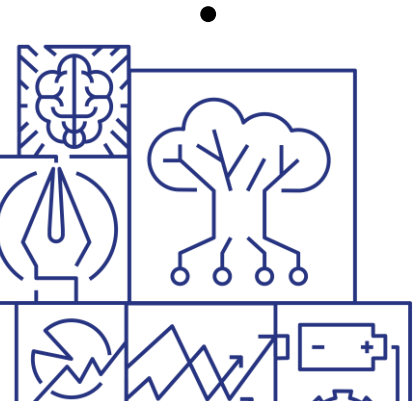
# Agenda

- Unidad 1: Aspectos generales de sistemas operativos
- Unidad 2: Procesos Parte 1

## 2.1 Descripción y Control de Procesos

### 2.1.1 Multiprocesos y Jerarquía de procesos

### 2.1.2 Control de Procesos



# Unidad 1

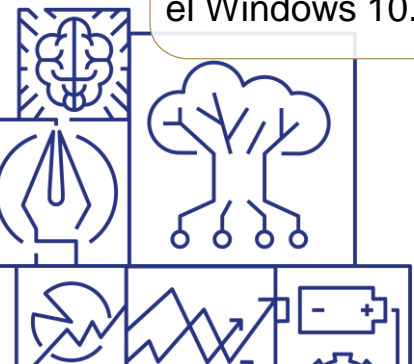
- **Unidad 1: Aspectos generales de sistemas operativos**
  - 1.2 Reconocimiento de los principales Sistemas Operativos: Windows, Linux, iOS, Android

## **1.2 Reconocimiento de los principales Sistemas Operativos: Windows, Linux, iOS, Android**

# Windows

Windows nació **como tal en 1975 por William H. Gates III y Paul Allen**, quienes decidieron programar por medio de una computadora PDP-10 de Digital Equipment Corporation.

Posteriormente se formó el MS-MOD Windows con una arquitectura un poco diferente, y con mejoras importantes dando lugar a que después del 2000 se crearan nuevas versiones hasta hoy en día que existe el Windows 10.



# Unix

Nació específicamente en 1969, creado por **Ken Thompson, Dennis Ritchie y Rudd Canaday, entre otros**. Inicialmente llamado UNICS, pero por ser fallido se modificó a UNIX.

GNU/LINUX, fue reescrito desde cero con el objetivo de una arquitectura totalmente libre, que se podía adaptar a dispositivos y computadoras.

En 1980 Microsoft crea versión llamada Xenix hasta 1989 que crear DOS, Apple en 2001 se apoyó de UNIX dando luz a Mac OS X. Con esto podemos concluir que **UNIX es la madre de los sistemas operativos**

# Linux

Se implementó con la idea de un sistema libre y abierto para que cualquier persona pueda acceder a la escritura del mismo y modificar su estructura según su gusto.

En 1983 que Richard Stallman creó por primera vez el proyecto GNU, todo ideado a partir de una estructura de UNIX, y compatible con POSIX, para dos años después dejar de ser un simple proyecto, y pasar a ser una fundación de software libre.

Primera versión nunca se publicó, en **1991** se lanza segunda versión (Linux 0,02), luego se lanza 0.03 esta y todas las versiones hasta la 0,10 fueron saltadas por fracasos y pruebas. En **1992** incrementa versiones hasta la 0.95, y debido a la aceptación se cuenta con las versiones actuales.

# iOS

Es uno de los más Jóvenes, su primer lanzamiento fue a mediados del 2001, a pesar de que se le otorga el nombre de MAC Sistema Operativo, gracias a la **versión Mac OS X lanzada en 1984**.

Tiempo después Apple **compró la compañía (1997)**, liberándose su verdadera y primera versión en 1999, **conocida como Mac OS X Server 1.0**, que fue lanzada a competir en el mercado en marzo de 2001, y que realmente era la versión número 10.

Su crecimiento **no fue muy fácil**, porque partes de su sistema eran un poco más complicados y costosos que los otros, **además que por ser uno de los más jóvenes compitió hasta hoy en día contra Microsoft y**

# Android

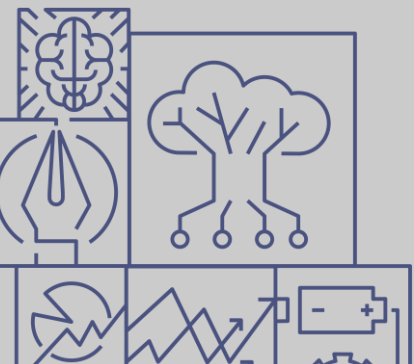
Creado por **Android Inc.**, una compañía adquirida por **Google en 2005**, Android se basa en **Linux**, un programa libre que, a su vez, está basado en **Unix**. El objetivo inicial de Android, de este modo, fue promover los estándares abiertos en teléfonos y computadoras (ordenadores) móviles.

Dada la gran cantidad de dispositivos equipados con Android, ya es posible encontrar más de **un millón de aplicaciones** que utilizan este sistema operativo para su funcionamiento. Android también se destaca por su **seguridad**, ya que los expertos han detectado **pocas vulnerabilidades** en su estructura.

# Unidad 2

- **Unidad 2: Procesos Parte 1**
- 2.1 Descripción y Control de Procesos
- 2.1.1 Multiprocesos y Jerarquía de procesos
- 2.1.2 Control de Procesos

## **2.1 Descripción y Control de Procesos**





# Definición de Procesos



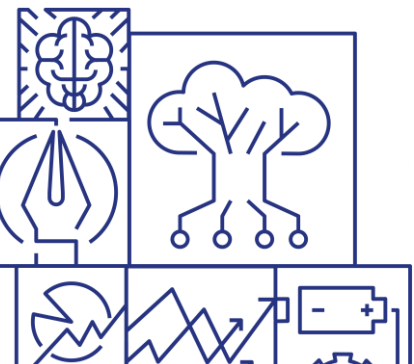
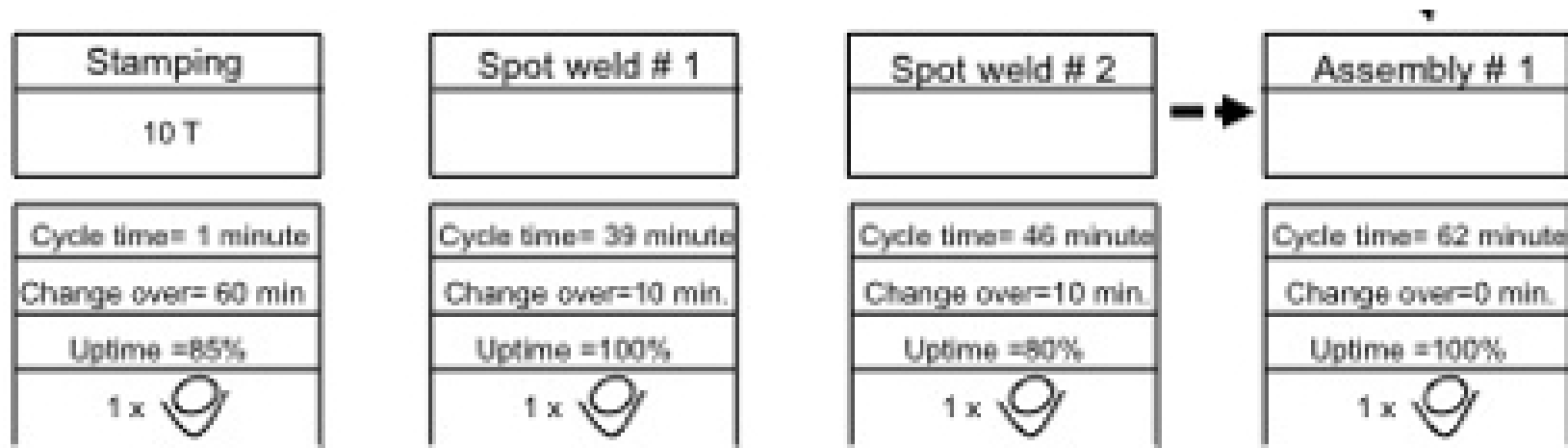
- Es una abstracción que hace referencia a cada caso de ejecución de un programa.
- No tiene por qué estar siempre en ejecución.
- La vida de un proceso pasa por varias fases, incluyendo la ejecución.

# Una Analogía

- Receta de un pastel (*el programa*), ingredientes (*los datos*), cocinero (*la CPU*), la mesa de cocina (*la memoria*), cada pastel (*un proceso*), el horno (*un dispositivo E/S*).
- ¡Nuestro chef puede tener varios pasteles a medio hacer!



- Cuando hay varias CPUs podemos tener ejecución paralela. Pero cada CPU sólo puede ejecutar un proceso a la vez. Normalmente número de procesos > número de CPUs.
- Cada proceso tiene un **UID** (identificador del usuario) del usuario que lo lanzó.
- Los procesos hijos tienen el identificador del padre.
- Hay usuarios que tienen un identificador de grupo(**GID**).



# Tipos de Procesos

## Acceso a CPU y Recursos

**Apropiativos:** Solo abandonan los recursos de manera voluntaria.

**No Apropiativos:** Permiten a otros procesos apropiarse de sus recursos.

## Permanencia en Memoria

**Residentes:** Permanecen en memoria durante toda su evolución

**Intercambiables:** SO puede llevar a disco en cualquier momento al proceso. (**Swappable**)

## Nivel de privilegios

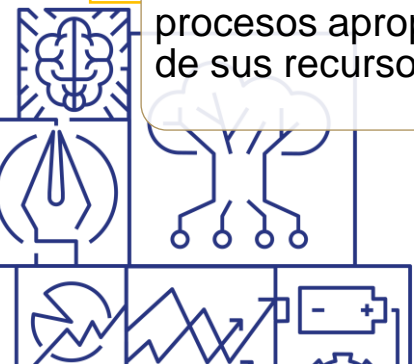
**Privilegiado:** se ejecutan en modo supervisor.

**No Privilegiado:** Los que ejecuta el usuario normalmente.

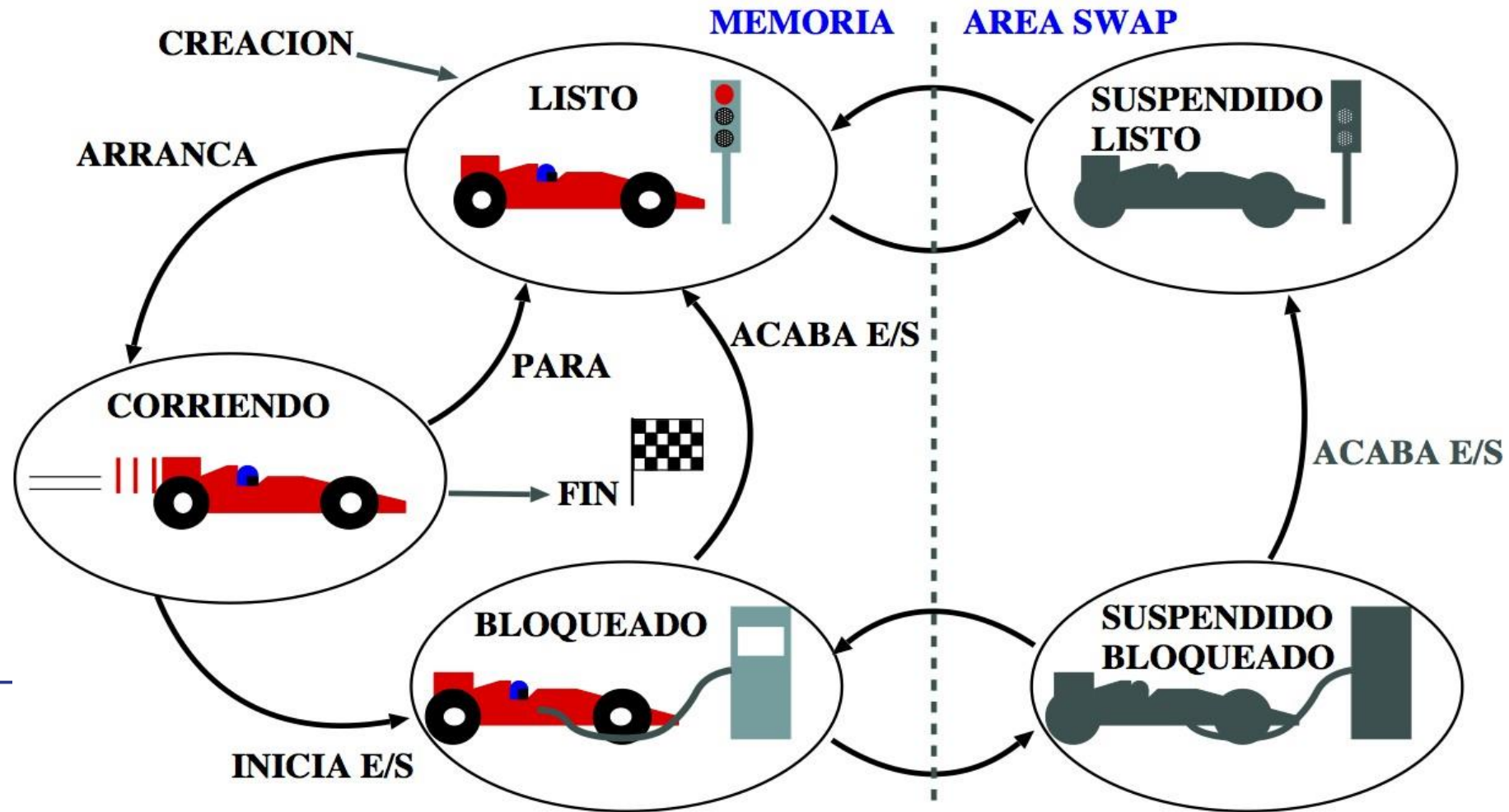
## Propietario

**De Usuario:** Diseñados por el usuario, ejecución en modo no protegido.

**De Sistema:** forman parte del SO (E/S, planificación de otros procesos, etc)



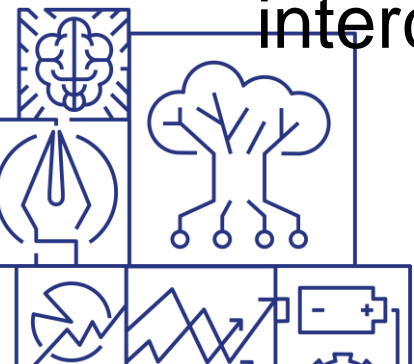
# Estados de un proceso



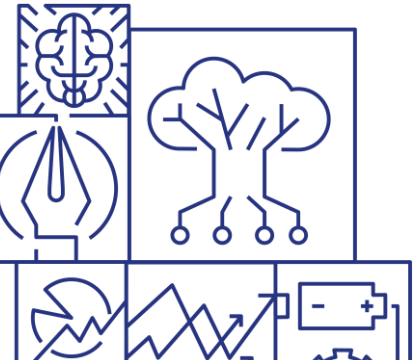
# Estados de un proceso

Terminología:

- **preparado** o **listo** (*ready*)
- **corriendo** o **en ejecución** (*running*)
- **bloqueado** (*blocked*) o **dormido** (*asleep*) o **en espera** (*waiting*).
- **suspender** (*swap out*) = enviar el proceso a área de intercambio (*swap*)
- **reanudar** (*swap in*) = traer el proceso desde área de intercambio a memoria



# Transiciones de Estado



## Paso a preparado

- **Creación:** se carga el programa en memoria e iniciamos el proceso
- **Desde Ejecución:** CPU pasa a ejecutar otro proceso (**cambio de contexto**)
- **Desde bloqueado:** acabó una E/S por la que un proceso estaba esperando
- **Desde suspendido Listo:** SO decide reanudarlo y ya estaba preparado desde antes.

## Paso a ejecución

- Se toma el primero de la cola de preparados cuando el reloj haya interrumpido el que estaba en ejecución.

## Paso a bloqueado

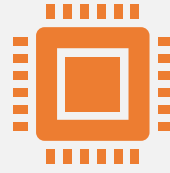
- Desde **ejecución**, al solicitar E/S o desde **suspendido-bloqueado** porque volvió a memoria (reanudación) pero no acabó E/S.

## Paso a suspendido listo o suspendido bloqueado

- SO puede decidir suspender un proceso parado (ya sea listo o bloqueado), pasando al estado correspondiente en cada caso.



# Creación de un proceso



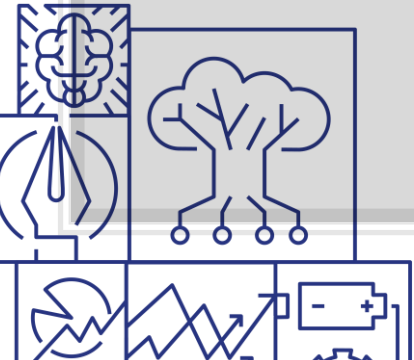
Al crear un proceso (suponemos que el código ya está en memoria) el SO debe:

asignarle un identificador  
crear e inicializar su BCP



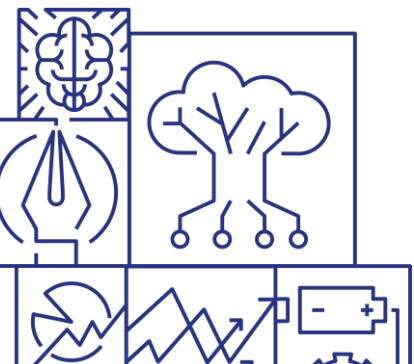
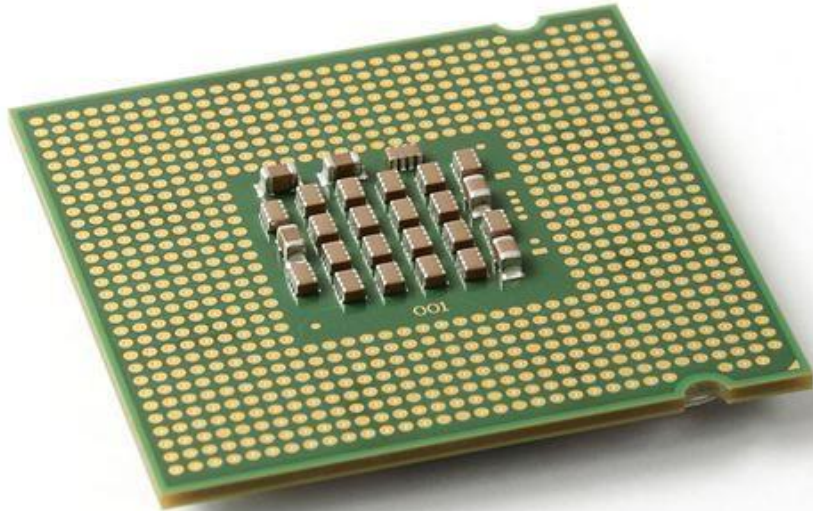
Actualizar el BCP para llevar cuenta de él.

pasarlo a la cola de preparados.



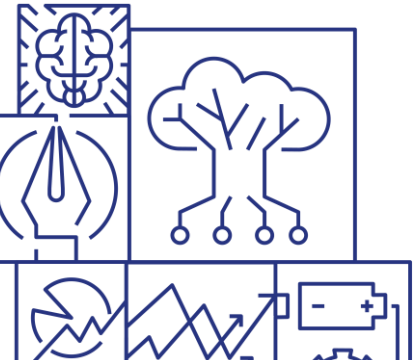


# Creación de procesos



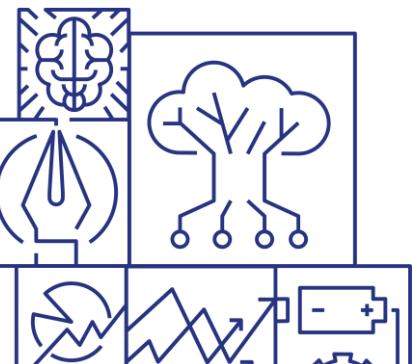
# Destrucción de un proceso

- **Destruir proceso:** se elimina la entrada BCP.
- Si hay **procesos hijos:** puede tener que esperar por ellos o finalizarlos forzosamente.



```
[tawbarta@localhost ~]$ ps axjf
```

PPID	PID	PGID	SID	TTY	TPGID	STAT	UID	TIME	COMMAND
0	2	0	0	?	-1	S	0	0:00	[kthreadd]
2	3	0	0	?	-1	S	0	0:00	\_ [ksoftirqd/0]
2	4	0	0	?	-1	S	0	0:00	\_ [kworker/0:0]
2	5	0	0	?	-1	S<	0	0:00	\_ [kworker/0:0H]
2	6	0	0	?	-1	S	0	0:00	\_ [kworker/u:0]
2	7	0	0	?	-1	S<	0	0:00	\_ [kworker/u:0H]
2	8	0	0	?	-1	S	0	0:00	\_ [migration/0]
2	9	0	0	?	-1	S	0	0:00	\_ [watchdog/0]
2	10	0	0	?	-1	S<	0	0:00	\_ [cpuset]
2	11	0	0	?	-1	S<	0	0:00	\_ [khelper]
2	12	0	0	?	-1	S	0	0:00	\_ [kdevtmpfs]
2	13	0	0	?	-1	S<	0	0:00	\_ [netns]
2	14	0	0	?	-1	S	0	0:00	\_ [bdi-default]
2	15	0	0	?	-1	S<	0	0:00	\_ [kintegrityd]
2	16	0	0	?	-1	S<	0	0:00	\_ [kblockd]

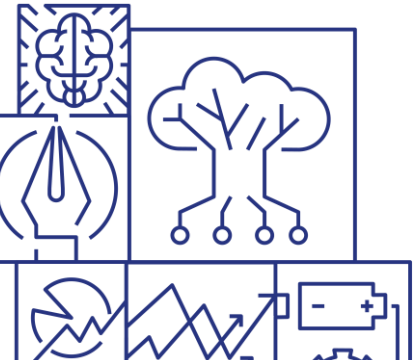


## **2.1.1 Multiprocesos y Jerarquía de procesos**



# Qué son procesos y subprocessos

- Un Subproceso es un conjunto de actividades que tienen una secuencia lógica para cumplir un propósito. Un Subproceso es un Proceso por sí mismo, cuya finalidad hace parte de un Proceso más grande. El proceso más grande se conoce como proceso Padre y el Subproceso como proceso hijo.



# Monoproceso

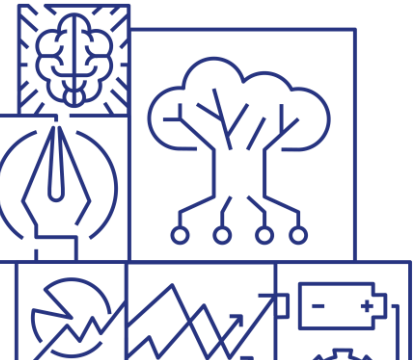
Un sistema operativo que es capaz de manejar solamente un procesador de la computadora, de manera que si la computadora tuviese más de uno le sería inútil.

Windows 95  
Windows 98  
Windows ME

# Multiproceso

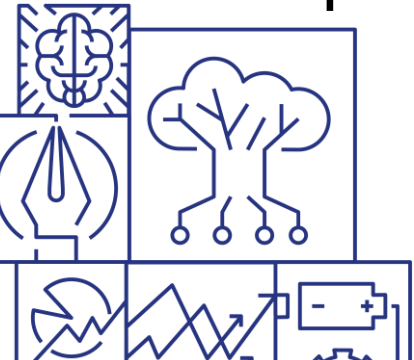
Un sistema operativo multiproceso se refiere al uso de múltiples procesadores para distribuir su carga de trabajo. Generalmente estos sistemas trabajan de dos formas: simétrica o asimétricamente.

UNIX /Linux  
Windows xp profesional

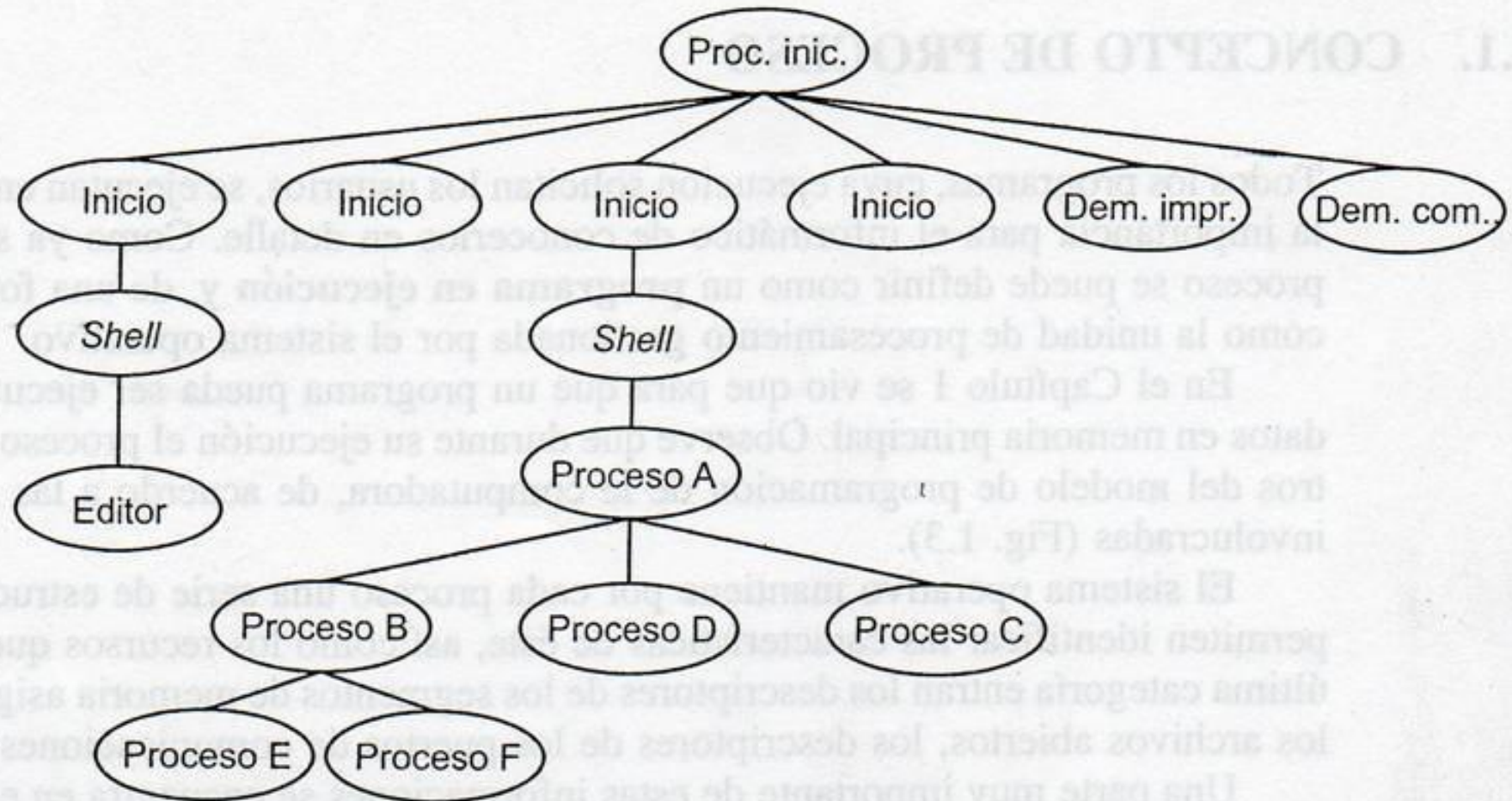


# Jerarquía de Procesos

- Desde el punto de vista macro, los procesos son las actividades claves que se requieren para manejar y, o dirigir una organización, es necesario mostrar la jerarquía de proceso en la siguiente grafica.
- Esta jerarquía muestra cinco niveles: nivel macroproceso, nivel proceso, nivel subprocesso, nivel actividades y nivel de tareas específicas a realizar en un proceso concreto.



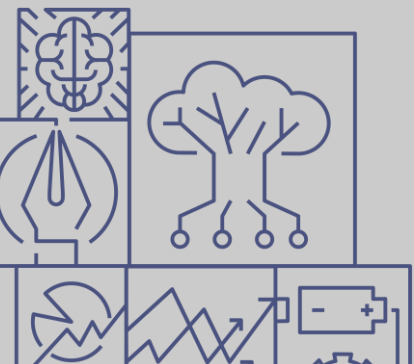




**Figura 3.1.** *Jerarquía de procesos.*

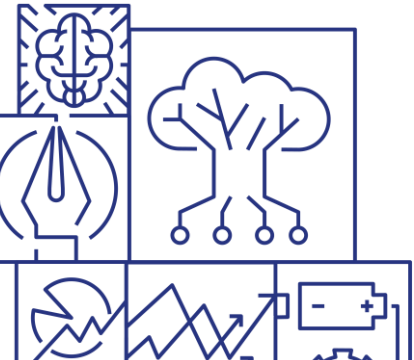
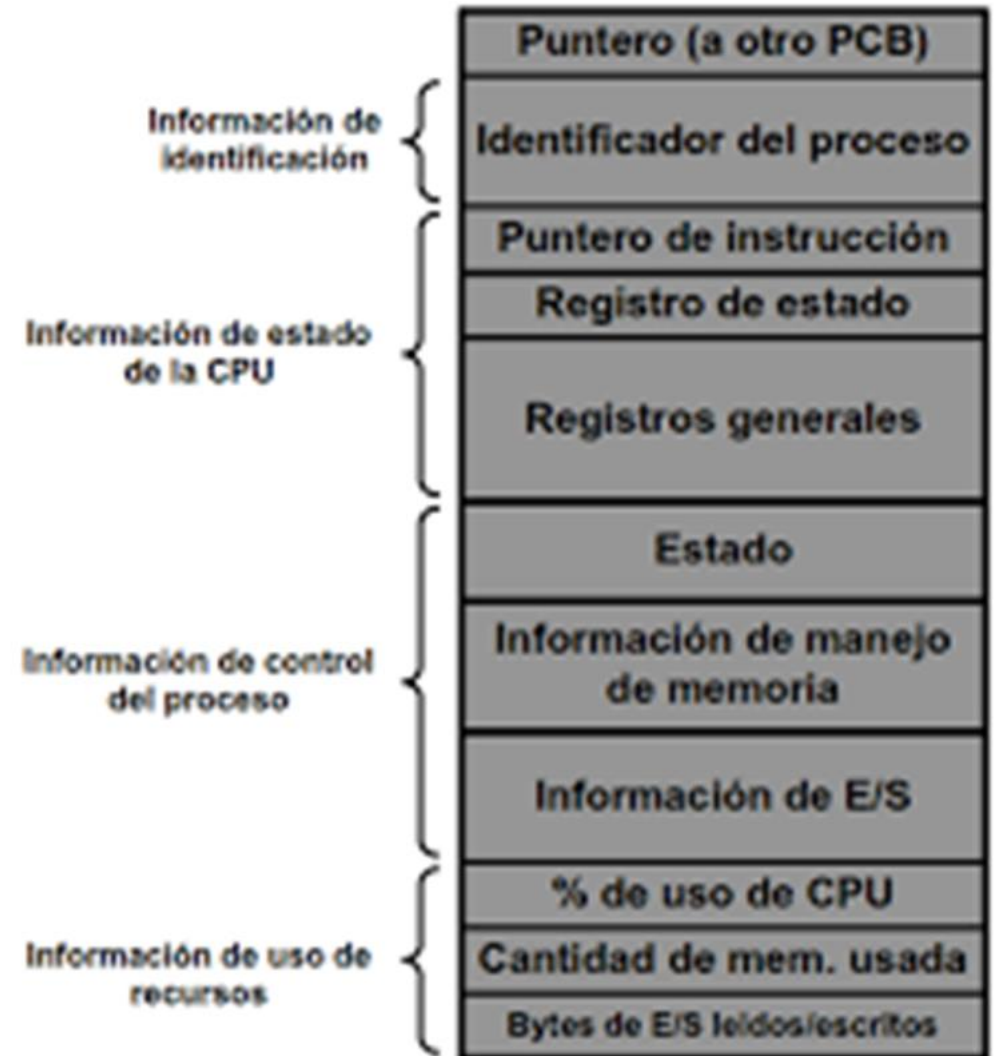


## 2.1.2 Control de Procesos

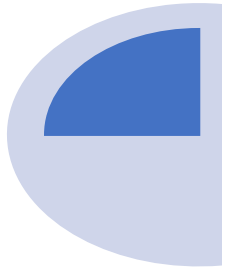


# Bloque de Control de Procesos

Bloque de Control de Proceso  
(en inglés *Process Control Block*,  
*PCB*) son los datos *particulares de cada proceso* que usa el SO para gestionar. Cuando el proceso termina, su BCP es borrado y el registro puede ser utilizado para otros procesos.



# Bloque de Control de Proceso



## Identificación

ID del proceso  
ID del proceso padre  
ID del usuario y grupo propietario

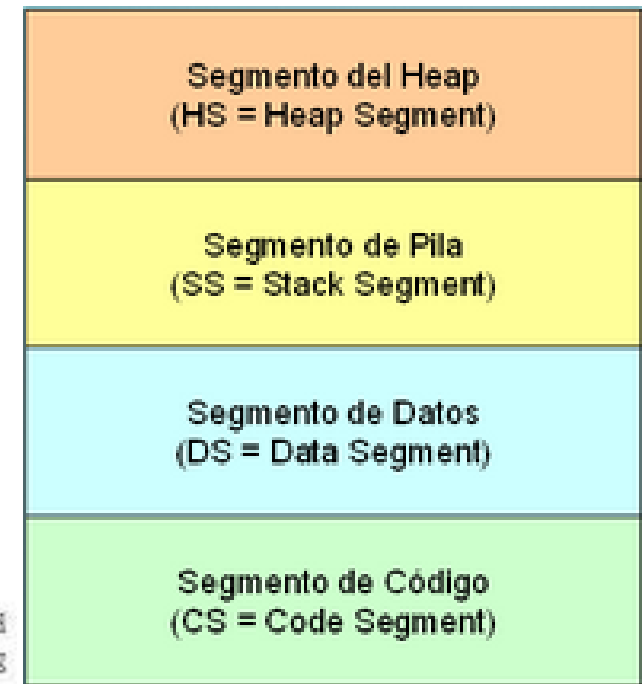
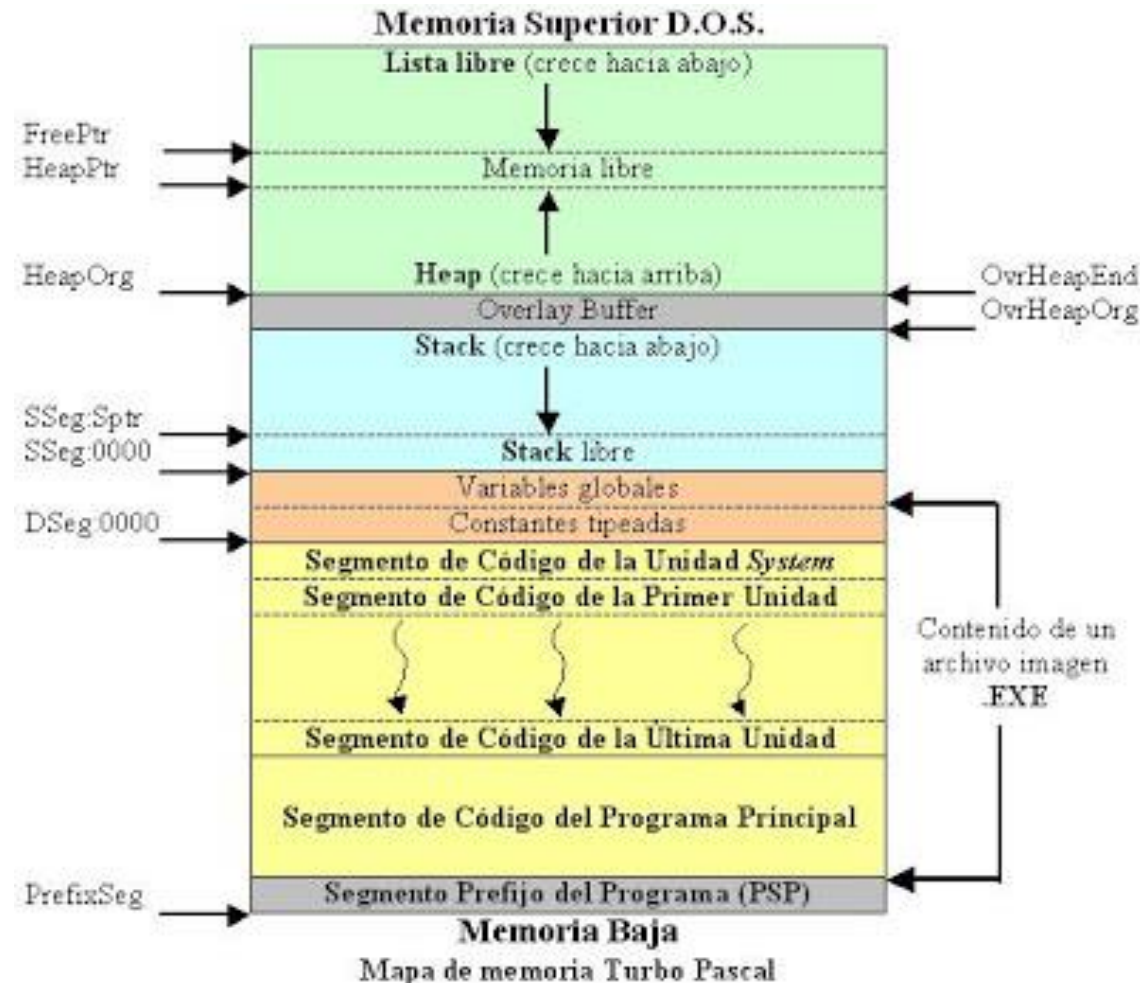


## Planificación

Estado del proceso  
Si estado = bloqueado, el evento por el que espera el proceso  
Prioridad del proceso  
Registros de CPU (acumuladores, punteros de la pila, registros índice y registros generales)  
Información de planificación de CPU (prioridad de proceso, punteros a colas de planificación, etc.)  
Información de gestión de memoria, información de contabilidad (tiempo de uso de CPU, números de procesos, etc.)  
Información de estado de dispositivos E/S (lista de archivos abiertos, etc.).

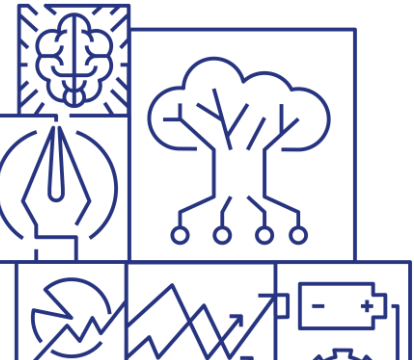
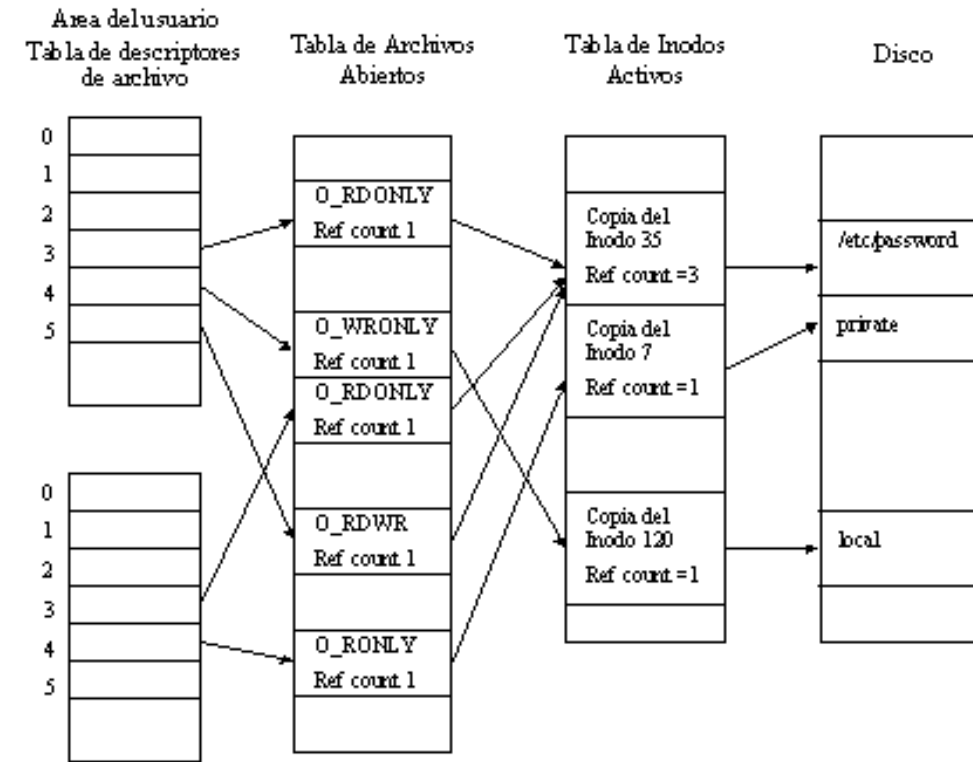
Punteros a  
segmentos de  
memoria asignados:

- Segmento de datos
- Segmento de código
- Segmento de pila
- Recursos asignados:



Archivos abiertos: tabla de descriptores o “manejadores” de archivos (file descriptors / file handles).

- Puertos de comunicación asignados
- Punteros para organizar los procesos en colas.
- Información para comunicación entre procesos: señales, mensajes



fidéлитas  
Universidad



Gracias

