Funciones de los sistemas operativos

1. Introducción

Un sistema operativo es el conjunto de programas que administran los recursos del sistema y sirve de interfaz con el usuario. El fin fundamental de un sistema operativo es coordinar la utilización que se hace del hardware dependiendo de los programas que se estén utilizando.

En general, hardware, software y usuario se estructuran en cuanto a la utilización del ordenador de forma jerárquica

Por tanto, una de las tareas fundamentales del sistema operativo será la gestión de los diferentes recursos del hardware: el **procesador**, la **memoria** y los **dispositivos de entrada/salida**

2. Gestión del procesador (CPU)

Un **proceso** se define como un **programa en ejecución**. Los procesos en un sistema operativo tienen las siguientes características:

- Para poder iniciar su ejecución un proceso ha de **residir completamente en memoria** y **tener asignados los recursos** que necesite.
- Cada proceso está **protegido** del resto de procesos; ningún otro proceso podrá escribir en las zonas de memoria pertenecientes a ese proceso.
- Los procesos pueden **pertenecer al usuario** o ser **propios del sistema operativo**. Los procesos que pertenecen al usuario se ejecutan en el llamado **modo usuario** y tienen **restricciones de acceso a los recursos** del hardware. Los procesos que pertenecen al sistema se ejecutan en **modo kernel o modo privilegiado** y **no tienen restricciones** de acceso.

- Cada proceso tendrá una **estructura de datos** llamada **bloque de control de proceso (BCP)**, donde se almacenará la información del mismo.
- Los procesos podrán **comunicarse, sincronizarse y colaborar** entre sin

Un **programa no es un proceso**, se convierte en proceso en el momento en que se pone en ejecución. Un programa no deja de ser un archivo más, guardado en dispositivos de almacenamiento secundario. Al ejecutarlo las instrucciones necesarias pasan a la memoria principal, pasa a estar en ejecución y por tanto, se convierte en un proceso.

Durante la ejecución de un proceso este debe de competir con el resto de procesos en ejecución por el uso de recursos hardware y software del sistema <- multiprogramación.

Los sistemas operativos disponen de los servicios necesarios para la gestión de los procesos. Estos servicios se encargan de tareas como su creación, finalización, ejecución periódica, cambio de prioridad, cambio de estado de los procesos, etc.

2.1. Bloque de control de procesos o BCP

Como hemos visto el bloque de control de procesos es la estructura de datos que utiliza el sistema operativo para almacenar la información necesaria para la gestión de un proceso.

En sistemas operativos multiproceso el sistema mantiene listas de bloques de control de procesos para cada en las que cada elemento de la misma es el BCP de uno de los procesos que hay en ejecución.

En el BCP se almacena información como:

- **Identificador de proceso** (PID). Número que permite identificar al proceso. La forma de asignarlos depende del sistema operativo. En Linux, cuando termina el proceso de arranque se ejecuta el proceso init con PID 1, que se encarga de iniciar el resto de procesos del sistema operativo.
- **Estado del proceso**: ejecución, preparado o bloqueado.
- **Prioridad del proceso**. Permite establecer la cantidad de tiempo que un proceso puede utilizar el procesador. En los sistemas operativos basados en Linux a este parámetro se le llama nice. Los valores que puede tener de nice un proceso están entre -20 (mayor prioridad) y 19 (menor prioridad). Por defecto los procesos se crean con prioridad 0.
- **Ubicación en memoria**: Dirección en memoria en la que se carga el proceso.
- **Recursos utilizados**: recursos hardware y software que requiere el proceso para utilizarse.

2.2. Estados de un proceso

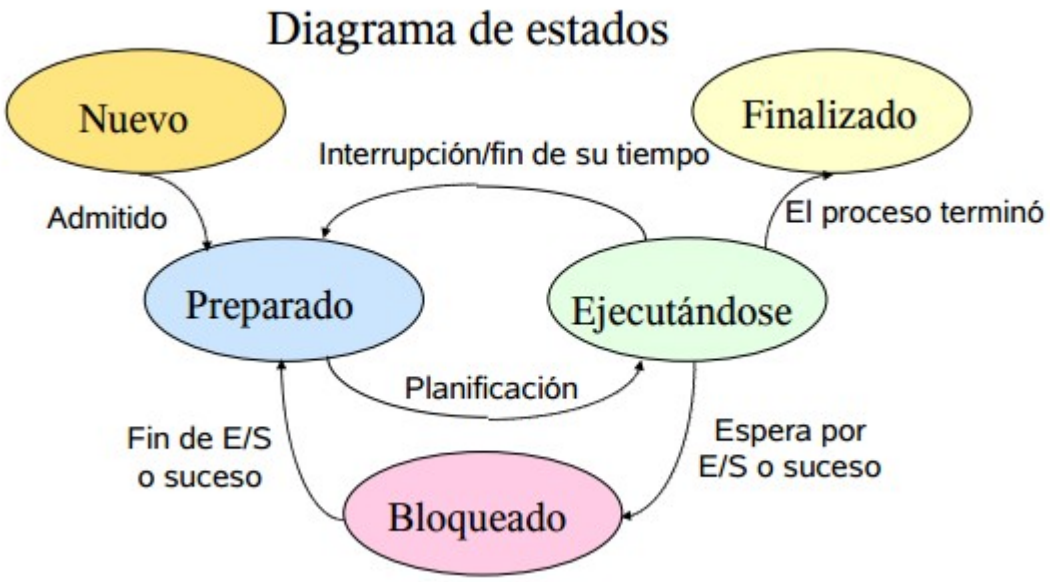
Una vez que un programa se ha ejecutado y se ha convertido en proceso, puede atravesar varias fases o estados hasta que finaliza su ejecución.

A medida que se ejecuta un proceso, cambia su estado. El estado de un proceso se define en parte por la actividad actual de dicho proceso. Cada proceso puede estar en alguno de los siguientes estados:

Nuevo: El proceso se está creando.
Ejecución: Se están ejecutando instrucciones.
Bloqueado o en espera: El proceso está esperando a que ocurra algún evento (Como la terminación de una operación de entrada o salida, o la recepción de una señal.
Preparado: El proceso está en espera de ser asignado a un procesador.
Terminado: El proceso ha terminado su ejecución.

A los cambios de estado por los que pasa un proceso se les llama transiciones.

El diagrama de estados que representa los posibles estados de un proceso y sus transiciones es de la forma:



2.3. Tiempos de los proceso

Para definir los estados por los que pasa un proceso se utilizan los siguientes tiempos:

- **Te** o tiempo de llegada: tiempo que tarda un proceso en llegar a memoria
- **Tx** o tiempo de ejecución: tiempo que tarda un proceso en ejecutarse.
- **Tr** o tiempo de respuesta: tiempo que transcurre desde que un proceso llega a memoria hasta que **finaliza** su ejecución
- **Ts** o tiempo de espera: tiempo que pasa desde que un proceso **llega a memoria** hasta que se **inicia su ejecución**
- **Q** o quantum: tiempo que el procesador asigna el uso del procesador a cada proceso. Utilizado en algoritmos de planificación en los que se establecen turnos

2.4. Algoritmos de administración de procesos

En todos los sistemas operativos existe un proceso llamado **planificador** o scheduler que se encarga de gestionar **cómo se reparte** la CPU entre los diferentes procesos que están en ejecución.

A la hora de gestionar dicho reparto de uso existen diferentes algoritmos que pueden utilizar los planificadores:

2.4.1. FIFO (First In, First Out). Primero en llegar primero en salir.

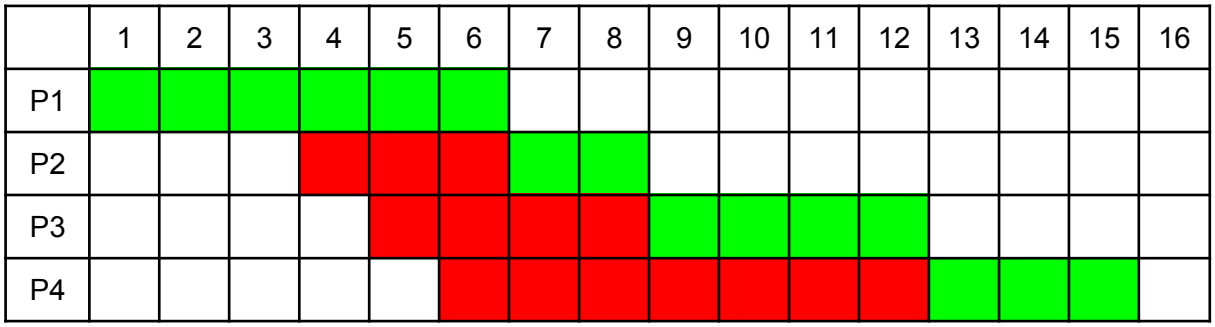
Cuando el planificador utiliza este algoritmo el procesador **ejecuta cada proceso hasta que termina**. Por tanto, los procesos que estén en cola permanecerán en la misma hasta que terminen de ejecutarse los procesos que llegaron antes a la cola.

Ejemplo:

Si a un sistema operativo monoprocesador que utiliza el algoritmo de planificación FIFO llega la siguiente secuencia de procesos:

Proceso	Te (tiempo de llegada)	Tx (tiempo de ejecución)
P1	0	6
P2	3	2
P3	4	4
P4	5	3

La secuencia de ejecución de los procesos en el procesador sería:



La tabla en la que especificaremos los tiempos de llegada y de ejecución medios y de cada proceso quedaría:

Proceso	Ts (tiempo de espera)	Tr (tiempo de respuesta)
P1	0	6
P2	3	5
P3	4	8
P4	7	10

Media	3,5	7,25

2.4.2. SJF (Shortest Job First). Primero se atiende al más corto

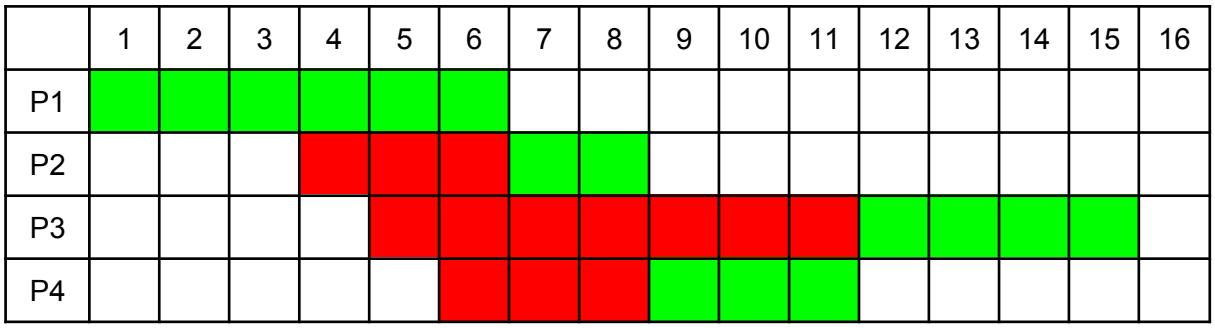
Cuando el planificador utiliza este algoritmo los procesos se ordenan en cola de acuerdo a su tiempo de ejecución, poniendo al principio los que tardan menos en ejecutarse. En resumen, este algoritmo selecciona al proceso con el próximo tiempo ejecución más corto.

Ejemplo:

Si a un sistema operativo monoprocesador que utiliza el algoritmo de planificación SJF llega la siguiente secuencia de procesos:

Proceso	Te (tiempo de llegada)	Tx (tiempo de ejecución)
P1	0	6
P2	3	2
P3	4	4
P4	5	3

La secuencia de ejecución de los procesos en el procesador sería:



La tabla en la que especificaremos los tiempos de llegada y de ejecución medios y de cada proceso quedaría:

Proceso	Ts (tiempo de espera)	Tr (tiempo de respuesta)
P1	0	6
P2	3	5
P3	7	11
P4	3	6
Media	3,25	7

2.4.3. SRTF (Shortest Remaining Time First). Primero se atiende al que menos tiempo le quede

Es similar a SJF, con la diferencia de que si un nuevo proceso pasa a listo se activa el planificador para ver si es más corto que lo que le queda por ejecutar al proceso en ejecución. Si es así el proceso en ejecución pasa a en espera. Y el proceso en que acaba de llegar pasa a en ejecución.

Ejemplo:

Si a un sistema operativo monoprocesador que utiliza el algoritmo de planificación SRJF llega la siguiente secuencia de procesos:

Proceso	Te (tiempo de llegada)	Tx (tiempo de ejecución)
P1	0	6
P2	3	2
P3	4	4
P4	5	3

La secuencia de ejecución de los procesos en el procesador sería:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
P1																
P2																
P3																
P4																

La tabla en la que especificaremos los tiempos de llegada y de ejecución medios y de cada proceso quedaría:

Proceso	Ts (tiempo de espera)	Tr (tiempo de respuesta)
P1	2	8
P2	0	2
P3	7	11
P4	3	6
Media	3	6,75

2.4.4. Round Robin. Turnos

Es un método para seleccionar todos los procesos de manera equitativa y en un orden racional, normalmente comenzando por el primer elemento de la lista hasta llegar al último y empezando de nuevo desde el primer elemento.

Todos los procesos son tratados con la misma prioridad.

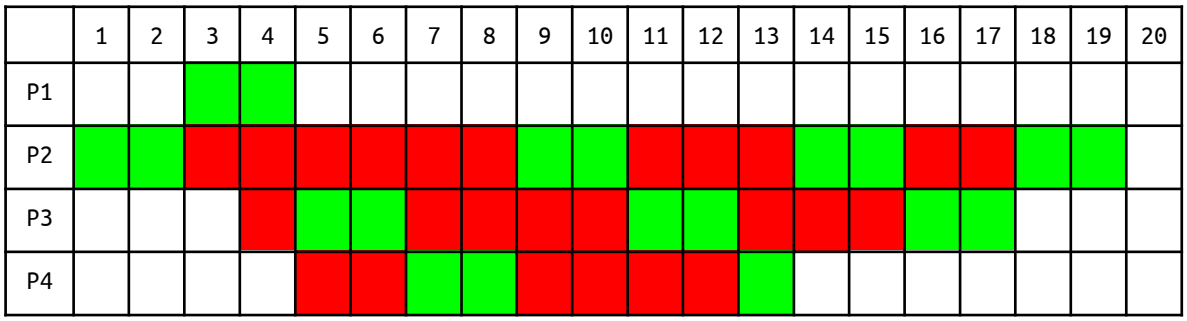
Se define un intervalo de tiempo denominado cuanto (quantum), cuya duración varía según el sistema. Ese tiempo será la duración del turno durante el cual cada proceso va a estar en ejecución.

Ejemplo:

Si a un sistema operativo monoprocesador que utiliza el algoritmo de planificación **Round Robin** con **q = 2** llega la siguiente secuencia de procesos:

Proceso	Te (tiempo de llegada)	Tx (tiempo de ejecución)
P1	2	2
P2	0	8
P3	3	6
P4	4	3

La secuencia de ejecución de los procesos en el procesador sería:



La tabla en la que especificaremos los tiempos de llegada y de ejecución medios y de cada proceso quedaría:

Proceso	Ts (tiempo de espera)	Tr (tiempo de respuesta)
P1	0	2
P2	11	19
P3	8	14

P4	6	9
Media	6,25	11

2.4.5. Por prioridad

Cada vez que termina un proceso el planificador mira la prioridad de los que están en cola. Pasará a ejecución el que tenga mayor prioridad de los que está en cola.

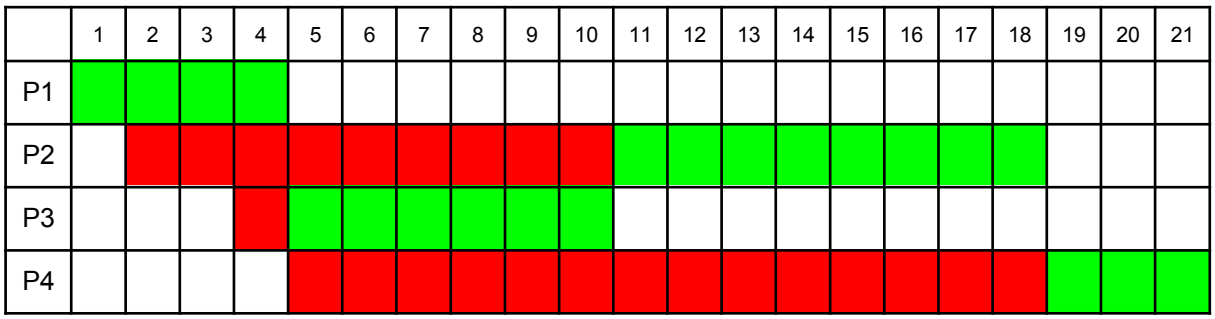
En los sistemas operativos se suele asignar **mayor prioridad** cuanto **menor** es el valor numérico de la misma y viceversa.

Ejemplo:

Si a un sistema operativo monoprocesador que utiliza el algoritmo de planificación

Proceso	Te (tiempo de llegada)	Tx (tiempo de ejecución)	Prioridad
P1	0	4	4
P2	1	8	2
P3	3	6	1
P4	4	3	3

La secuencia de ejecución de los procesos en el procesador sería:



La tabla en la que especificaremos los tiempos de llegada y de ejecución medios y de cada proceso quedaría:

Proceso	Ts (tiempo de espera)	Tr (tiempo de respuesta)
P1	0	4
P2	9	17
P3	1	7
P4	14	17

Media	6	11,25