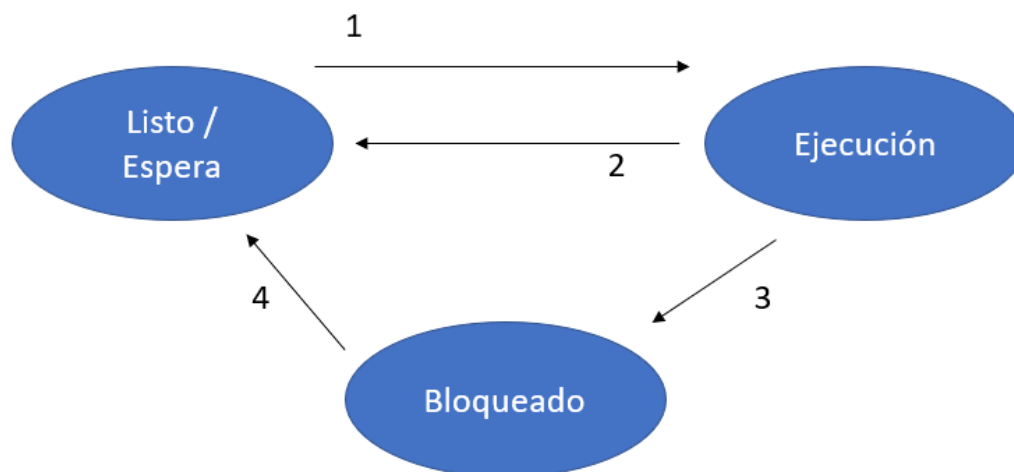


Sistemas Informáticos

Tema 2 – Gestión de Procesos

Gestión de procesos

Un **proceso** es un sistema en ejecución.



Algoritmos de gestión de procesos:

Los **algoritmos apropiativos** no permiten la expulsión de procesos para tratar otros, es decir, no procederá a tratar otro proceso hasta que acabe con el primero que tiene.

Los **algoritmos no apropiativos** SI permiten la expulsión de procesos, por lo que podrán parar un proceso que se esté gestionando para tratar otro con mayor prioridad.

Tipos de algoritmos:

FIFO: PRIMERO EN ENTRAR, PRIMERO EN SALIR

PROCESOS	LLEGADA	T. EJECUCIÓN	T.espera	T.respuesta
A	0	8	0	8
B	3	4	5	9
C	6	2	6	8
D	10	3	4	7
E	15	6	2	8
F	24	4	0	4
		27	2,833333	7,33333333

PROCESOS	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
A	x	x	x	x	x	x	x	x																				
B				e	e	e	e	e	x	x	x	x																
C							e	e	e	e	e	e	x	x														
D											e	e	e	e	x	x	x											
E																e	e	x	x	x	x	x	x					
F																									x	x	x	x

SJF: PRIMERO EL MÁS CORTO

PROCESOS	LLEGADA	T. EJECUCIÓN	T.espera	T.respuesta
A	0	8	0	8
B	3	4	7	11
C	6	2	2	4
D	10	3	4	7
E	15	6	2	8
F	24	4	0	4
		27	2,5	7

PROCESOS	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
A	x	x	x	x	x	x	x	x																				
B				e	e	e	e	e	e	e	x	x	x	x														
C							e	e	x	x																		
D											e	e	e	e	x	x	x											
E																e	e	x	x	x	x	x	x					
F																									x	x	x	x

SRTF: PRIMERO EL DE TIEMPO RESTANTE MÁS CORTO

PROCESOS	LLEGADA	T. EJECUCIÓN	T.espera	T.respuesta
A	0	8	9	17
B	3	4	0	4
C	6	2	1	3
D	10	3	0	3
E	15	6	6	12
F	24	4	0	4
		27	2,666667	7,16666667

PROCESOS	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
A	x	x	x	e	e	e	e	e	e	x	e	e	e	x	x	x	x											
B				x	x	x	x																					
C							e	x	x																			
D																												
E																												
F																												

ROUND ROBIN (R-R): OPERACIÓN POR RONDAS

El **Quantum** es el tiempo de gestión de cada proceso.

						QUANTUM		3																													
PROCESOS	LLEGADA	T. EJECUCIÓN	T.espera	T.respuesta	PROCESOS	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27				
A	0	8	12	20	A	x	x	x	e	e	e	e	e	x	x	x	e	x	e	e	e	e	e	x	x												
B	3	4	5	9	B				x	x	x	e	e	e	e	e	x																				
C	6	2	0	2	C							x	x																								
D	10	3	2	5	D								x	x			e	e	x	x	x																
E	15	6	2	8	E																	x	x	x	e	e	x	x	x								
F	24	4	0	4	F																										x	x	x	x			

Concurrencia de procesos

En los sistemas de tiempo compartido (aquellos con varios usuarios, procesos, tareas, trabajos que reparten el uso de CPU entre estos) se presentan muchos problemas debido a que los procesos compiten por los recursos del sistema.

- **Condiciones de Carrera o Competencia:** Ocurre cuando dos o más procesos acceden a un recurso compartido sin control, de manera que el resultado combinado de este acceso depende del orden de llegada (Acceso al mismo recurso al mismo tiempo, EJ ingresar y sacar dinero a la vez).
- **Postergación o Aplazamiento Indefinido(a):** Consiste en el hecho de que uno o varios procesos nunca reciban el suficiente tiempo de ejecución para terminar su tarea (EJ proceso no libera recurso y otro se queda bloqueado indefinidamente a la espera).
- **Condición de Espera Circular:** Ocurre cuando dos o más procesos forman una cadena de espera que los involucra a todos (Cena de los filósofos).

IMPORTANTE: Solución a **cena de los filósofos** o **babero durmiente** mediante **SEMÁFOROS** para dar turnos de uso a los “recursos”.

Gestión de memoria

Los procesos van a utilizar la memoria del sistema para ejecutarse, hay varias maneras de repartir los procesos en la memoria para su uso, intentando tener la menor fragmentación del sistema para no saturarlo/ralentizarlo.

La **fragmentación** es el hueco que dejan los procesos en las particiones de memoria al ocuparla, puede ser que no deje ya que el proceso puede ocupar el tamaño exacto de la partición de memoria, pero en otros casos puede dejar espacio de la partición sin utilizar.

La **recolocación** permite que el S.O. cambie la posición de un proceso en la memoria para no sufrir una fragmentación interna.

La **protección** se enfoca en la seguridad y privacidad de los procesos y los datos que tienen en la memoria.

Para repartir la memoria, esta se particiona de 2 maneras diferentes:

- **Particiones estáticas de tamaño fijo:** La partición se establece en el momento de crearla y no puede ser cambiada más tarde.
- **Particiones estáticas de tamaño variable:** El tamaño de la partición puede ser modificado después de crearla.

La **paginación** se basa en dividir la memoria en bloques de igual tamaño llamados páginas donde el sistema operativo mantiene una tabla de páginas donde se indica qué páginas están en la memoria y dónde se encuentran, permitiendo que el S.O. acceda a cualquier parte de los procesos de memoria al instante.

La **segmentación** divide un proceso en varias partes llamadas segmentos, cada uno de estos representa una función o porción de los datos del proceso. El S.O. los mantiene en una tabla que indica que segmentos están en la memoria y dónde se encuentran, permitiendo acceder a ellos al instante.

Algoritmos de gestión de memoria

Para asignar los procesos en estas particiones de memoria, se suelen utilizar varios **algoritmos**:

- **Primero en ajustarse:** Asigna los procesos sin importar el tamaño que tengan en la primera partición en que entren sin importar el tamaño de la partición (ejemplo, un proceso de 13k lo incluye en una partición de 30k por ser la 1ª disponible cuando hay una de 13k justos, pero en la 4ª posición).
- **Mejor en ajustarse:** Primero se revisará el tamaño de las particiones y luego asignará el proceso a la partición que produzca una menor partición al ocuparse por el proceso (ejemplo, si tenemos un proceso de 13k y 2 particiones, una de 30k y otra de 14k, se asignará el proceso a la de 14k ya que solo causará 1k de fragmentación).
- **El peor en ajustarse:** Primero revisará el tamaño de las particiones y luego asignará los procesos a las particiones que mayor fragmentación tengan (ejemplo, si tenemos 2 particiones, una de 30k y otra de 14k y tenemos un proceso de 13k, lo asignará a la de 30k produciendo 17k de fragmentación).
- **Siguiente en ajustarse:** Asigna los procesos en la primera partición de memoria en la que se ajuste, dando igual la fragmentación que pueda provocar. **A DIFERENCIA DE “PRIMERO EN AJUSTARSE”, ESTE CONTINUA DESDE DONDE DEJO EL PROCESO ANTERIOR.**

Primero en ajustarse		Mejor ajustarse		Peor en ajustarse		Siguiente en ajustarse	
20k	2k	20k		20k	2k	20k	2k
14k	13k	14k	13k	14k		14k	13k
20k		20k		20k	13k	20k	
12k		12k		12k		12k	
7k		7k		7k		7k	
2k		2k	2k	2k		2k	
15k		15k		15k		15k	

Gestión de Datos

Un **fichero** es un mecanismo de abstracción que sirve como unidad lógica de almacenamiento de información. El fichero agrupa una colección de informaciones relacionadas entre sí y definidas por su creador. A todo fichero le corresponde un nombre único que lo identifique entre los demás ficheros.

Es necesario que el sistema operativo cuente con un sistema que se encargue de administrar la información almacenada en los dispositivos en forma de ficheros: de esto se encargan los **sistemas de ficheros**.

Un **sistema de ficheros** es el aspecto más visible de todo sistema operativo y existe por razones tecnológicas, ya que no hay memoria principal lo suficientemente grande como para no necesitar de almacenamiento secundario. Surge debido a la necesidad del sistema operativo de poder gestionar la información de forma eficiente y estructurada, además de establecer unos parámetros de seguridad y protección en entornos críticos.

Cualquier sistema operativo distingue entre varios **tipos básicos de ficheros**, que será la clasificación que consideremos nosotros:

- **Regulares o Normales:** Aquellos ficheros que contienen datos (información).
- **Directorios:** Aquellos ficheros cuyo contenido es información sobre otros ficheros, normalmente un vector de entradas con información sobre los otros ficheros.
- **De dispositivo:** Existen dispositivos cuya E/S se realiza como si fuesen ficheros, por lo tanto, es razonable asociarles ficheros para simplificar y hacer más transparente el intercambio de información con dichos dispositivos.

Aunque se imponga al sistema operativo el desconocimiento del tipo de ficheros que manipula, sí se hace una distinción de este de forma transparente: a través de las extensiones del nombre. Mediante la extensión del nombre del fichero (una cadena de caracteres de pequeña longitud) se puede determinar el tipo del fichero.

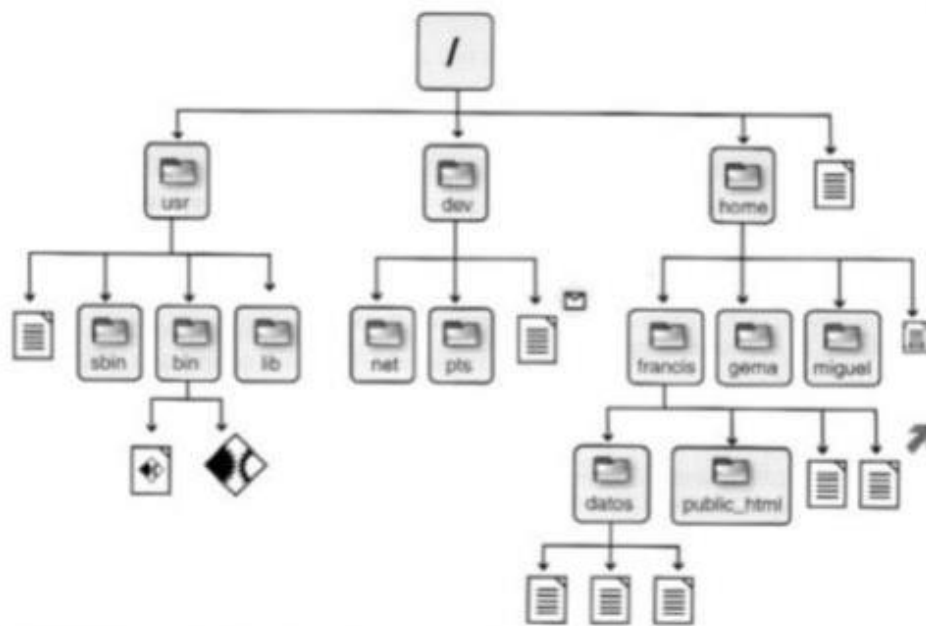
De este modo, aunque el sistema operativo no conozca internamente la estructura de los ficheros, si es capaz de manejarlos eficientemente gracias al uso de estas extensiones. Esta es la aproximación de los sistemas operativos de Microsoft.

Estructuras de directorios

Hay varias formas de organizar los directorios sobre un disco:

- Directorio de un nivel. En este tipo de organización solo se permite un nivel de directorio.
- Directorio de dos niveles. En este tipo de organización, un directorio puede incluir dentro otro directorio, pero éste ya no puede incluir otro más.
- Directorio con estructura arborescente. Prácticamente no tiene limitaciones. Un directorio puede incluir otros directorios, sin importar su número, y estos nuevos directorios pueden contener otros directorios.

El árbol es de raíz única, de modo que cada fichero tiene un único nombre de ruta de acceso. Un directorio (o subdirectorio) contiene a su vez ficheros y/o subdirectorios, y todos los directorios poseen el mismo formato interno. Las entradas del directorio indican si el objeto referenciado es un fichero o un subdirectorio.



Se define **directorio padre** de un fichero o subdirectorio como el directorio en el que se encuentra su entrada de referencia. Cada fichero o directorio (a excepción del directorio raíz) posee un único directorio padre.

Se define **directorio hijo** de un directorio como el directorio que tiene por padre al primero. Un directorio puede contener múltiples directorios hijos, y cada directorio (a excepción del raíz) es hijo de algún otro.

¿Qué se almacena realmente en una entrada de directorio? La siguiente tabla muestra algunas de estas informaciones, aunque en un sistema de ficheros concreto pueden no estar todas las que son ni ser todas las que están:

Nombre del fichero	El nombre simbólico del fichero.
Tipo de fichero	Para aquellos sistemas que contemplan diferentes tipos.
Ubicación	Un puntero al dispositivo y a la posición del fichero en el dispositivo.
Tamaño	Tamaño actual del fichero (en bytes, palabras o bloques) y el máximo tamaño permitido.
Posición actual	Un puntero a la posición actual de lectura o escritura sobre el fichero. Su lugar exacto en el dispositivo.

Protección	Información referente a los permisos de acceso al fichero.
Contador de uso	Número de procesos que están utilizando simultáneamente el fichero.
Hora y fecha	De creación, último acceso, etc.
Identificación	Del proceso creador del fichero, del último que accedió al fichero, en qué forma lo hizo, etc.

La **ruta absoluta** de un fichero se define desde la raíz o dirección padre del equipo, mientras que la **ruta relativa** se hace desde una carpeta en específico del directorio.

1. Ruta absoluta para :

- datos: /home/francis/datos
- net: /dev/net
- sbin: /usr/sbin

2. Ruta relativa desde net para:

- francis: ../../home/francis
- datos: ../../home/francis/datos
- lib: ../../usr/lib

Métodos de asignación

La forma de ubicar los ficheros físicamente sobre el disco o, dicho de otro modo, los distintos métodos existentes para asignar espacio a cada fichero dentro del disco se define en 3 tipos:

- Asignación **contigua**: El método de asignación contigua funciona de forma que cada fichero ocupe un conjunto de bloques consecutivos en el disco. Como se dijo en apartados anteriores, cada bloque del disco posee una dirección que confiere una organización lineal al conjunto de bloques (los bloques están seguidos uno detrás de otro). De esta forma el acceso al bloque i+1 desde el bloque i no requiere normalmente movimiento alguno de la cabeza de lectura/escritura y, cuando sí lo requiere, se trata sólo de saltar a la pista siguiente.
- Asignación **enlazada**: Permite almacenar los ficheros de forma no continua. Siguiendo este esquema, cada fichero no es más que una lista enlazada de bloques, que pueden encontrarse en cualquier lugar del disco. La entrada del directorio posee únicamente un puntero al primer bloque y un puntero al último. Cada bloque, a su vez, contendrá un puntero al siguiente bloque.
- Asignación **indexada**: Permite gestionar los ficheros mediante un **bloque índice**. A cada fichero le corresponde su propio bloque índice, que no es más que una tabla de direcciones de bloques, donde la entrada i apunta al bloque i del fichero. La entrada de un fichero en el directorio sólo necesita mantenerla dirección del bloque índice para localizar todos y cada uno de sus bloques.

Administración del espacio libre:

En un sistema informático, los ficheros se crean y se destruyen con frecuencia. Debido a que el espacio de disco no es ilimitado, se hace necesario reutilizar el espacio ocupado por ficheros que han sido borrados para almacenar nuevos ficheros.

El sistema operativo debe mantener, pues, una lista de bloques (clusters) libres (no asignados a ningún fichero).

A la hora de crear un fichero, el sistema operativo examina esta lista en busca de bloques libres, los asigna y elimina dichos bloques de la lista. Cuando se borra un fichero de forma efectiva, los bloques que éste ocupaba se añaden a la lista de bloques libres.

- Se indica **P** si son **1s** y **H** si son **0s**.
- Se indica el nº de la posición en la que empieza.
- Se indica el nº de las posiciones que ocupa.
- Se indica con una flecha → si se avanza en el sistema de archivos o con una **X** si ya no hay más archivos.

P 0 2 → H 2 4 → P 6 2 → H 8 6 → P 14 3 → H 17 2 → P 19 6 → H 25 5 → P 28 5 X

Una forma similar de no perder de vista a los bloques libres, sin el engorro (en cuanto a espacio ocupado) que supone el mantener una lista en memoria, es la de implementar un mapa de bits (también llamado vector de bits o, en inglés, bitmap). En este esquema, cada bloque del disco se representa por medio de un bit, que estará puesto a un valor lógico concreto si el bloque está asignado a algún fichero y a su complementario cuando el bloque esté libre.

11 0000 11 000000 111 00 111111 000 11111

MBR y GPT

Los discos duros se deben particionar para organizar la información convenientemente, poder almacenar programas, datos e instalar sistemas operativos. Llamamos **partición** a una división del espacio de almacenamiento de forma contigua de un disco duro.

Los discos duros de nuestro equipo pueden estructurar sus particiones atendiendo a dos estándares:

- **MBR** (Master Boot Record o Registro de Arranque Maestro): Es el esquema de particionamiento de los sistemas con estándar BIOS. Es más antiguo, pero aún se sigue empleando por su compatibilidad con sistemas operativos.
- **GPT** (GUID Partition Table o tabla de particiones de GUID): Esquema de particionamiento de los sistemas con estándar UEFI. A cada partición se le asigna un identificador global único (GUID). Este estándar mejora al estándar BIOS tradicional y solventa sus limitaciones:
 - **Permite hasta 128 particiones diferentes de S.O.**
 - **Soporta discos de mayor tamaño (MBR limitado a 2,2 TB).**
 - **Permite trabajar en 32 y 64 bits.**
 - **Inicio de sistema más rápido.**
 - **Mayor seguridad frente a MBR BIOS.**
 - **Conexión directa a internet.**
 - **Interfaz más amigable para el usuario.**
 - **Tiene un gestor de arranque propio, no está vinculado a ningún sistema operativo.**

Gestor de arranque Windows

El gestor de arranque Windows es **BOOTMGR** (Windows Boot Manager), hace uso de un almacén de datos de configuración de arranque (BDC, Boot Configuration Data) y para realizar modificaciones en la configuración se emplea el programa **bcdedit.exe** (mediante opciones en línea de comandos).

Para una configuración simple se puede emplear el programa **msconfig** y otras aplicaciones que permiten una edición gráfica del arranque de manera más sencilla.

Windows Boot Manager se encarga de:

- Cargar las aplicaciones de arranque de Windows.
- Mostrar el menú de selección por el usuario, donde se muestran las opciones de arranque (si es que hay más de un S.O. reconocido con el gestor de arranque).
- Localizar el cargador del sistema operativo (OS Loader), seleccionado en la opción anterior.
- Cargar el cargador del sistema operativo y transferirle el control.

Gestor de arranque Linux

Linux normalmente emplea el gestor GRUB 2, es muy potente y flexible, puede lanzar la mayoría de los sistemas operativos.

Los archivos y directorios que intervienen en el funcionamiento y gestión de GRUB 2 en Ubuntu son:

- Archivo de configuración principal de GRUB 2 (/boot/grub/grub.cfg). Es un producto de varios scripts y no debe ser modificado directamente.
- La configuración del menú gráfico durante el arranque se gestiona mediante el archivo /etc/default/grub. Se puede modificar el tiempo de espera, la selección por defecto, establecer una contraseña... se edita con privilegios de root(admin).
- En el directorio /etc/grub.d/ se encuentra un conjunto de script ejecutables numerados. Su lectura en orden de numeración permite:
 - o 00_header: Carga de información básica del grub.
 - o 05_debian_theme: Establece la configuración de la imagen de fondo, color, texto...
 - o 10_linux: se encarga de localizar el kernel de Linux.
 - o 20_memtest86+: Localiza y añade la opción de testeo de memoria.
 - o 30_os-prober: Busca otros S.O. en el disco y los añade al menú.
 - o 40_custom: Archivo donde el usuario puede agregar entradas nuevas al menú del GRUB 2.