



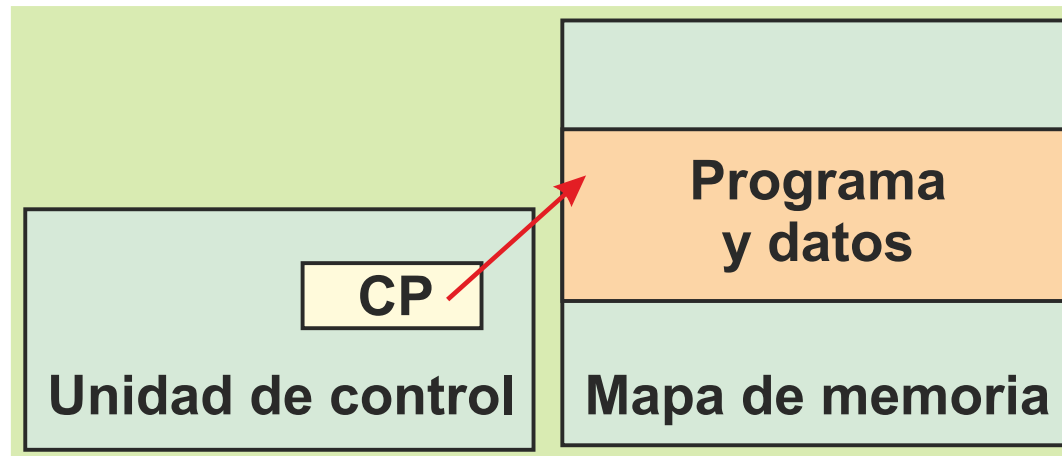
# SISTEMAS OPERATIVOS INTRODUCCIÓN

Pedro de Miguel Anasagasti



# INTRODUCCIÓN AL SISTEMA OPERATIVO

- **Máquina desnuda:** Término que se aplica a un computador carente de sistema operativo.
- El término resalta el hecho de que un computador en sí mismo no hace nada. Como sabemos, solamente es capaz de repetir a alta velocidad la secuencia de: lectura de instrucción de máquina, incremento del CP y ejecución de la instrucción leída.
- Para que el computador realice una función determinada.
  - Ha de tener en el mapa de memoria un programa máquina específico para realizar dicha función.
  - Ha de conseguirse que el registro CP contenga la dirección de comienzo de ese programa.



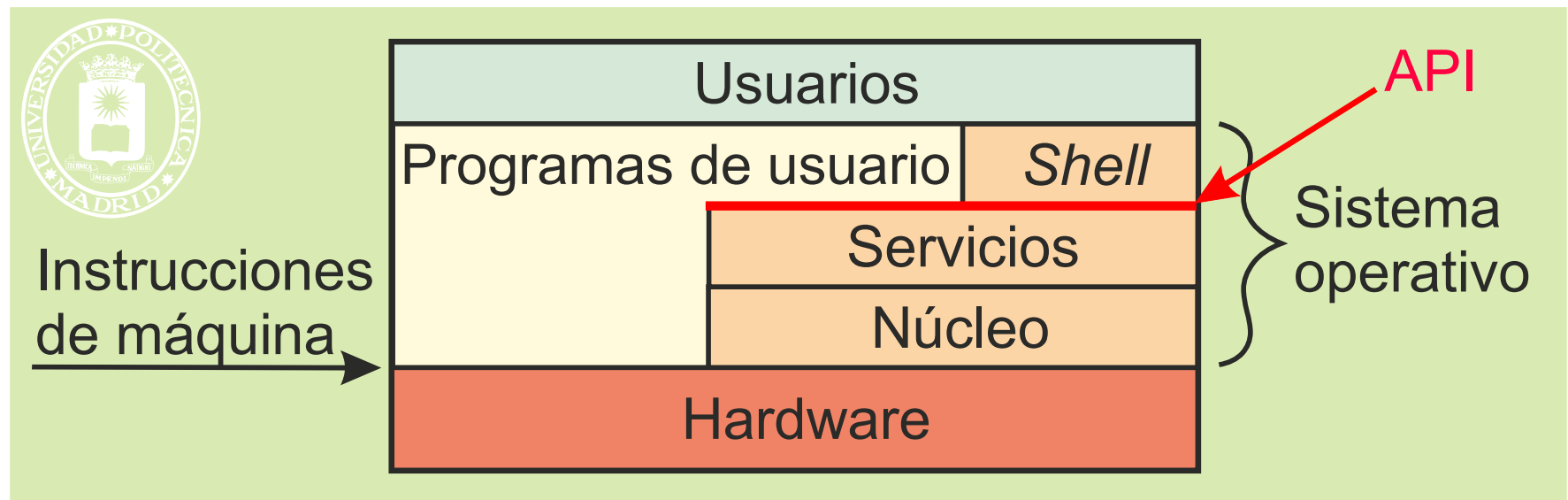
- La misión del sistema operativo es completar (vestir) a la máquina mediante una serie de programas que permitan su cómodo manejo y utilización.

## Sistema Operativo

- Capa de SW que facilita el uso del computador (conjunto de programas).

## Funciones del Sistema Operativo

- Soporte de usuario (**mandatos**).
- Máquina ampliada (**servicios** a los programas).
- Gestor de recursos (UCP, memoria, ficheros...) entre los programas activos.

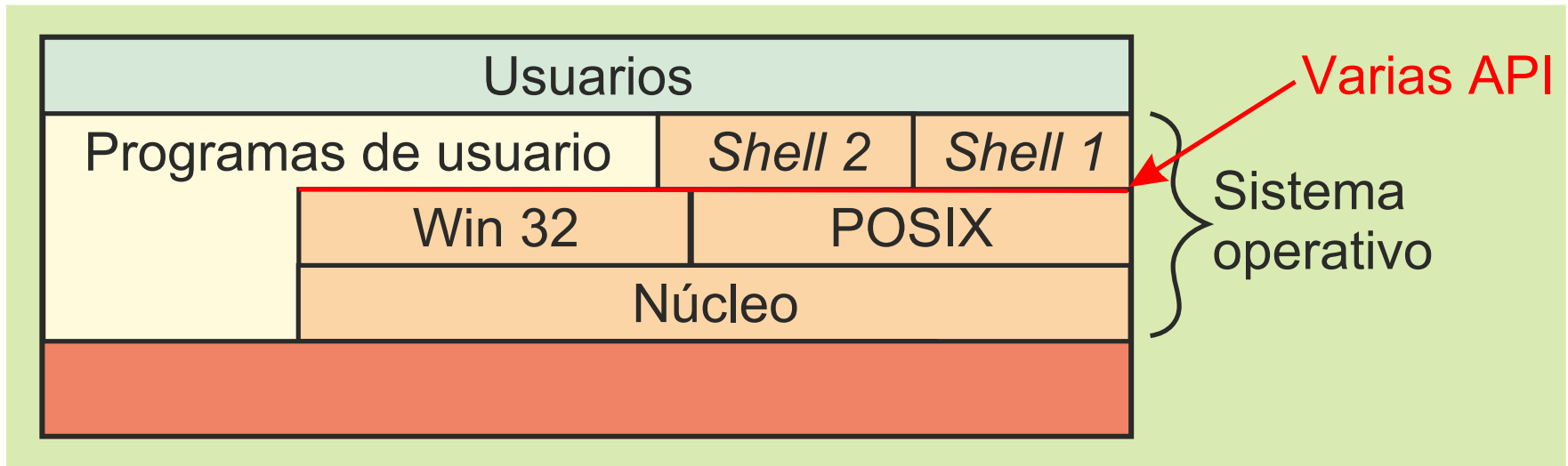


**API: Application Programming Interface.**



- **Soporte de usuario.**
  - Intérprete de mandatos o Shell.
- **Máquina ampliada (servicios).**
  - Ejecución de programas (procesos).
  - Órdenes de E/S.
  - Operaciones sobre ficheros.
  - Servicios de memoria.
  - Comunicación entre procesos.
  - Detección y tratamiento de error.
- **Gestor de recursos (UCP, memoria, ficheros, puertos...).**
  - Asignación y recuperación de recursos.
  - Los recursos asignados pueden ser físicos o lógicos.
  - Protección de los usuarios.
  - Contabilidad/monitorización.

- **Núcleo:** Es el componente del SO que gestiona el HW. Es la parte del SO que obligatoriamente ha de ejecutar en modo núcleo.
- **Servicios:** Como el Win32 o el POSIX. Suministra una capa de servicios o API (Application Programming Interface).





## Usuario

- Persona autorizada a utilizar un sistema.
- Se identifica en la autenticación mediante.
- Código de cuenta.
- Clave (password).
- Internamente el SO le asigna el “uid” (user identifier).

## Super-usuario o administrador

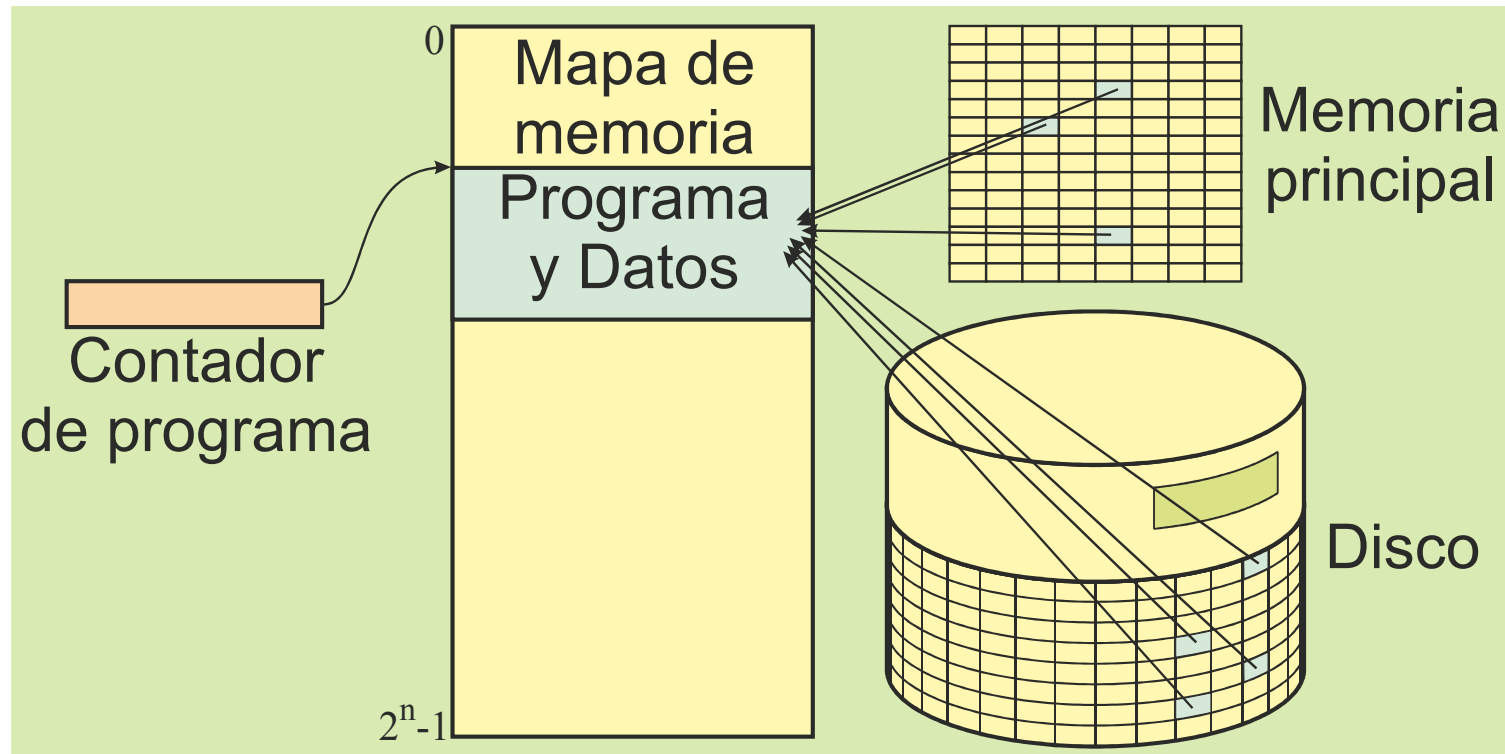
- Tiene todos los derechos.
- Administra el sistema.
- En UNIX el super-usuario tiene uid = 0.

## Grupo de usuarios

- Los usuarios se organizan en grupos: “gid” (group identifier).
  - Alumnos de 2º.
  - Profesores del DATSI.
- Todo usuario ha de pertenecer a un grupo.

## Programa en ejecución

- El programa ha de estar traducido a código máquina.
- El programa ha de estar almacenado en disco.
- El programa ha de cargarse en memoria (en el mapa de memoria).
- El contador de programa ha de apuntar al programa



**Información asociada:** Imagen de memoria, BCP y estado procesador





# PROTECCIÓN



- Es necesario proteger unos usuarios de otros, de forma que los programas de un usuario no interfieran con los programas de otro y que no puedan acceder a la información de otro.
- Una de las funciones principales del SO es proteger unos usuarios de otros.
- Como el SO es un conjunto de programas, no puede supervisar las acciones de los programas cuando estos están ejecutando.

**Si ejecuta un programa de usuario NO ejecuta el sistema operativo**

- Esta supervisión la tiene que hacer directamente el HW, para lo que cuenta con unos mecanismos que impiden que los programas de los usuarios realicen operaciones contrarias a la seguridad.
- Tanto el procesador como la unidad de memoria tienen mecanismos de protección.



## Mecanismos

- El procesador tiene **dos modos de ejecución**.
  - **Núcleo**: ejecuta todas las instrucciones de máquina. Permite acceder a la E/S y registros especiales.
  - **Usuario**: ejecuta un subconjunto de las instrucciones. No permite acceder al mapa de E/S ni a parte del mapa de memoria.
- Las interrupciones pasan automáticamente a modo de ejecución núcleo (sólo se puede pasar de esta forma).
- La memoria tiene mecanismos de protección que limitan las direcciones que puede acceder un programa.

## Consideraciones

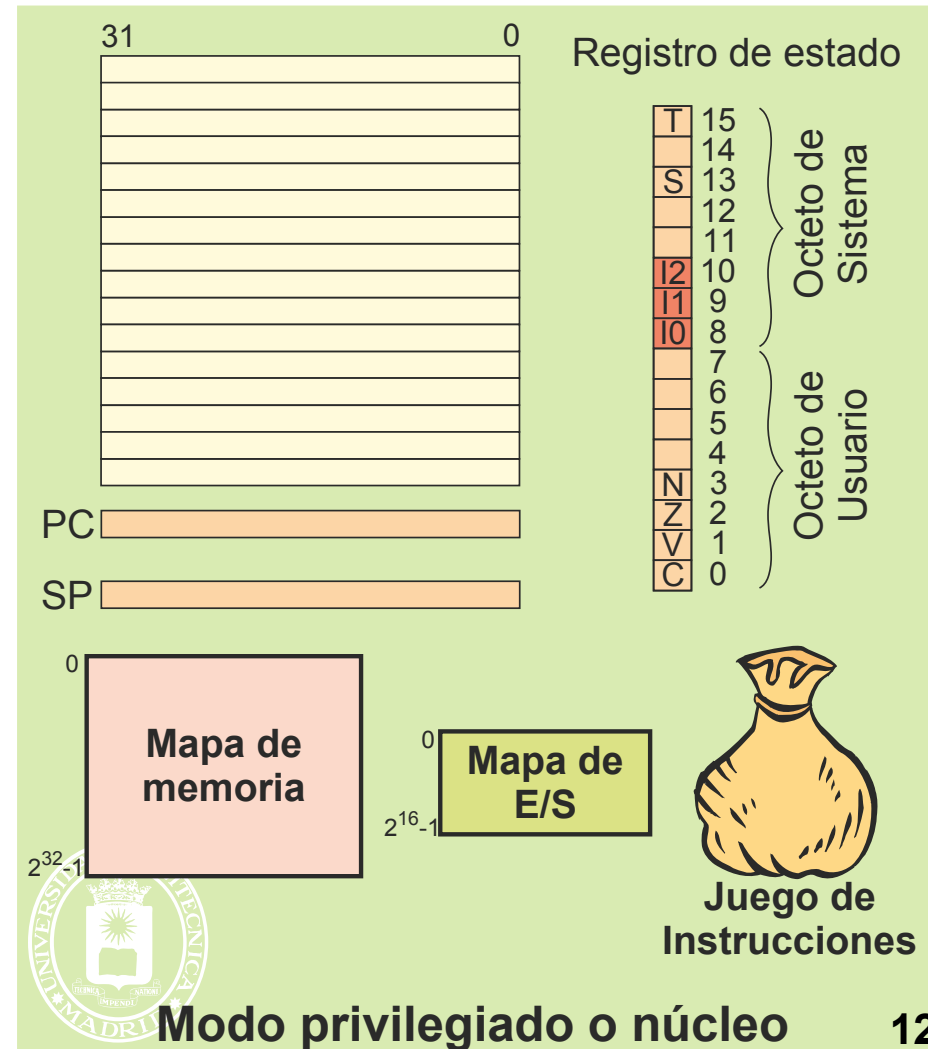
- EL SO es el encargado de tratar todas las interrupciones. Por tanto es el único programa que ha de ejecutar en modo núcleo.
- Los programas de usuario no pueden acceder al mapa de E/S, por tanto no pueden **acceder directamente** a los periféricos. Lo que significa que no pueden leer los ficheros de otros usuarios.
- Los programas de usuario acceden a los periféricos **a través del SO**, que sólo permite realizar las operaciones autorizadas. Mediante la **instrucción TRAP**

# MODELO DE PROGRAMACIÓN DE USUARIO Y DE NÚCLEO

© Latsi UPM 2014



- El computador puede funcionar en modo usuario o núcleo.
- El modo usuario es restringido. No se puede acceder a todos los elementos de almacenamiento y no se pueden ejecutar todas las instrucciones.





La seguridad es una de las funciones fundamentales del SO.

Está basada en los mecanismos de protección y tiene dos facetas: autenticación y privilegios.

## Mecanismos HW de protección.

- Vigilan la ejecución de los programas.
- Informan de los intentos de trasgresión de los programas.

**Autenticación:** garantiza la identidad de los usuarios.

- Algo que conoce el usuario.
- Propiedad del usuario.
- Algo que tiene el usuario.
- Lugar del usuario.

**Privilegios:** definen lo que puede hacer cada usuario.

- Información por recurso.
- Información por usuario.



## Permisos UNIX

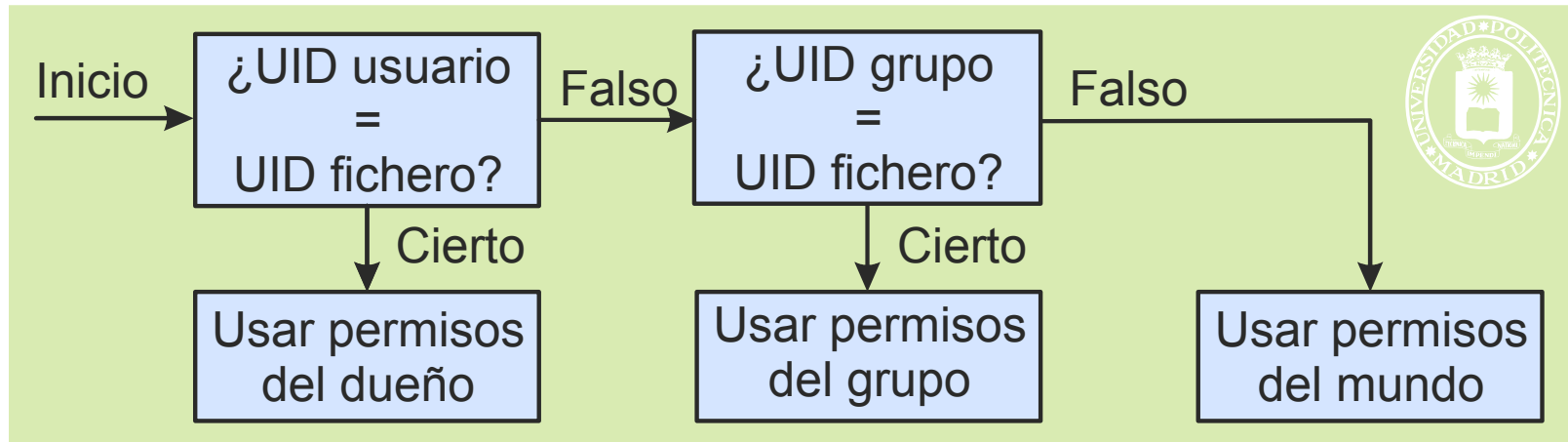
- Para ficheros de datos.

- r: Especifica que el fichero se puede leer.
- w: Especifica que el fichero se puede escribir.
- x: Especifica que el fichero se puede ejecutar.

Dueño	Grupo	Mundo
$\mathcal{R}$ W $\mathcal{X}$	$\mathcal{R}$ W $\mathcal{X}$	$\mathcal{R}$ W $\mathcal{X}$

- Para ficheros de directorio.

- r: Especifica que el directorio se puede leer, es decir, se puede hacer un “ls” para conocer su contenido.
- w: Especifica que el directorio se puede escribir, es decir, se puede añadir o borrar un fichero del directorio.
- x: Especifica que el directorio se puede atravesar para seguir analizando un nombre de fichero. (/home/datsi/asignaturas/ssoo/practicas/leeme.txt)

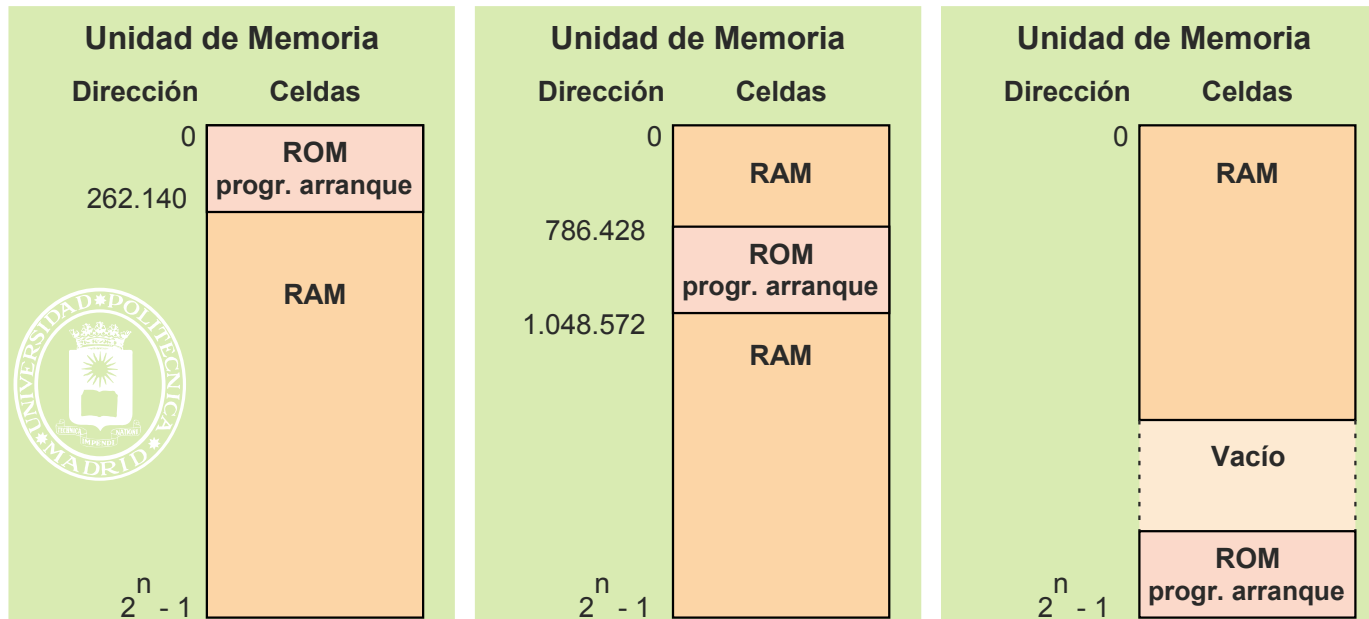


r-x rw- rw- ¿Puede el dueño modificar el fichero?



# ARRANQUE DEL COMPUTADOR

- El Reset carga valores predefinidos en registros.
  - CP ← dirección de arranque del cargador ROM.
- Se ejecuta el Programa de Arranque del sistema.
  - Que hace un test del sistema.
  - Y trae a memoria RAM el boot del SO, que está en el disco.

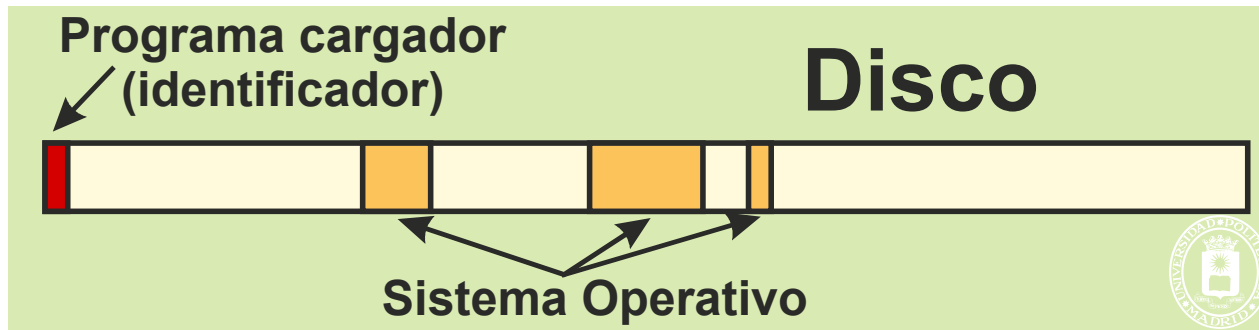


- El Programa de Arranque funciona en modo Real y modo Núcleo
  - accede a los periféricos
  - las tablas de páginas de la memoria virtual las crea el SO (no hay SO )

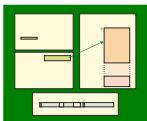
(En el caso del PC la ROM contiene, además, SW de E/S (BIOS)).



El SO está almacenado en el disco.



- El Programa de Arranque ROM carga el sector de boot del SO y le da control.
- El boot del SO carga el SO residente y da control al programa de arranque del SO, que:
  - Comprueba el hardware.
  - Comprueba el sistema de ficheros.
  - Establece las tablas internas del SO, como son las tablas de páginas.
    - A partir de este momento se empieza a trabajar con memoria virtual.
  - Crea procesos (según el tipo de SO).
    - Proceso INIT.
    - Procesos o tareas del sistema (demonios).
    - Procesos de login (uno por terminal).
- Tras la autenticación, el proceso login se convierte en shell.



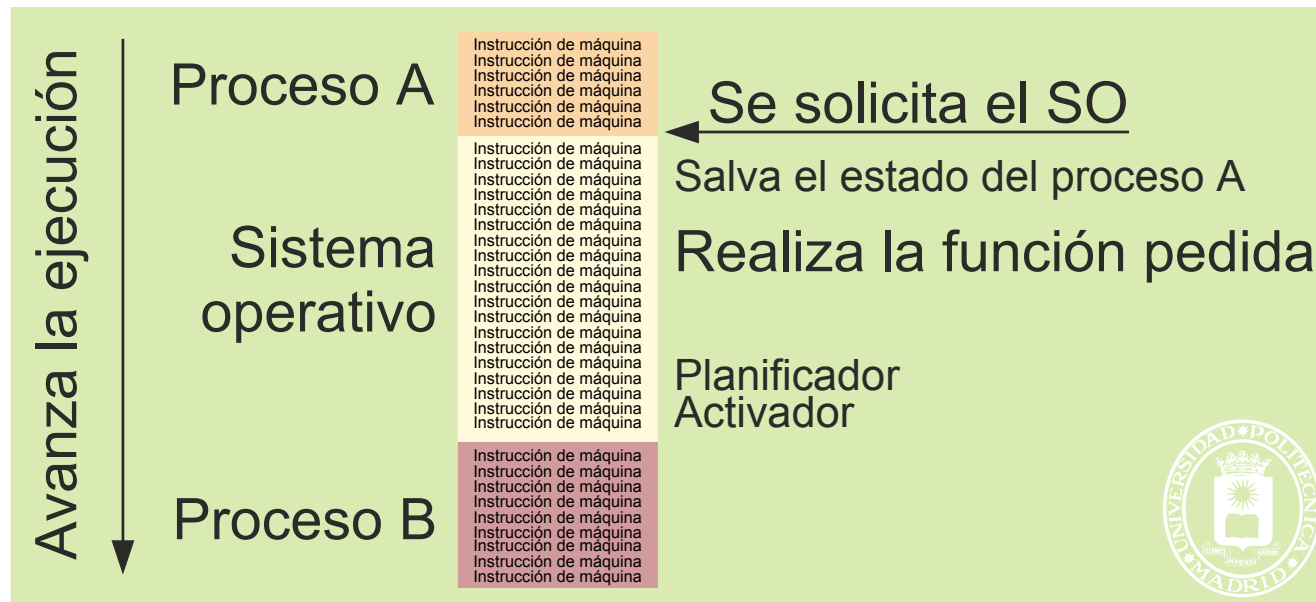


# ACTIVACIÓN DEL SO



## El SO se activa por interrupciones

- Solicitudes emitidas por los programas para solicitar servicios al SO.
- Interrupciones producidas por los periféricos.
- Excepciones del SW o del HW.



## Ciclo simplificado del SO

- Salva el estado del Proceso A en curso (contenido de los registros del procesador), para poder seguir su ejecución más adelante.
- Realiza la función solicitada.
- Planifica y activa un Proceso B. (Puede planificar el A otra vez)



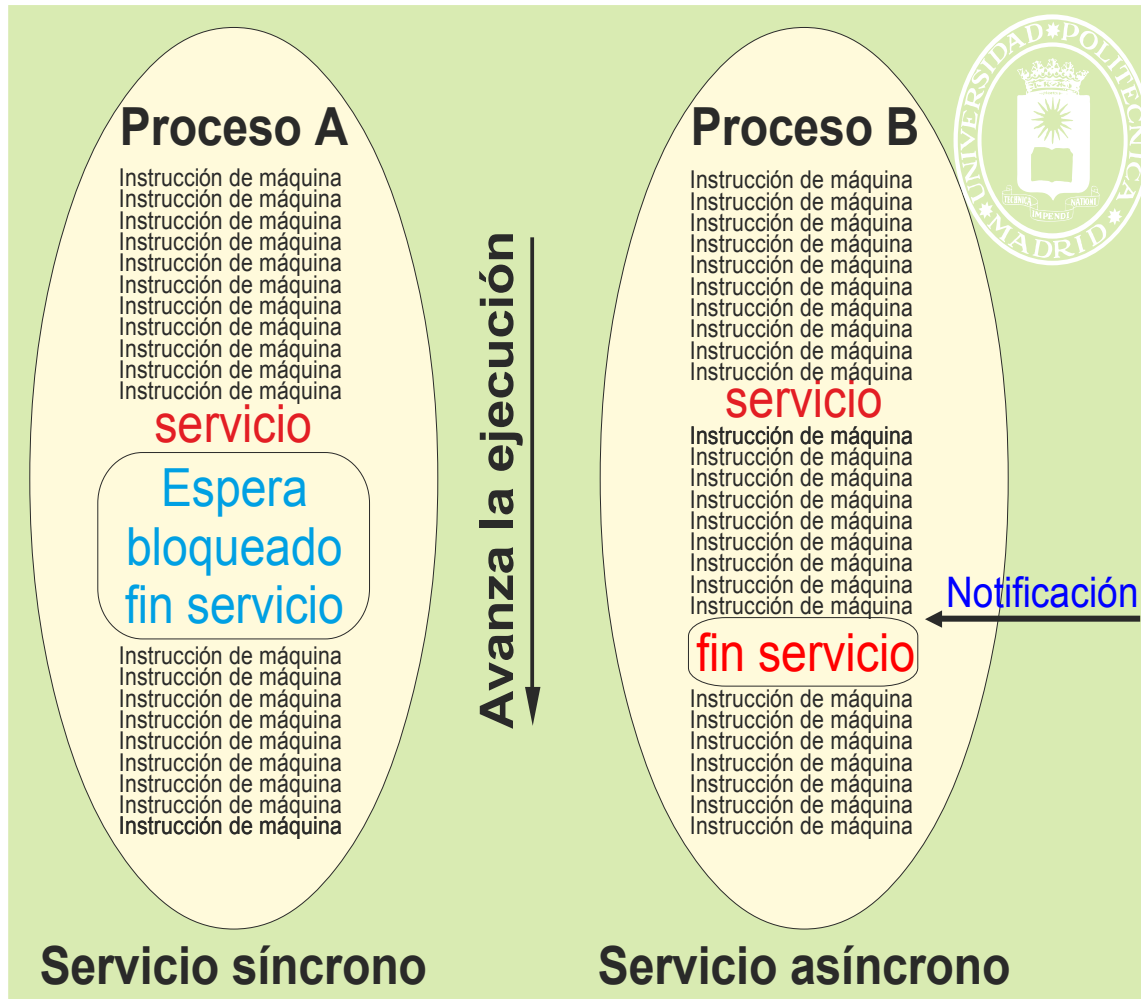
- Posibles formas de solicitar un servicio:
  - Por llamada → problema de seguridad, por lo que **no se puede usar**.
  - Por interrupción o TRAP (pone el procesador en modo núcleo).
- Rutina de biblioteca: Programa de máquina que prepara la llamada al SO.

**Sintaxis de la llamada** a la rutina de biblioteca en alto nivel:

```
mivariable = servicio(parámetro 1, parámetro 2, etc.);
```

- En muchos casos los parámetros de la rutina de biblioteca son referencias a estructuras que recibe y/o modifica el SO.
- En UNIX el servicio devuelve generalmente un entero. El valor -1 significa que el servicio ha fracasado:
  - Por error
  - Porque le llega una señal al proceso (`errno = EINTR`)
  - Variable global `errno` y fichero `errno.h` (función `perror`)
- Dado que casi todos los servicios del SO pueden fallar por falta de recursos o por falta de privilegios, en un programa profesional es **imprescindible** incluir código que trate el caso de **fracaso del servicio**.
- En los ejemplos que plantearemos en algunos de los temas no incluiremos este tratamiento de error, puesto que complica el flujo normal de ejecución que se quiere destacar.

- Síncrono o bloqueante
- Asíncrono o no bloqueante



El uso de servicios síncronos es mucho más sencillo



## La rutina de biblioteca:

- Almacena los parámetros del servicio siguiendo un convenio con el SO (p.e. en la pila del proceso).
- Realiza el TRAP.

## El SO

- Bloquea al proceso
  - Recoge los parámetros.
  - Comprueba que puede realizar el servicio solicitado.
  - Realiza el servicio.
    - Puede requerir varios pasos.
    - Si hay espera se pone en ejecución otro proceso.
  - Genera la respuesta al proceso y la almacena siguiendo un convenio con la rutina de biblioteca.
- Desbloquea al proceso

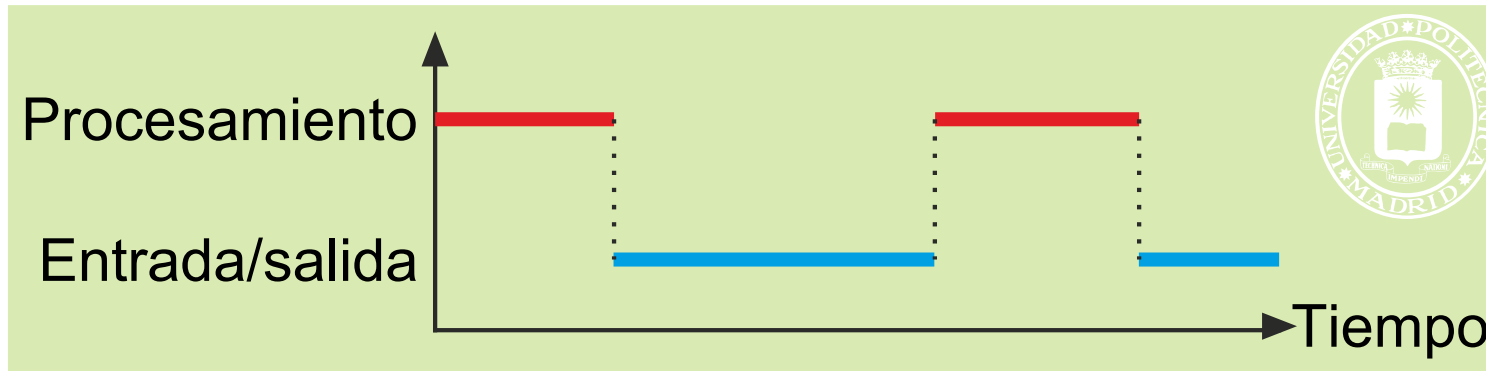
La rutina de biblioteca toma la respuesta del SO y la deposita en la variable de retorno `mivariable`.



# PROCESOS

## Base de la multitarea

- Paralelismo real entre E/S y el procesador (técnica de DMA).
- Los procesos alternan fases de:
  - E/S
  - y de procesamiento.



- La memoria almacena varios procesos.

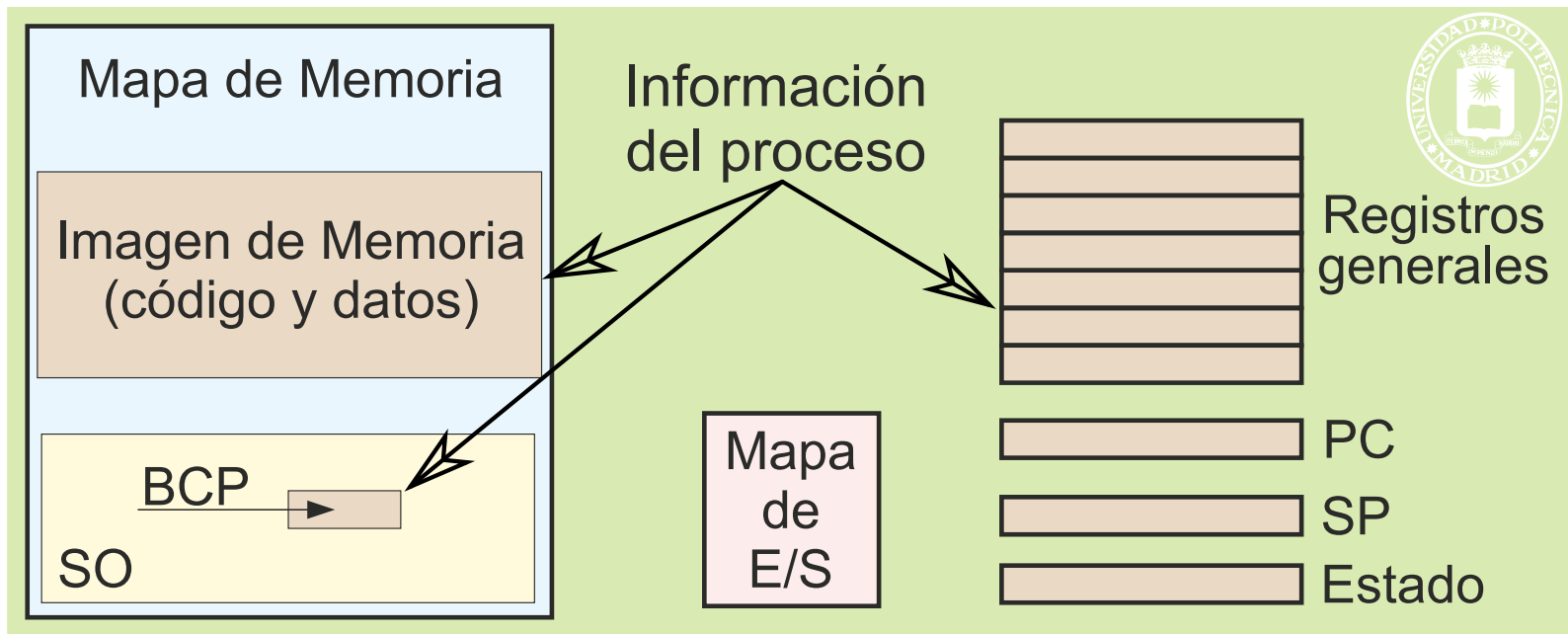




- Proceso**
- Programa en ejecución.
  - Unidad de procesamiento gestionada por el SO.

## Información del proceso

- Imagen de memoria (core image).
- Estado del procesador (registros del modelo de programación).
- Bloque de control del proceso BCP. Algunos contenidos:
  - Identificador “pid”
  - Estado (registros)
  - Segmentos de memoria
  - “uid” y “gid”
  - Señales
  - Puertos
  - Ficheros abiertos
  - Semáforos
  - Temporizadores



## Familia de procesos

- Proceso hijo.
- Proceso padre.
- (Proceso emparentado).

## Vida de un proceso

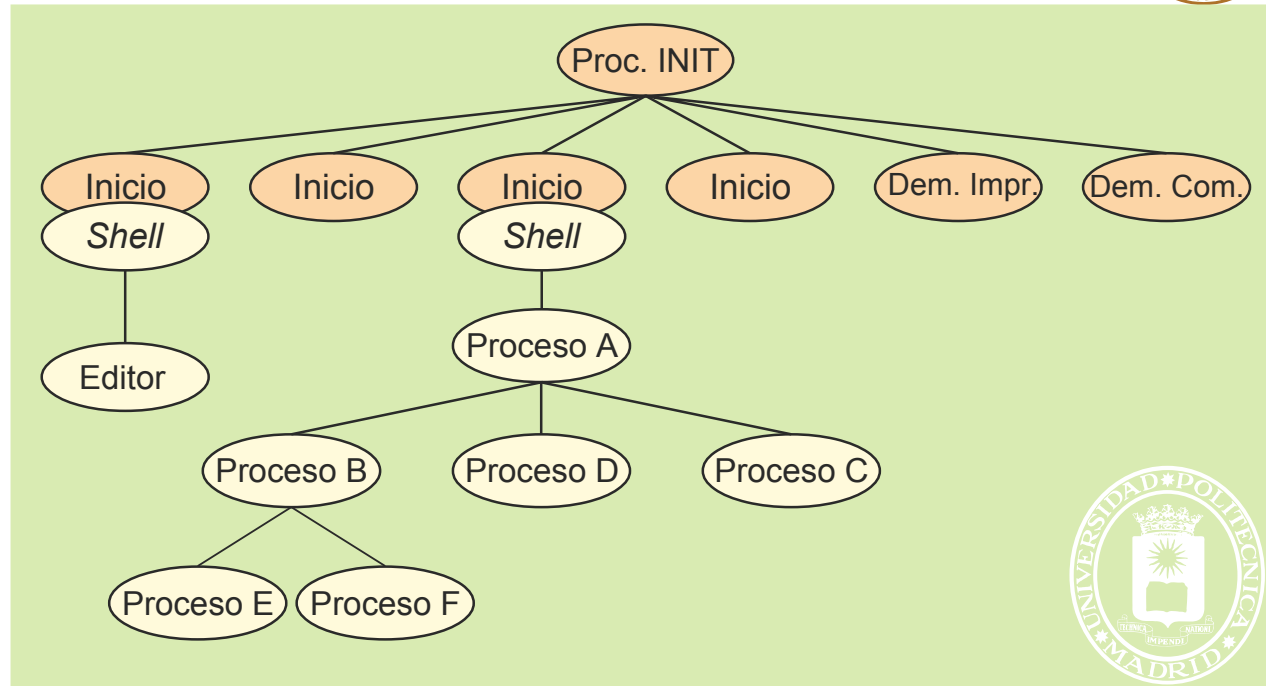
- Crea.
- Ejecuta.
- Muere o termina.

## Ejecución del proceso

- No interactivo (Batch y segundo plano)
- Interactivo o primer plano.

## Grupo de procesos

- Grupos de procesos dependientes de cada inicio.





## **VISIÓN GENERAL DEL SISTEMA**

