

SISTEMAS OPERATIVOS – PRIMER PARCIAL

TEMA 1: Soporte físico [UPM]

- 1.1: Estructura y funcionamiento del computador (1.1)
- 1.2: Modelo de programación del computador: modos de operación (1.2)
- 1.3: Interrupciones (1.3)
- 1.4: Entrada/Salida (1.6.1, 1.6.4)

TEMA 2: Introducción a los sistemas operativos [UPM]

- 2.1: ¿Qué es un sistema Operativo? (2.1)
- 2.2: Activación del sistema operativo (2.3)
- 2.3: Tipos de sistemas operativos (2.4)
- 2.4: Componentes del sistema operativo (2.5)
- 2.5: Diseño de los sistemas operativos (2.9.1)
- 2.6: Historia de los sistemas operativos (2.10)

TEMA 3: Procesos [UPM]

- 3.1: Concepto de Proceso (3.1)
- 3.2: Multitarea (3.2)
- 3.3: Información del proceso (3.3)
- 3.4: Vida de un proceso (3.4)
- 3.5: Threads (3.8)
- 3.6: Servicios (3.13.1: Identificación de procesos, creación de procesos y terminación de procesos)

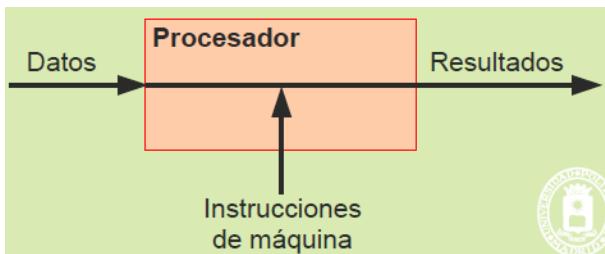
TEMA 4: Interbloqueos [Apuntes]

- 4.1: Introducción (4.1)
- 4.2: Tipos de recursos (4.2)
- 4.3: Modelo del sistema (4.3)
- 4.4: Definición y caracterización del interbloqueo (4.4)
- 4.5: Detección y recuperación del interbloqueo (4.5)
- 4.6: Prevención del interbloqueo (4.6)
- 4.7: Evitación del interbloqueo (4.7)

TEMA I – SOPORTE FÍSICO

I.1. Estructura y funcionamiento del computador

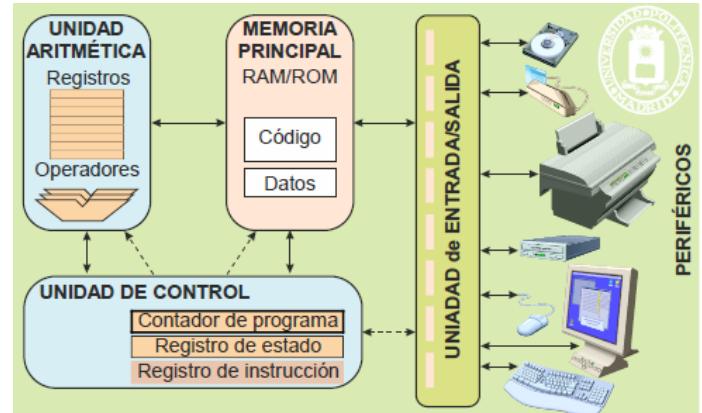
El computador es una máquina en el que entra un flujo de datos de entrada, son procesados (flujo de instrucciones máquina) y genera un flujo de datos resultado:



I. Enumerar los componentes básicos del computador (I.1)

Las máquinas funcionan con una arquitectura von Neumann (MUCE):

- A) Memoria principal (RAM+ROM).
- B) Unidad Aritmética-lógica.
- C) Unidad de control.
- D) Unidad de entrada/salida → Relacionado con los periféricos.

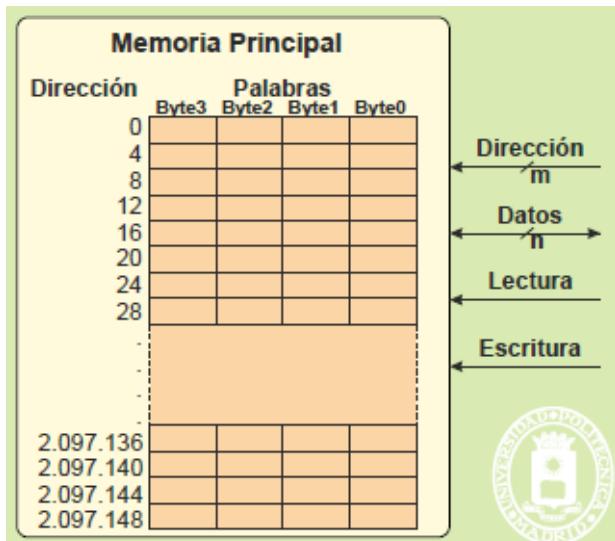


Procesador/Unidad Central de Proceso (UCP)¹ = Unidad aritmética + unidad de control.

A) Memoria Principal = RAM (Random Access Memory) + ROM (Read Only Memory).

Funciones:

- Residen los programas máquinas (permanentes) → ROM.
- Procesamiento de datos cargados en memoria dinámica → RAM.



2. Conocer cómo se realiza el acceso a la memoria principal (concepto de "palabra") (I.1)

La memoria está formada por celdas idénticas.

El acceso a los datos puede ser de lectura o escritura.

Se reserva una palabra de varios bytes para el direccionamiento directo² (típicamente 4 u 8 bytes, múltiplos de 4 u 8).

¹ Un circuito integrado puede incluir varios procesadores (llamados núcleos o cores).

² El direccionamiento directo se refiere a la secuencia lineal normal de un proceso en ejecución.

El direccionamiento indirecto se refiere a la interrupción de un proceso para gestionar otro proceso (ver I.3).

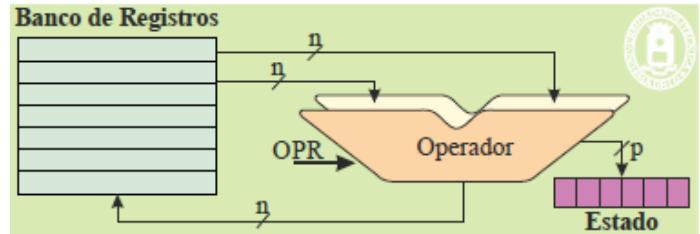
B) Unidad Aritmético-lógica (UAL)

LA UAL realiza una serie de operaciones aritméticos y lógicas sobre uno o dos operandos almacenados en un banco de registros de la UAL o provienen directamente de la Memoria Principal.

La UAL da como salida el resultado de la operación aritmético o lógica y, además resultados adicionales que sirven para la toma de decisiones de los programas:

Se pone a “1” el bit de estado si el resultado es:

- Cero.
- Signo negativo.
- Lleva acarreo.
- Lleva desbordamiento.



C) Unidad de control (UC)

Se encarga de coordinar el funcionamiento del conjunto:

- 1º) Lee la instrucción máquina en la que comienza el programa.
- 2º) Interpreta la instrucción (aritmético, lógica, de salto...).
- 3º) Lee los datos de memoria referenciados (si los hay).
- 4º) Ejecuta la instrucción.
- 5º) Almacena el resultado de la instrucción (si lo hay)



3. Conocer la funcionalidad de los principales registros del procesador: PC, RI, RE (1.1)

Registros de la UC	
Contador de programa (Program Counter) PC	Indica la siguiente instrucción máquina a ejecutar.
Puntero de pila (Stack Pointer) SP	Sirve para manejar cómodamente una pila en memoria principal
Registro de instrucción (Register Instruction) RI	Permite almacenar la instrucción máquina en curso de ejecución
Registro de estado (Register State) RS	Almacena información derivada de la ejecución del programa (bits de estado aritméticos) y cómo ha de comportarse el computador (bits de interrupción, modo de ejecución...).

D) Unidad de Entrada/Salida

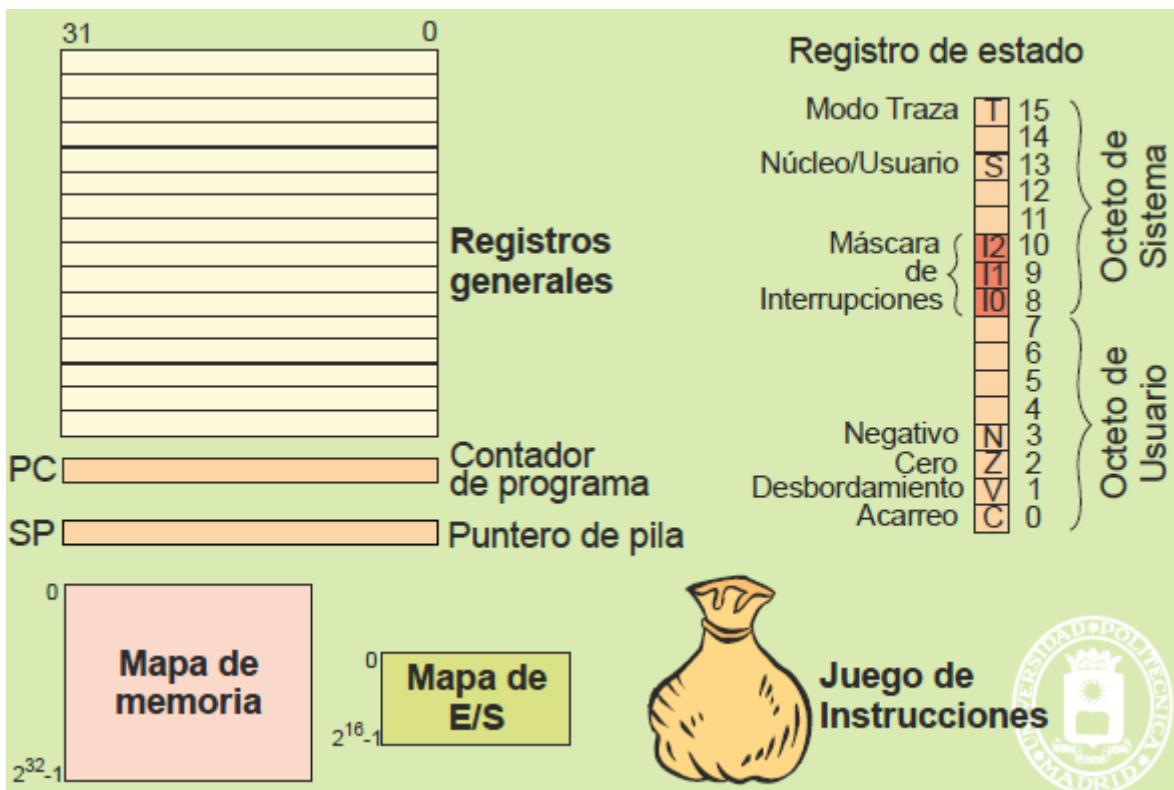
La Unidad de E/S se encarga de hacer la transferencia de información entre la Memoria Principal y los periféricos o viceversa. Se puede gestionar de dos formas:

- Mediante la Unidad de control (E/S programada).
- Acceso directo a memoria (DMA).

1.2. Modelo de programación del computador

4. Conocer el modelo de programación del computador (1.2.intro)

El modelo de programación a bajo nivel de un computador (Instruction Set Architecture -ISA-) define los recursos y características que ofrece al programador de bajo nivel:



Elementos de almacenamiento → Datos sobre los que se aplicarán las instrucciones máquinas:

- Registros generales (UAL).
- Unidad de Control (PC, SP, RS).
- Mapas de memoria: Direcciones a la RAM y ROM.
- Mapa de E/S: Direcciones de registros E/S. A veces, se integran dentro de los mapas de memoria.

1.2.1. Modos de ejecución

5. Conocer los modos de ejecución del procesador, su influencia sobre el conjunto de instrucciones disponible y su relación con la seguridad (1.2.1)

El computador presenta dos modelos de programación:

- 1) **Modo privilegiado o núcleo:** Se tiene acceso a todos los recursos del computador.
- 2) **Modo usuario:** Se limitan acciones, como:
 - a. Acceso a determinados o partes de registros.
 - b. Acceso a mapas de memoria y mapas de E/S.
 - c. Se reserva un puntero de pila (SP) aparte.

El computador siempre inicia en modo privilegiado y, según el/los bit/s de registro de estado se establece el modo usuario o privilegiado al iniciar sesión, esto es modificable.

1.2.2. Secuencia de funcionamiento del procesador

6. Conocer la secuencia de funcionamiento del procesador (1.2.2)

La Unidad de Control (UC) establece el funcionamiento del computador dentro de un bucle continuo e ininterrumpido³. La secuencia:

1º) Lectura de Memoria Principal de la instrucción máquina apuntada por el Contador de Programa (CP de la UC).

2º) Variación del Contador de Programa (CP)⁴, cuatro casos:

- 2.1. Incremento: Es el más habitual, apunta a la siguiente instrucción (secuencia lineal).
- 2.2. Salto o bifurcación: Rompe la secuencia lineal y salta a otra sección.
- 2.3. Interrupciones externas o internas: El SO rompe la secuencia lineal y da un salto.
- 2.4. Instrucción máquina de llamada al sistema: Produce un efecto similar a la interrupción (p.ej. TRAP, INT, SC).

3º) Ejecución de la instrucción, que puede a su vez provocar:

- Lectura de operandos en memoria.
- Escritura de resultados en memoria.

1.2.3. Registros de control y estado

7. Conocer los conceptos "estado del procesador" y "estado visible" (1.2.3)

Estado del procesador: Es el contenido de todos los registros del procesador en un instante determinado (estático).

Estado visible: Es un subconjunto del estado del procesador visible en el modo de ejecución de usuario, como PC, SP y parte del estado (bits de estado aritméticos, de modo de ejecución y control de interrupciones. Otros registros no lo son, como el de gestión de memoria.

³ Algunos procesadores tienen la instrucción de parada (p.ej. HALT), es poco utilizada, en algunos portátiles para ahorrar batería

⁴ Interés:

Programadores (2.1 y 2.2 – Incremento y salto o bifurcación).

Sistemas Operativos (2.3 y 2.4 – Interrupciones e instrucción máquina de llamada al sistema).

I.3. Interrupciones

Una interrupción es un programa que se activa como una señal en la unidad de control.

9. Enumerar los pasos de que consta el ciclo de aceptación de interrupción (I.3)

El ciclo de aceptación de la interrupción son los pasos que realiza la Unidad de Control para gestionar la interrupción:

- 0º Termina la ejecución de la instrucción máquina.
- 1º Escribe en el Puntero de Pila (SP) registros del procesador (registros de estado y contador del programa).
- 2º Cambia el modo de ejecución de usuario a privilegiado o núcleo (si no lo estaba ya).
- 3º Direccionamiento indirecto: Carga un nuevo valor de una tabla de interrupciones (Interrupt Descriptor Table) y la carga en el contador del programa (vector de interrupción⁵).
- 4º Se ejecuta el programa de tratamiento de interrupción. Normalmente, se inhiben nuevas interrupciones.

10. Distinguir entre interrupciones síncronas y asíncronas (I.3)

Interrupciones síncronas: Es consecuencia directa de las instrucciones máquinas que se están ejecutando en la Unidad de Control de la CPU.

Interrupciones asíncronas: Se producen independientemente de las señales de reloj de la CPI. Son generadas por otros dispositivos hardware.

11. Conocer qué causas pueden generar una interrupción (I.3)

Interrupciones	Tipo	Descripción
Excepciones software (o hardware síncronas)	síncronas	Son producidas directa o indirectamente por el programa en ejecución: Problemas de ejecución, depuración (punto de ruptura), fallo de página.
Instrucciones máquina de llamada al sistema	síncronas	Son instrucciones que permiten a un programa generar interrupciones. P.ej. las instrucciones TRAP, INT o SC.
Excepciones hardware asíncronas	asíncronas	Son producidas por un error en el hardware. P.ej. Error de paridad en bus, en memoria, fallo de alimentación, temperatura excedida...
Interrupciones externas	asíncronas	Son producidas por elementos externos al procesador. P.ej. El reloj, drivers, otros procesadores...

12. Explicar el concepto de inhibición selectiva de interrupciones: nivel, máscara... (I.3)

Los procesadores asignan prioridades a las interrupciones y, suelen estar ordenadas así:

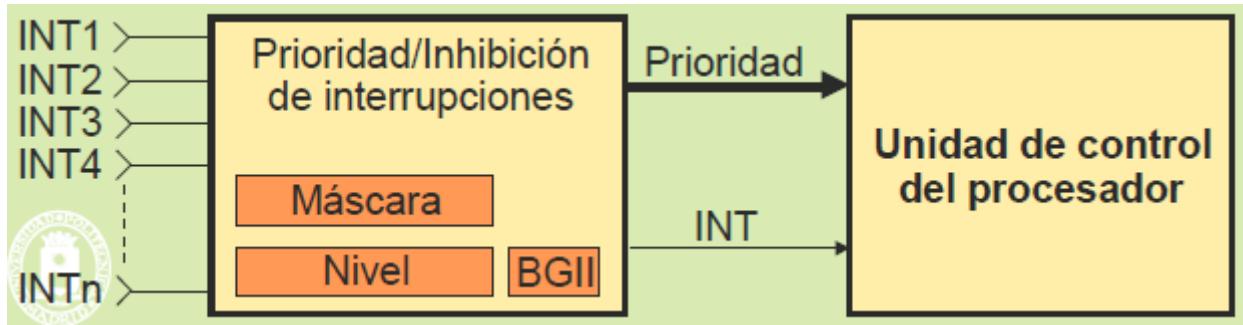
Excepciones HW > Excepciones SW > Interrupciones externas > Llamadas al sistema

El mecanismo de inhibición selectiva sirve para gestionar las interrupciones que llegan a la unidad de control.

Los elementos del mecanismo son:

- BGII (Biestable General de Inhibición de Interrupción).
- Registro de nivel.
- Registro de máscara.

⁵ El vector interrupción redirecciona a la tabla de interrupciones IDT en función del tipo de interrupción (identifica el problema y decide qué algoritmo es el más adecuado para su resolución).



1º La UC recibe una interrupción.

2º El BGII se inhibe.

3º Llega otra interrupción.

4º El registro de nivel puede actuar de dos maneras:

- Inhibe todas las interrupciones.
- Inhibe las interrupciones que tengan prioridad igual o menor que la aceptada. En este caso:

5º El registro de máscara controla las interrupciones aceptadas y en espera mediante un bit por línea de interrupción⁶

6º La rutina de tratamiento de interrupciones restituye los valores de los registros de inhibición de interrupciones.

Los valores de máscara, nivel y BGII sólo pueden ser modificados en modo privilegiado.

13. Enumerar los pasos de que consta el retorno de interrupción (1.3)

El retorno de interrupción es una instrucción máquina que devuelve el estado de ejecución desde el final de la rutina del tratamiento de interrupción hasta el último punto en el que fueron guardados los datos antes de la interrupción (ejemplo de la instrucción máquina: RETI).

1º La rutina de tratamiento de interrupciones finaliza.

2º Se ejecuta la instrucción máquina de retorno RETI.

3º Se restituyen:

- Los valores de registro de estado (niega el bit de interrupción).
- Los valores del contador de programa (devuelve los valores de direccionamiento directo secuencial).

⁶ Las interrupciones en espera a veces se pierden.

1.6. Entrada/Salida

Los mecanismos de Entrada/Salida (E/S) del computador tienen por objeto el intercambio de información entre los periféricos y la memoria o los registros del procesador.

1.6.1. Características de la entrada/salida

14. Enumerar algunas características de la E/S: transferencia por bloques, velocidad de transmisión (1.6.1)



El intercambio de información se desgrana:

Los periféricos y la memoria o registros se intercambian grandes porciones de información.

Las grandes porciones de información se descomponen en transferencias por bloques.

Las transferencias por bloques, a su vez, se descomponen en transferencias elementales.

Las transferencias se realizan por hardware, por lo que se requiere el modo privilegiado (actualmente método DMA)

Los periféricos necesitan un *control permanente*. P.ej. Saber si la impresora está encendida o apagada.

Los periféricos tienen su propio ritmo de funcionamiento. El programa se debe *sincronizar con el periférico*, el ancho de palabra suele ser un byte y sus velocidades de transmisión de información son muy variables.

Algunos periféricos tienen un *tamaño de información privilegiado*. P.ej. Los sectores del disco magnético tienen 512B

15. Explicar el concepto de concurrencia entre E/S y procesador: espera activa y espera pasiva (1.6.4)

Concurrencia: Dos procesos son concurrentes cuando se ejecutan de manera que sus intervalos de ejecución se solapan. Los computadores presentan dos modos básicos de realizar operaciones de E/S:

- **No concurrente:** La E/S programa, en la cual exige que el procesador esté ejecutando un programa de E/S.

- **Concurrente:** E/S por interrupciones y E/S por DMA (Direct Access Memory). Proceso:

- 1º) Envío de la orden al periférico: Se escriben instrucciones máquina en los registros del controlador del periférico.
- 2º) Lectura o escritura de datos.
- 3º) Fin de la operación.

Espera activa: Un programa queda ejecutándose en bucle hasta que ocurra un evento. No es recomendable porque consume tiempo del procesador que podría estar dedicándoselo a otros programas.

Espera pasiva: El programa espera que ocurra un evento y no está ejecutándose. El procesador sigue ejecutando otros programas hasta que reciba un dato que generará una interrupción externa que se encarga de despertarlo.

I6. Conocer las características de los modos básicos de E/S (programada, por interrupciones y DMA) e Identificar en que situaciones tenemos concurrencia entre E/S y procesador (I.6.4)

No concurrente

E/S programada: El periférico exige al procesador que ejecute un programa de E/S.

Concurrente

E/S por interrupciones: El procesador inicia la ejecución del periférico. El controlador del periférico ejecuta la rutina de interrupción donde establece qué dato hará que ejecute su interrupción externa y lo almacena en memoria principal (espera pasiva). Cuando ocurre el evento, la interrupción externa despierta el programa del periférico.

E/S por DMA: El controlador del dispositivo se encarga directamente de transferir datos entre el periférico a la memoria, sin interrumpir al procesador. Una vez terminada la transferencia de todos los datos, el controlador genera una interrupción externa de forma que se sepa que ha terminado.

Otro tipo más evolucionado de DMA → *Canal* (Spiling de las impresoras).

El controlador del periférico crea en memoria principal una cola de trabajos para almacenar los resultados de las órdenes. El canal sólo interrumpe al procesador cuando la cola se encuentra vacía.

TEMA 2 – INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS

2.1. ¿Qué es un Sistema Operativo?

1. Comprender qué es un sistema operativo y lo que motiva su aparición (2.1.intro, 2.1.1)

Un Sistema Operativo (SO) es un programa que tiene encomendadas una serie de funciones cuyo objetivo es simplificar el manejo y la utilización del computador, haciéndolo seguro y eficiente.

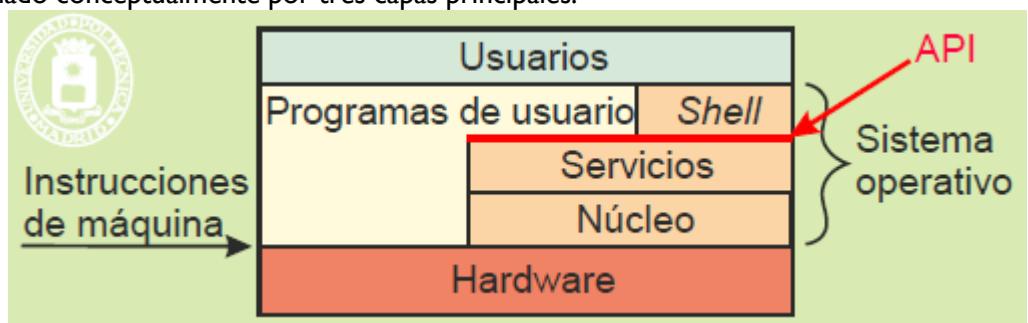
El objetivo del computador es ejecutar programas, pues el objetivo del SO es facilitar la ejecución de esos programas ¿cómo? ReMS (Recursos – Ejecutar – Mandatos – Servicios).

- Gestionando los recursos del computador.
- Ejecutando los mandatos de los usuarios.
- Ejecutando servicios para los programas en ejecución.

2. Conocer cómo se estructura un sistema operativo (núcleo, servicios e intérprete de comandos), así como las funciones que tiene encomendadas (Gestión de recursos, ejecución de servicios y ejecución de comandos) (2.1.intro, 2.1.1)

El Sistema Operativo está formado conceptualmente por tres capas principales:

- i. Núcleo (kernel).
- ii. Capa de servicios o de llamadas al sistema.
- iii. Capa de intérprete de comandos (Shell).



- i. Núcleo (kernel): Gestiona los recursos hardware del sistema y suministra la funcionalidad básica del SO.
- ii. Capa de servicios: Actúa como una máquina extendida a los procesos facilitando la elaboración de programas sobre el SO gracias a una API (Application Programming Interface).
- iii. Shell: Es una interfaz que permite al usuario interactuar con el computador a través de instrucciones⁷.

3. Conocer cómo se activa el sistema operativo mediante interrupciones (llamadas al sistema, interrupciones externas e interrupciones hardware) y qué pasos básicos se llevan a cabo para dicha activación. (2.3.intro)

El computador arranca en modo privilegiado ayudado por el Sistema Operativo. Una vez arrancado, el SO cede el control a los procesos⁸ (programas) y a los periféricos. El SO sólo vuelve actuar cuando se produce una interrupción. Los tipos de interrupciones son:

Interrupciones	Tipo	Descripción
Excepciones software (o hardware síncronas)	síncronas	Son producidas directa o indirectamente por el programa en ejecución: Problemas de ejecución, depuración (punto de ruptura), fallo de página.
Instrucciones máquina de llamada al sistema	síncronas	Son instrucciones que permiten a un programa generar interrupciones. P.ej. las instrucciones TRAP, INT o SC.
Excepciones hardware asíncronas	asíncronas	Son producidas por un error en el hardware. P.ej. Error de paridad en bus, en memoria, fallo de alimentación, temperatura excedida...
Interrupciones externas	asíncronas	Son producidas por elementos externos al procesador. P.ej. El reloj, drivers, otros procesadores...

⁷ Algunos autores no consideran a la Shell como parte del SO porque se suele ejecutar en modo usuario.

⁸ Un proceso es un programa en ejecución, que pasa de estar almacenado en la memoria secundaria a estar cargado en la memoria principal (si no hay suficiente espacio en la RAM, en la memoria virtual/zona de intercambio o swap).

Todas “despiertan” al SO. Comienza la secuencia simplificada de activación del SO:

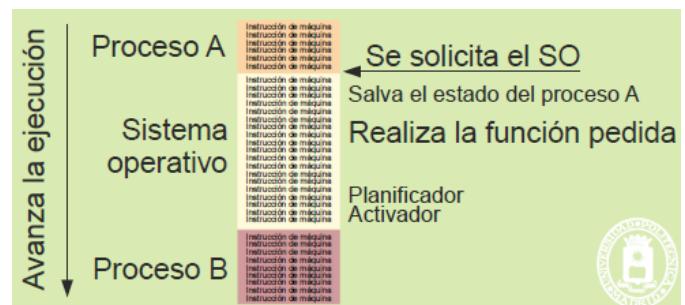
- 1º) El Proceso A está en ejecución.
- 2º) Se recibe una interrupción y el SO pasa a modo privilegiado.

- 3º) El SO salva el estado del Proceso A.

- 4º) El SO recibe el vector de interrupción y entra en acción el planificador que, mediante direccionamiento indirecto, carga el programa de tratamiento de interrupción.

- 5º) El SO finaliza el activador, módulo que se encarga de restituir los valores del proceso B.

- 6º) El computador vuelve al modo usuario.



4. Conocer los diferentes tipos de sistemas operativos: características básicas y evolución (2.4)

Los SO, dependiendo de sus características, pueden ser⁹:

- Según el número de procesos simultáneos: Monotarea/Multitarea – Monoproceso/Multiproceso (es lo mismo).
- Según la forma de interacción con el usuario: Interactivo o por lotes (batch).
- Según el número de usuarios simultáneos: Monousuario/Multiusuario – Personal/De tiempo compartido (lo mismo)
- Según el número de procesadores: Monoprocesador/Multiprocesador.
- Según el número de hilos (threads) por proceso: Monothread/Multithread.
- Según el uso: Cliente, servidor, empotrado, de tiempo real.
- Según la movilidad: Fijos o móviles.

Según el número de procesos simultáneos:

- Monotarea o monoproceso: El SO sólo permite un proceso en cada instante. P.ej. MS-DOS.
- Multitarea o multiproceso o de tiempo compartido: El SO permite que coexistan varios procesos a la vez, repartiendo el tiempo del procesador entre estos procesos. ¡No confundir con multiprocesador!

Según la forma de interacción con el usuario:

- Interactivo: El SO permite que el usuario dialogue con los procesos a través de, por ejemplo, un terminal.
- Por lotes o *batch*: El SO gestiona una cola de trabajos, sin interacción del usuario.

Según el número de usuarios simultáneos:

- Monousuario o personal: El SO está previsto que soporte únicamente un usuario activo a la vez.
- Multiusuario o de tiempo compartido: El SO está preparado para soportar varios usuarios activos a la vez. Por tanto, obligatoriamente, ha de ser multitarea (de tiempo compartido).

⁹ Los tipos no son excluyentes, p.ej. un SO servidor también será, generalmente, multiprocesador, multithread y multiusuario.

Según el número de procesadores:

- Monoprocesador: La máquina sólo cuenta con un procesador.
- Multiprocesador: La máquina cuenta con varios procesadores y el SO tiene que ser capaz de gestionarlos simultáneamente a todos ellos que, estarán ejecutando procesos diferentes.

Según el número de hilos (threads) por proceso:

- Monothread: El SO sólo maneja un único hilo de ejecución
- Multithread: El SO maneja varios hilos de ejecución de procesos simultáneamente.

Según el uso:

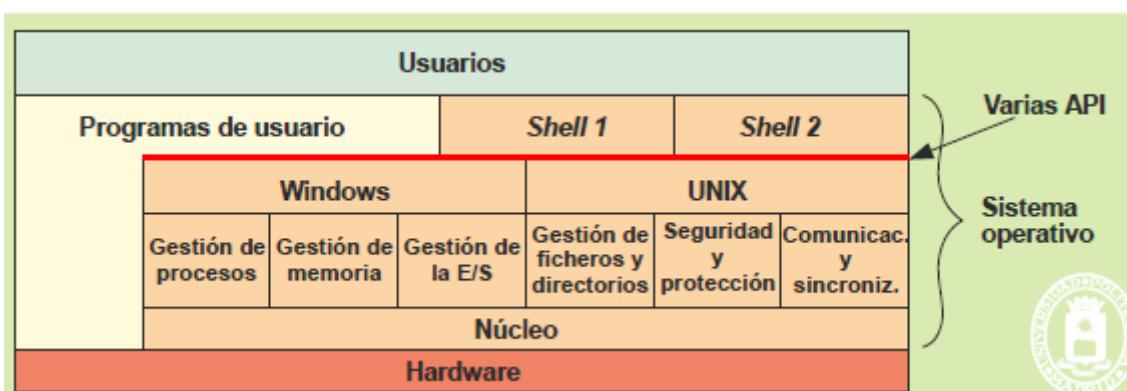
- Cliente: El SO está diseñado para poder acceder a servidores.
- Servidores: El SO está optimizado para que sus usuarios sean sistemas remotos.
- Empotrado o embebido: El SO interacciona con un sistema físico y no con un usuario, suelen tener poca memoria, potencia y disco duro. Están destinados a una función específica o dedicada. P.ej. Electrodomésticos (Linux) y tarjetas inteligentes con criptografía (p.ej. MULTOS, SOLOS (de Schlumberger) y Sun's JavaCard).
- De tiempo real o comunicaciones: El SO garantiza que los procesos se ejecuten en un tiempo predeterminado. P.ej. el guiado de un misil o el control de una central eléctrica.

Según la movilidad:

- Fijos: SO instalados en dispositivos no transportables.
- Móviles: El SO es parecido al de los computadores personales, pero simplificados y optimizados (mejor consumo de la batería, alta frecuencia de encendido y apagado...).

5. Comprender los componentes básicos de los sistemas operativos, como paso fundamental para el desarrollo del resto del temario (2.5.intro) (leer además los puntos 2.5.1 a 2.5.5, se verán de forma detallada en el resto de temas de la asignatura)

El SO está formado por tres capas: Núcleo, capa de servicios y capa de intérprete de comandos (Shell).



Las funciones del núcleo (kernel) son:

- Gestión de recursos (como es el procesador).
- Tratamiento de interrupciones.
- Funciones básicas de manipulación de memoria.

Las funciones de la capa de servicios son¹⁰:

- Gestión de procesos.
- Gestión de memoria.
- Gestión de E/S.
- Gestión de ficheros y directorios.
- Gestión de comunicación y sincronización entre procesos.
- Gestión de seguridad frente ataques del exterior y protección interna (modo usuario/privilegiado).

Las funciones de la capa de intérprete de comandos es la interactuación entre el usuario y el SO¹¹.

Leer los puntos 2.5.1. a 2.5.5. (páginas 44 a 51 del PDF)

6. Conocer las diferentes formas de estructurar un sistema operativo: monolítico, estructurado por capas, modelo cliente-servidor y máquina virtual (2.9.1)

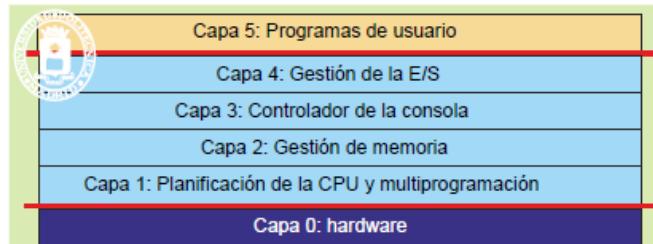
Un SO se puede estructurar de diversas formas para realizar sus funciones:

SO sin estructura:

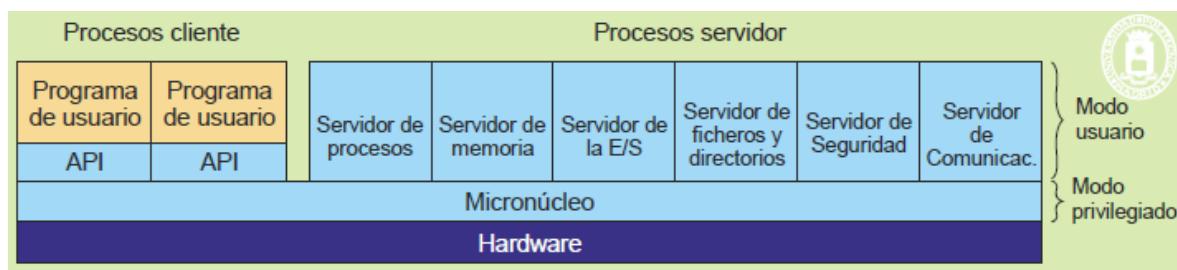
- Monolítico: Todos los componentes se encuentran integrados en un único gran programa (SO) que se ejecuta en un único espacio de direcciones, no oculta información y se tiene que controlar en modo privilegiado. Ej. UNIX, MS-DOS...

SO con estructura por capas:

- Sistemas por capas: El SO se organiza en una jerarquía de capas. Cada capa es independiente y se comunica con las colindantes mediante una interfaz, desconociendo como se implementan. Facilita enormemente la depuración y verificación del sistema.



- Modelo cliente-servidor: El SO divide los servicios y funciones en dos partes: Una parte servidor con una estructura de micronúcleo (o nanonúcleo) la cual ejecuta los procesos en modo privilegiado; y una parte cliente la cual ejecuta los procesos en modo usuario. Facilita el desarrollo, la depuración y la fiabilidad en la ejecución, pero es más lento porque cada proceso servidor utiliza espacios de memoria diferentes:



Máquina Virtual

La máquina virtual es un monitor dentro de una máquina real que se comporta como esta última. La máquina virtual contiene una réplica virtual¹² del hardware con recursos propios y se comunica con la máquina real para solicitarle otros recursos, como por ejemplo la comunicación con los periféricos físicos.

¹⁰ En la máquina pueden coexistir diferentes servicios de máquinas extendidas que no se comunicarán (p.ej. Windows y Linux).

¹¹ En la máquina pueden coexistir varias Shells, unas textuales y otras gráficas.

¹² La máquina virtual replicada puede ser idéntica a la real (p.ej. VMware) o una máquina distinta (JVM -Java Virtual Machine-).

Ventajas MV:

- Añade un nivel de multiprogramación.
- Añade un nivel de aislamiento que favorece la seguridad.

Desventajas MV:

- Hay que recompilar los programas para cada máquina virtual.
- Sobre carga computacional, puede hacer más lenta la ejecución.

Exokernel: Es una capa adicional del kernel encargada de asignar recursos a las máquinas virtuales, lo hace en modo privilegiado.

7. Conocer la evolución de los sistemas operativos y las circunstancias tecnológicas que la motivaron (2.10) (solo leer)

Leer de la página 64-69 (PDF)

TEMA 3 – PROCESOS

3.1. Concepto de proceso

1. Entender el concepto de proceso, concepto fundamental en el desarrollo de los sistemas operativos y unidad básica de procesamiento considerada por el sistema operativo (3.1)

Concepto general

Un proceso es un programa puesto en ejecución por el sistema operativo.

Concepto más preciso

Un proceso es la unidad de procesamiento gestionada por el sistema operativo.

Entorno de proceso (Variables de entorno en Windows)

Son tablas NOMBRE=VALOR almacenadas en la pila del proceso. El SO busca los programas ejecutables en la tabla:

PATH=/usr/bin /home/pepe/bin

TERM=vt100

HOME=/home/pepe

PWD=/home/pepe/libros/primero

PATH → Lista de directorios en los que el SO busca los programas ejecutables.

TERM → Tipo de terminal para interpretar la instrucción y la interacción del usuario.

HOME → Directorio inicial asociado al usuario.

PWD → Directorio de trabajo actual.

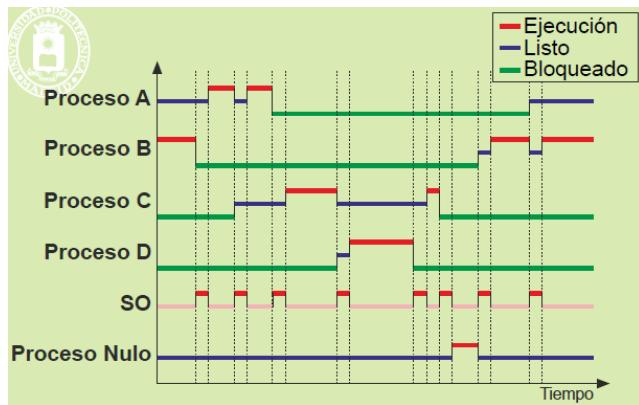
2. Entender qué significa un sistema operativo multitarea, así como los conceptos que involucra: multiprogramación, memoria virtual, paralelismo E/S y procesador, etc (3.2)

Según el número de procesos simultáneos:

- Multitarea o multiproceso o de tiempo compartido: El SO permite que coexistan varios procesos a la vez, repartiendo el tiempo del procesador entre estos procesos. ¡No confundir con multiprocesador!

Características de la multitarea

- **Paralelismo E/S y procesador**



En un sistema multitarea se aprovechan las fases de entrada/salida de unos procesos para realizar las fases de procesamiento de otros.

El procesador está constantemente ejecutando instrucciones, que pueden ser:

- Ejecución.
- Listo.
- Bloqueado.
- Nulo.

- **Multiprogramación**

La multiprogramación se refiere al número de procesos activos que mantiene un sistema multitarea.

A mayor grado de multiprogramación, más necesidad de memoria y menor rendimiento del computador. La memoria que se puede utilizar es:

- **Memoria real**

En los sistemas con memoria real los procesos almacenan en la memoria principal disponible el programa completo en ejecución.

• Memoria virtual

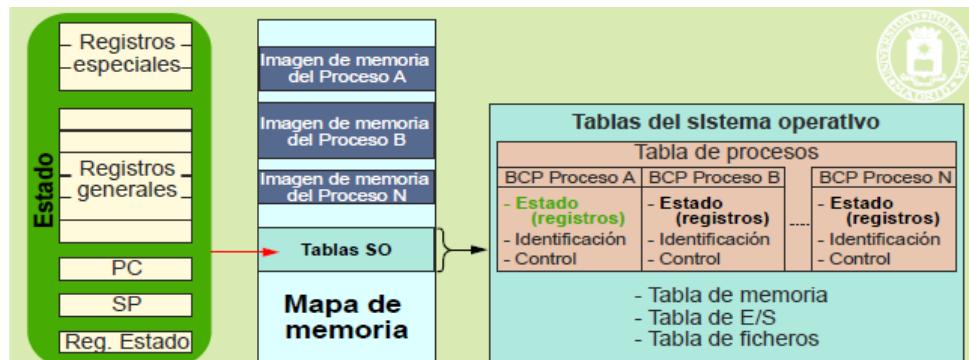
En los sistemas con memoria virtual los procesos almacenan en memoria principal un conjunto residente que, al aumentar el número de procesos se hacen cada vez más pequeños los conjuntos residentes.

Puede llegar un punto, en el que el conjunto residente sea tan pequeño que no represente adecuadamente el conjunto del trabajo y se produzca fallos de **hiperpaginación** (trashing) que el SO debe solucionar migrando las páginas.

3. Conocer qué información alberga un proceso para ser considerado como unidad básica de procesamiento por el sistema operativo, así como la necesidad de ese tipo de información (planificación, almacenamiento en memoria, etc.) (3.3)

El proceso es la unidad de procesamiento gestionada por el sistema operativo. Tiene una serie de elementos que se organizan en tres grupos:

- I) Estado del procesador
- II) Imagen de memoria
- III) Tablas del Sistema Operativo



- I) Estado del procesador

El proceso en ejecución reside en los registros y varía de acuerdo con el flujo de instrucciones máquina ejecutadas. El proceso cuando termina de ejecutarse, su estado se guarda en el Bloque de Control de Proceso (BCP). Por tanto, cuando un proceso está ejecutándose, el BCP no tiene actualizado el estado de ese proceso y, si surge alguna interrupción, el SO debe salvar el estado de ese proceso en el BCP.

- II) Imagen de memoria del proceso

4. Conocer el ciclo de vida de un proceso: creación (3.4.1), interrupción (3.4.2), activación (3.4.3) y terminación (3.4.4)

La vida de un proceso se descompone en las siguientes fases:

- 1º) El proceso se crea:
 - Al arrancar el sistema operativo (proceso padre de todos los procesos hijo).
 - Otro proceso (padre) solicita al SO el servicio de creación de otro proceso (hijo).
- 2º) El proceso se ejecuta en rachas de procesamiento y rachas de espera.
- 3º) El proceso termina o muere.

1º) Creación

1. El proceso padre solicita al SO el servicio de creación de otro proceso.
2. El SO asigna un espacio de memoria de varias regiones.
3. El SO selecciona un BCP libre de la tabla de procesos.
4. El SO rellena el BCP con los valores del proceso creado (memoria asignada, valores iniciales de registro...)
5. El SO carga en:
 - Región de texto: El código más las rutinas de sistema para su ejecución.
 - Región de datos: Datos iniciales contenidos en el fichero objeto.
 - Región de pila: Entorno del proceso y argumentos de las rutinas.

Al finalizar, se obtiene un proceso listo para ejecutar y, cuando lo considere oportuno, el activador lo ejecuta.

2º Interrupción del proceso

- 1º Un proceso está en ejecución, por tanto, su información se encuentra en los registros de la máquina.
- 2º Se produce una interrupción (Sw, Hw, externa o llamada al sistema).
- 3º El SO guarda los registros del proceso anterior en el BCP.
- 4º El SO cambia el bit de estado (de modo usuario a modo privilegiado) y el contador de programa para comenzar a ejecutar el tratamiento de interrupción.
- 5º El SO recibe el vector de interrupción y entra en acción el planificador que, mediante direccionamiento indirecto, carga el programa de tratamiento de interrupción.
- 6º El SO finaliza el activador, módulo que se encarga de restituir los valores del proceso (p.ej. con la instrucción RETI). La instrucción restituye los valores del registro de estado y el contador del programa.
- 7º La instrucción restituye:
 - El registro de estado:
 - o El bit de estado cambiando de modo privilegiado a modo usuario.
 - o El bit de inhibición de interrupción, según como los tenía antes de la interrupción.
 - El contador de programa: Se vuelve a la siguiente instrucción máquina del proceso interrumpido.

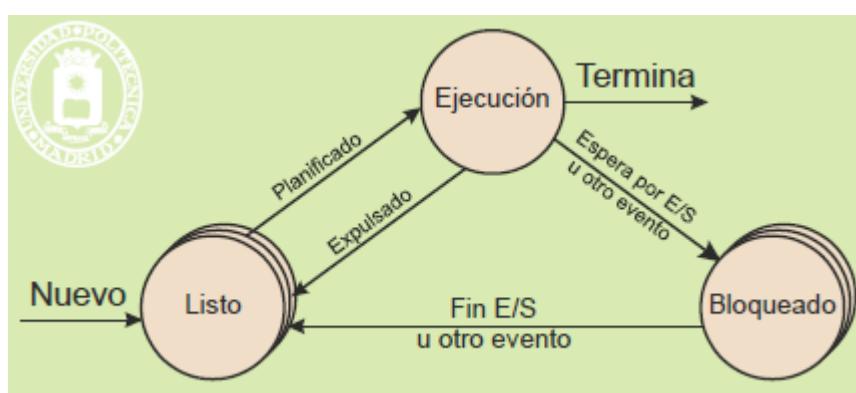
3º Terminación del proceso

Los recursos del sistema son repartidos por el SO a los procesos, por ello, se contabiliza mediante un contador. Cuando el proceso termina o muere:

- El contador llega a “0” → El recurso se libera.
- El contador no llega a “0” → El SO decrementa el valor de dicho contador, pero no se libera.

5. Conocer el conjunto de estados por los que un proceso puede pasar y las circunstancias que lo conducen a dicho estado. Entender qué significa cada estado, cómo se entra y se sale de dicho estado, qué recursos se tiene asignado en cada estado, etc. (3.4.5, 3.4.6)

Los procesos en un sistema multitarea pueden tener tres estados básicos:



Un proceso está en estado **listo** cuando puede entrar en la fase de procesamiento (ver 3.4.4.1. Creación) mientras, se encuentra almacenado en el BCP. El planificador del SO decide cuando se ejecutará.

Un proceso está en estado de **ejecución** o en fase de procedimiento cuando reside en los registros del procesador. El proceso vuelve a listo cuando el SO decide que lleva mucho tiempo ejecutándose o se carga en memoria un proceso más prioritario.

Un proceso está **bloqueado** cuando está esperando a que ocurra un evento (espera pasiva) y no puede continuar ejecutándose hasta que ocurra un evento, mientras se almacena el proceso en el BCP.

Los procesos por lotes o *batch* que entran en el sistema pueden tener otros dos estados aparte de los tres vistos:

Estado en espera: Se encuentran en una lista de procesos y serán ejecutados cuando el SO disponga de los recursos necesarios.

Estado suspendido: El SO retira algunos procesos a la zona de intercambio para evitar que se produzca la hiperpaginación (se reduzca demasiado el conjunto residente en memoria).

6. Conocer el concepto de cambio de contexto (3.4.7)

El cambio de contexto consiste en pasar de ejecutar un proceso A a ejecutar otro proceso B. El cambio de contexto exige dos cambios de modo. El primer cambio de modo viene producido por una interrupción, pasándose a ejecutar el sistema operativo. El segundo cambio de modo lo realiza el sistema operativo al activar el proceso B.

Hay dos cambios de contexto:

A) Cambio de modo por interrupción (proceso → SO)

El proceso A se está ejecutando y llega una interrupción (Ver 3.4.4. Interrupción de proceso 1°-5°).

B) Cambio de modo por activación (SO → proceso)

El SO activa el proceso B (Ver 3.4.4. Interrupción de proceso 6°-7°).

7. Conocer los conceptos de usuario (grupo) real y efectivo de un proceso (3.4.8)

Usuarios

UID o usuario real: Es el usuario responsable de la ejecución de un proceso.

EUID o usuario efectivo: Es el usuario sobre el que se está ejecutando un proceso. Por ejemplo: Creación de ficheros, cambiar permisos...

Grupos

GID o grupo real: Es el grupo responsable de la ejecución de un proceso.

EGID o grupo efectivo: Es el grupo sobre el que se está ejecutando un proceso.

8. Conocer el concepto de hilo y lo que le diferencia de proceso (3.8.intro)

Un proceso es la unidad de procesamiento gestionada por el Sistema Operativo y está formado por:

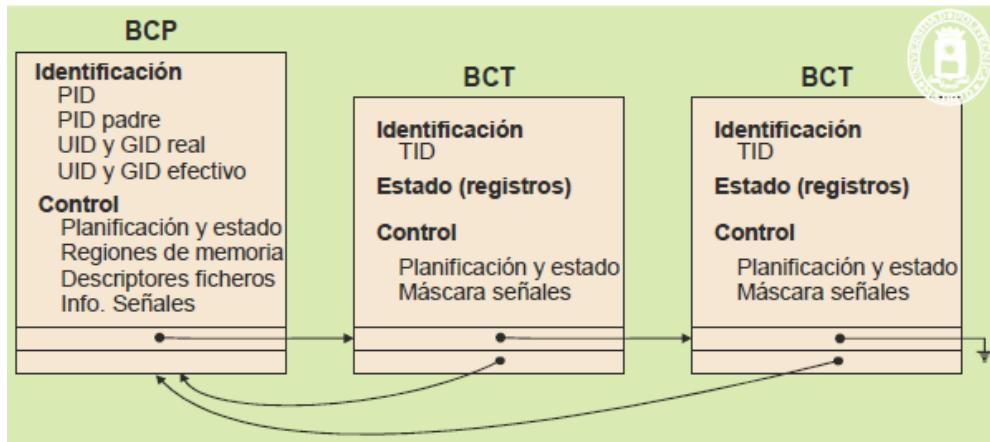
- Activo: Conjunto de todos los bienes y derechos asignados al proceso.
- Flujo de ejecución: Conjunto ordenado de instrucciones máquina del programa que va ejecutando el proceso.

Los hilos, threads o procesos ligeros es la secuencia del flujo de ejecución. Los elementos ligados al thread son:

- Estado del procesador.
- Estado de la ejecución (listo, bloqueado o en ejecución).
- Pila.

Todos los threads del mismo proceso comparten el activo.

Los threads de un mismo proceso comparten BCP (direcciones de memoria de proceso). A su vez, los threads de un mismo proceso necesitan una estructura BCT (Bloque de Control de Thread) para identificar el thread, asignarle un registro y controlar el estado y la máscara de señales.



9. Conocer y usar los servicios del sistema operativo UNIX asociados a la gestión de procesos: creación, terminación, etc. (3.13.1: Identificación de procesos, creación de procesos y terminación de procesos)

Parte práctica

TEMA 4. INTERBLOQUEOS

1. Obtener una idea intuitiva de qué son los interbloqueos y las distintas estrategias utilizadas para tratarlos (4.1)

Interbloqueos o bloqueos mutuos son situaciones en los sistemas donde varios procesos necesitan utilizar simultáneamente un mismo conjunto de recursos.

Las técnicas de tratamiento de interbloqueos se dividen en tres categorías:

- Prevención: El sistema se diseña con una serie de características que impiden las situaciones de interbloqueo.
- Evitación: Supervisa la evolución del sistema para evitar “estados no seguros” en un futuro.
- Detección y recuperación: En el sistema se producen interbloqueos que son resueltos por diversos métodos.

2. Clasificar los tipos de recursos y conocer cuáles son de importancia en relación al estudio de los interbloqueos (4.2)

Los recursos pueden clasificarse en:

- *Físicos (memoria, procesador, disco, red...) o *lógicos (ficheros, bases de datos...).
- *Reutilizables (hardware, ficheros...) o consumibles (mensajes o señales).
- *Exclusivos (procesador...) o compartidos (fichero abierto en modo lectura).
- Simples (sólo existe un único ejemplar o instancia del recurso) o múltiples (varias unidades del mismo).
- Expropiables (procesador, los datos se salvaguardan en el BCP) o no expropiables.

(*los más importantes para el estudio de los interbloqueos)

3. Conocer que elementos son de importancia para modelar el sistema (4.3.intro)

4. Utilizar un grafo de asignación de recursos (GAR) para representar los sistemas a estudiar (4.3.1)
5. Utilizar la notación matricial para representar los sistemas a estudiar (4.3.2)
6. Definir qué es un interbloqueo (4.4.intro)
7. Enumerar y explicar las condiciones necesarias (condiciones de Coffman) para la existencia de interbloqueos (4.4.1)
8. Conocer las condiciones necesarias y suficientes para la existencia de interbloqueos: comprender el concepto de reducción (4.4.2)
9. Conocer bajo qué circunstancias las condiciones de Coffman son suficientes para que exista interbloqueo (4.4.2)
10. Identificar algunas situaciones de interbloqueo a partir de su GAR (4.4.2)
11. Explicar en qué consiste la técnica de detección y recuperación (4.5.intro)
12. Entender y utilizar el algoritmo de detección utilizando tanto la notación matricial como el GAR (4.5.1)
13. Conocer cuándo se debería activar el algoritmo de detección (4.5.1)
14. Conocer cómo se puede recuperar un sistema que presenta un interbloqueo (4.5.2)
15. Explicar las técnicas de prevención de interbloqueos (4.6)
16. Explicar en qué consiste la técnica de evitación de interbloqueos (4.7.intro)
17. Comprender qué es un estado seguro (4.7.1)
18. Entender y utilizar el algoritmo del banquero (4.7.2)