SISTEMAS INFORMÁTICOS UD-2

INTRODUCCIÓN A LOS SISTEMAS OPERATIVOS Y LAS MÁQUINAS VIRTUALES (II)

GESTIÓN DE PROCESOS

ÍNDICE

[GESTIÓN DE PROCESOS 3](#_Toc120109766)

[A. DEFINICIÓN DE PROCESO 3](#_Toc120109767)

[B. PCB 3](#_Toc120109768)

[C. ESTADOS DE UN PROCESO 3](#_Toc120109769)

[D. BLOQUEOS 4](#_Toc120109770)

[1. BLOQUEO MUTUO 5](#_Toc120109771)

[E. INTERRUPCIONES 5](#_Toc120109772)

[F. MÓDULOS DE GESTIÓN DE PROCESOS 5](#_Toc120109773)

[G. CAMBIO DE CONTEXTO 5](#_Toc120109774)

[H. PLANIFICACIÓN DE PROCESOS 6](#_Toc120109775)

[I. CLASIFICACIÓN ALGORITMOS DE PLANIFICACIÓN 6](#_Toc120109776)

[J. ALGORITMOS DE PLANIFICACIÓN 7](#_Toc120109777)

[1. FIFO (First In, First Out) 7](#_Toc120109778)

[2. SJF (Shor Job First) 7](#_Toc120109779)

[3. SRT (Shortest Remaining Time) 8](#_Toc120109780)

[4. Round Robin 8](#_Toc120109781)

[5. Prioridades 9](#_Toc120109782)

# GESTIÓN DE PROCESOS

## DEFINICIÓN DE PROCESO

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

Cuando abrimos (invocamos) un programa, el SO crea una imagen de este en la memoria RAM. Y, para controlar este proceso y darle todo lo necesario para ejecutarse, crea un **bloque de control o PCB**.

Este PCB se crea por cada proceso.

## PCB

Almacena toda la información de cada proceso. Son los metadatos del proceso:

* Tabla

  Descripción generada automáticamentePID del proceso. Número de identificación del proceso.
* Estado. El estado de ejecución, preparado o bloqueado.
* Localización en la memoria RAM.
* Recursos usados para la ejecución.
* Propietario o lanzador del proceso. A quién le pertenece o puede hacer uso de él.
* Prioridad del proceso.

## ESTADOS DE UN PROCESO

Cada sistema utiliza sus propios estados para sus procesos:

* **Preparado**. El proceso está parado debido a que el sistema todavía no le puede dar el control de la CPU. El proceso está en cola de espera para tomar ese control.
* **Bloqueado**. El proceso se bloquea cuando no puede seguir siendo ejecutado hasta que se resuelve la causa de su bloqueo. El proceso está parado y no tiene el control de la CPU.
* **Ejecutando**. El proceso está siendo leído por el procesador y siendo ejecutado, es decir, tiene el control de la CPU.

Un proceso puede tener varias transiciones:

1. De preparado a ejecutando.
2. Diagrama, Esquemático

   Descripción generada automáticamenteDe ejecutando a bloqueado. Por ejemplo, intentar acceder al disco duro, mientras no consiga la información, el proceso no puede seguir ejecutándose.
3. De bloqueado a preparado.
4. De ejecución a preparado.

El ciclo de un proceso nuevo comienza siendo admitido y pasando a cola de preparados. A continuación, el planificador le da acceso a la CPU, esta ejecuta las primeras líneas de código y el proceso entra en ejecución. En este punto hay 3 posibilidades:

1. El proceso termina de ejecutarse.
2. El proceso para continuar su ejecución necesita información de la que no dispone, por ejemplo, información que se encuentra en el HDD y el proceso pasa a estar bloqueado hasta que se resuelva la situación. Cuando la situación sea resuelta pasa a preparado y deberá esperar a que el planificador le vuelva a dar el control de la CPU.
3. Diagrama

   Descripción generada automáticamenteEl proceso es interrumpido porque es necesario ejecutar otro proceso, en ese caso el programa entra en espera y vuelve a la cola de preparado. Una interrupción puede ser, por ejemplo, pulsar una tecla o mover el ratón.

## BLOQUEOS

Se producen cuando un proceso requiere una condición que el sistema no puede satisfacer. Por ejemplo, tenemos un proceso A que está usando un recurso R y entonces un proceso B necesita ese mismo recurso por lo que el sistema bloquea B hasta A libere dicho recurso.

1. BLOQUEO MUTUO

Se produce cuando dos procesos se bloquean esperando recursos ocupados por el otro. Ambos procesos quedarán bloqueados de forma indefinida.

Por ejemplo, un proceso A está usando un recurso R1 y se bloquea por el recurso R2. Un proceso B está usando un recurso R2 y se bloquea por el recurso R1. En este caso los procesos A y B no podrán continuar y se quedarán bloqueados.

En estos casos el SO actuará a través de la expropiación (quitando el recurso a uno de los dos) o de la detección preventiva (previendo que esto se va a dar y ejecutando una solución).

## INTERRUPCIONES

Son la forma de comunicarse entre un recurso hardware y la CPU. Es una línea eléctrica que se activa en la CPU.

Esta activación provoca que la CPU deje de ejecutar lo que está haciendo y salte a otra parte de la memoria.

La CPU comenzará a ejecutar la rutina de interrupción asociada a esa línea.

## MÓDULOS DE GESTIÓN DE PROCESOS

Existen dos módulos:

* **Planificador (*scheduler*).** Se encarga de decidir a qué proceso se le asigna la CPU, haciendo uso de algoritmos de planificación.
* **Distribuidor (*dispatcher*).** Se encarga de quitar el control de la CPU al proceso y pasarlo a la cola de preparados y dejar el control de la CPU a otro proceso. Esto es lo que se conoce como **cambio de contexto**.

## CAMBIO DE CONTEXTO

Se produce cuando el planificador cambia al proceso que está siendo ejecutado.

Pasos:

1. El planificador decide expulsar al proceso.
2. Almacena el estado del proceso en su bloque de control.
3. Calcula el siguiente proceso que debe entrar.
4. Restaura el estado del proceso entrante.
5. Sigue la ejecución.

En el estado del proceso se almacena una fotografía de todos los parámetros de éste, es decir, todos los datos del proceso en ese instante (los recursos que está ejecutando, la siguiente instrucción a ejecutar, etc.).

Destaca el valor del contador de programa que nos recordará en qué dirección dejamos el proceso.

Este es un proceso que no es gratuito, ralentiza el sistema porque lleva un tiempo.

## PLANIFICACIÓN DE PROCESOS

El planificador de procesos se va a encargar de asignar los recursos decidiendo qué proceso pasa de preparado a ejecución.

Aplica un algoritmo de planificación que es el método utilizado para seleccionar del proceso a ejecutar.

Se lleva a cabo mediante algoritmos con el objetivo de minimizar los siguientes parámetros:

* **Tiempo de retorno.** Es el tiempo que trascurre desde que sometemos un proceso al sistema hasta que ese proceso finaliza.
* **Tiempo de espera.** Tiempo en el que ha estado el proceso en la cola de procesos preparados.
* **Tiempo de respuesta.** Tiempo entre que el usuario hace una petición hasta que el sistema empieza a responder.

## CLASIFICACIÓN ALGORITMOS DE PLANIFICACIÓN

Existen diferentes tipos de algoritmos que harán que el SO actúe de forma más o menos agresiva, es decir, unos quitarán el control de la CPU al proceso y otros no.

* **No expulsora o no expropiativa.** Los procesos dejan la CPU de forma voluntaria. FCFS o SJF.
* **Expulsora o expropiativa.** Los procesos pueden perder la posesión de la CPU sin solicitarlo. Round-Robin, SRT, múltiples colas y de múltiples colas realimentadas.

## ALGORITMOS DE PLANIFICACIÓN

1. FIFO (First In, First Out)

Se sigue una cola normal, el primero que llega es atendido y se completa su proceso. Es decir, hasta que no termina un proceso no puede entrar el siguiente.

Tabla

Descripción generada automáticamente con confianza mediaTiene la ventaja de que es fácil de implementar, pero la desventaja de que el tiempo de espera promedio es muy largo.

1. SJF (Short Job First)

Se le asocia a cada proceso el tiempo de uso de la CPU. Cuando la CPU queda disponible, se asigna al proceso cuya ráfaga de CPU sea más corta.

Imagen que contiene Tabla

Descripción generada automáticamenteNo es expropiativo. Si un trabajo comienza a ejecutarse, termina de ejecutarse.

1. SRT (Shortest Remaining Time)

Modalidad expropiativa del anterior.

En este caso si llega a cola un proceso más corto que el que tiene el control de la CPU. El sistema quita el control a este programa y se lo da al programa que acaba de acceder a cola.

Imagen que contiene Tabla

Descripción generada automáticamente

1. Round Robin

Diseñado para sistemas de tiempo compartido. Se define una unidad de tiempo pequeña, llamada “quantum” de tiempo o porción de tiempo.

La cola de procesos listos se trata como una cola circular de tipo FIFO.

Imagen que contiene Calendario

Descripción generada automáticamenteEl planificador de la CPU recorre la cola de procesos listos, asignando la CPU a cada proceso durante un intervalo de tiempo de hasta un “quantum”.

1. Prioridades

Se asocia a cada cola una prioridad y la CPU se asigna al proceso que tiene la prioridad más alta.

A cada cola se le suele asignar su algoritmo propio.

En cada cambio de contexto de proceso, se asigna el procesador al proceso de la cola más prioritario que esté el primero en la lista de preparados.

En caso de que un proceso quede bloqueado, deja el procesador al siguiente en la lista de preparados (según prioridades).

Imagen que contiene Calendario

Descripción generada automáticamenteEl proceso bloqueado, una vez resuelto el bloqueo, vuelve a la cola de preparados de su prioridad en el último lugar.