

PROGRAMACIÓN DE SERVICIOS Y PROCESOS
**TÉCNICO EN DESARROLLO DE APLICACIONES
MULTIPLATAFORMA**

Procesos

ÍNDICE

| | |
|---|----|
| / 1. Introducción y contextualización práctica | 3 |
| / 2. Introducción a los procesos | 4 |
| / 3. Estados de un proceso | 4 |
| / 4. Caso práctico 1: “Un problema inesperado” | 5 |
| / 5. Gestión de procesos | 6 |
| / 6. Planificación de procesos I: Tipos | 7 |
| / 7. Planificación de procesos II: algoritmos de planificación | 7 |
| / 8. Caso práctico 2: “¿Qué planificador es mejor utilizar en el sistema operativo?” | 8 |
| / 9. Cambios de contexto | 9 |
| / 10. Resumen y resolución del caso práctico de la unidad | 10 |
| / 11. Bibliografía | 10 |

OBJETIVOS



Conocer el concepto de proceso.

Conocer e identificar todos los estados por los que puede pasar un proceso.

Conocer diferentes planificadores de procesos.

Conocer el concepto de cambio de proceso.



/ 1. Introducción y contextualización práctica

En esta unidad, trataremos en profundidad el concepto de proceso en un sistema operativo, qué es y cómo funciona.

Lo primero que veremos es el diagrama de estados de un proceso, que nos indicará todos los estados por los que va a pasar este durante su vida útil, así como las condiciones que deben darse para que se pase de un estado a otro.

También veremos todas las operaciones que podremos hacer con los procesos, es decir, cómo podremos gestionarlos.

Seguidamente, conoceremos cómo se planifican los procesos dentro del sistema operativo, y varios de los planificadores más usados por estos.

Por último, estudiaremos qué es un cambio de contexto y cómo afecta a los procesos cuando se lleva a cabo.

Todos estos conceptos los estudiaremos en los temas restantes, así que no te preocupes, que tenemos mucho tiempo por delante.

Escucha el siguiente audio, en el que planteamos el caso práctico que iremos resolviendo a lo largo de esta unidad:



Fig. 1. Ordenador.



Audio Intro. "Lanzando procesos"

<https://bit.ly/30AF48a>





/ 2. Introducción a los procesos

Ya sabemos que un proceso se puede definir como un programa en ejecución. Si queremos definir un proceso de una forma aún más completa, podemos decir que es una actividad caracterizada por la ejecución de una secuencia de instrucciones, que tiene un estado actual y un conjunto de recursos que toma del sistema.

Independientemente del sistema operativo que estemos utilizando (GNU/Linux, Windows o MacOS), todos son **multitarea**, lo cual nos va a permitir ejecutar varios procesos a la vez, pudiendo compartir un núcleo del microprocesador o varios. De esta forma, podremos ir ejecutando las tareas una tras otra o podremos realizar, poco a poco, cada tarea; las tendremos todas completadas al final, pero las podremos realizar de una forma más rápida.

Todo esto es gracias a la gestión de procesos que nos proporciona el propio sistema operativo.

Dentro de los procesos, podemos distinguir los siguientes tipos de acuerdo al **modo de ejecución**:

- **Procesos por lotes:** Este tipo de procesos están **formados por una serie de tareas que se van a realizar**. El usuario que las ejecuta únicamente está interesado en su resultado final, y no en su ejecución. Un ejemplo podría ser pasar el antivirus a nuestro ordenador.
- **Procesos interactivos:** En este tipo de procesos, habrá una **interacción del usuario y del propio proceso**, que puede pedir al usuario datos necesarios para su ejecución. Un ejemplo de podría ser un procesador de textos.
- **Procesos en tiempo real:** estos consisten en **ciertas tareas en las que el tiempo de respuesta por parte del sistema es crucial**. Un ejemplo podría ser un brazo mecánico en una operación o el sistema de conducción automática de un coche.

Además, atendiendo al origen de la ejecución, podemos distinguir también dos tipos de procesos:

- **Procesos en modo Kernel:** Son los procesos que ejecuta el propio núcleo del sistema operativo y que el usuario no podrá controlar. Realizan las tareas más delicadas, como la gestión de memoria, la gestión de tráfico de red, etc.
- **Procesos en modo usuario:** estos los lanza el propio usuario. Podemos decir que son los procesos más «normales» o usuales.



Vídeo 1. "Partes de un proceso"

<https://bit.ly/2CRNla5>



/ 3. Estados de un proceso

Es lo que conocemos como planificador. Se encarga de crear y poner en ejecución a los procesos en el sistema operativo, y el usuario intervenir de forma directa. Para que los procesos funcionen correctamente, existe el ciclo de vida de un proceso, que se compone de una serie de estados por los que pasará el proceso a lo largo de su ejecución, teniendo que cumplir ciertos requisitos para pasar de un estado a otro.

Los estados por los que puede pasar un proceso son:

- **Nuevo:** El proceso sea crea a partir de un fichero ejecutable, aunque aún no haya sido admitido en el grupo de procesos ejecutables por el sistema operativo.



- **Listo:** El proceso está creado, pero aún no está listo para ejecutarse, ya que el planificador del sistema operativo no lo ha seleccionado para entrar a ejecución.
- **En ejecución:** El planificador del sistema operativo, siguiendo un algoritmo de planificación concreto, ha elegido al proceso para que se ejecute. Este se ejecutará de forma continua hasta que se cumpla su tiempo máximo de ejecución o hasta que el planificador lo saque de este estado, pasando de 'nuevo' a 'Listo'. Si el proceso necesita algún recurso, lo podrá pedir mediante interrupciones.
- **Bloqueado / En espera:** El proceso ha pedido algún recurso mediante una interrupción y está esperando que ese recurso le sea concedido. Cuando se le conceda, se desbloqueará y volverá a pasar al estado 'Listo'.
- **Terminado:** El proceso ha finalizado su ejecución y libera todos los recursos que estaba acaparando, esperando que el sistema operativo lo destruya.

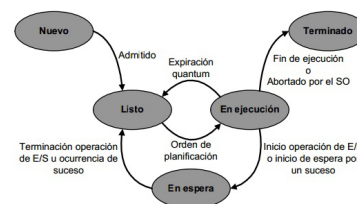


Fig. 2. Diagrama de estados de un proceso.



Vídeo 2. "Ejemplo de ciclo de vida de un proceso"

<https://bit.ly/3eV15DI>



/ 4. Caso práctico 1: "Un problema inesperado"

Planteamiento: Pilar y José están repasando el diagrama de estados de los procesos. «Es más complicado de lo que pensaba, la gestión de procesos no es algo que se pueda hacer de forma sencilla», le dice Pilar a José.

Sin embargo, José parece que ha entendido a la perfección cómo funciona este diagrama y cómo pasan unos procesos de un estado a otro, pero en su investigación le surge una duda que no es capaz de resolver. «Oye Pilar, ¿qué pasará si nos quedamos sin memoria RAM disponible?». Pilar se queda mirando a José sin saber qué responder.



Fig. 3. Gestionando qué procesos ejecutar.

Nudo: ¿Qué crees que pasará con la gestión de procesos cuando el ordenador se quede sin RAM disponible?

Desenlace: Seguro que casi todos tenemos unas cantidades enormes de memoria RAM en nuestros ordenadores: 8 GB, 16 GB, 32 GB..., pero hay ordenadores que no tienen esas cantidades de RAM, como puede pasar en un servidor de una pequeña empresa, la cual no tiene fondos suficientes para hacerse con un ordenador potente.

En esos casos, puede ocurrir que, en cierto momento, el sistema operativo se quede sin memoria RAM disponible, por lo que los procesos tendrán que salvarse de alguna forma, ya que, si no, se perdería todo.

La solución a esto nos la ofrece el SWAPING. El SWAPING viene dado por una pequeña partición en el disco duro que automáticamente el sistema operativo creará cuando es instalado y que nos servirá para estos casos.

De esta forma, cuando el ordenador se quede sin RAM disponible, este espacio de disco actuará como un fragmento más de RAM y se guardarán ahí los procesos que estén bloqueados, que podremos recuperar más adelante de una forma segura.

/ 5. Gestión de procesos

El encargado principal de toda la gestión de los procesos es el propio sistema operativo, que casi siempre sigue las órdenes del usuario. Así, cuando un usuario da la orden de abrir un programa, en realidad es el sistema operativo el responsable de crear y poner en ejecución el proceso que corresponde al programa en sí.

Cuando el procesador, mediante la orden del planificador del sistema operativo, pasa a **ejecutar un nuevo proceso**, es el sistema operativo quien debe guardar todo el contexto que tiene el proceso actual y quien debe restaurar el contexto que tenía el proceso que el planificador de procesos ha decidido ejecutar. El sistema operativo también es el responsable de controlar toda la **comunicación y sincronización** entre los procesos, además de la eliminación y terminación estos.

Ya sabemos que los sistemas operativos actuales nos permiten trabajar con multiprogramación, es decir, que admiten la ejecución de varios procesos en memoria para maximizar el uso del procesador. Todo esto es posible debido a que los procesos irán intercambiándose con los que están en ejecución, para que todos usen el procesador de igual forma, teniendo, así, una ejecución concurrente de los procesos.

Para esto, el sistema operativo organiza los procesos en varias colas:

- Una cola de procesos que van a contener todos los procesos que hay en el sistema.
- Una cola de procesos preparados, que contiene todos los procesos en estado 'Listo' que están esperando que el planificador los seleccione para ejecutarse.
- Varias colas que contienen los procesos que están en estado 'Bloqueado' y que están esperando alguna operación de entrada/salida. Habrá una cola de estas por cada dispositivo que permita una operación de entrada/salida.

Existen 4 eventos que van a provocar la creación de un proceso:

1. El arranque del sistema.
2. La ejecución, desde un proceso, de una llamada al sistema para la creación de otro proceso.
3. Una petición de usuario para crear un proceso.
4. El inicio de un fichero por lotes.

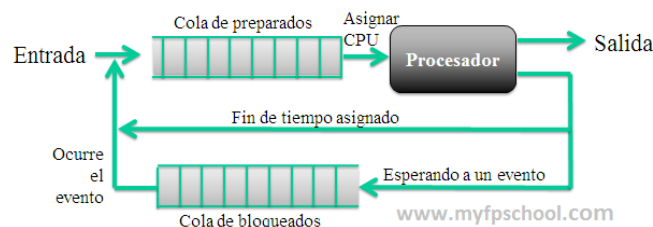


Fig. 4. Colas de procesos.



/ 6. Planificación de procesos I: Tipos

Para poder gestionar todas las colas de procesos que tenemos, necesitamos un planificador de procesos, que se encargará de seleccionar los procesos que van a ejecutarse en el procesador, imponiendo un orden de ejecución entre los procesos de las colas.

Existen, principalmente, dos tipos de planificación posibles:

- **Planificación de procesos a corto plazo:** Este tipo de planificadores seleccionan qué proceso de la cola de procesos preparados va a pasar a ejecución. Estos se invocan muy frecuentemente. Hablamos del orden de milisegundos, que es cuando se produce un cambio de estado del proceso en ejecución, por lo que deben ser muy rápidos en la toma de decisiones. Esto implica que los algoritmos de este tipo de planificadores sean muy sencillos, por ejemplo:
 - **Planificación sin desalojo o cooperativa:** En este tipo de planificación, solo se cambia el proceso en ejecución si se ha bloqueado o terminado.
 - **Planificación apropiativa:** Aparte de los casos cubiertos por la planificación cooperativa, este tipo de planificador cambiará el proceso en ejecución si en cualquier momento otro proceso con mayor prioridad puede ejecutarse.
 - **Tiempo compartido:** Este tipo de planificadores, cada cierto tiempo (que llamaremos quantum), quitan el proceso que estaba en ejecución y seleccionan otro proceso para que se ejecute. En este caso, todas las prioridades son consideradas iguales.
- **Planificación de procesos a largo plazo:** Este tipo de planificadores seleccionan los procesos nuevos que deben pasar a la cola de procesos preparados. Estos planificadores son invocados con poca frecuencia, ya que son muy lentos.

Los planificadores no se encargan de crear las estructuras que componen los procesos, únicamente se encargan de decidir, mediante un algoritmo, qué proceso se puede ejecutar en qué momento. Las estructuras de los procesos se crearán solo una vez, que es cuando el proceso se crea.

Los planificadores también son conocidos como scheduler.



Fig. 5. Decidiendo el proceso que se va a ejecutar.

/ 7. Planificación de procesos II: algoritmos de planificación

El hecho de planificar los procesos que se van a ejecutar es la práctica de un conjunto de políticas y mecanismos del propio sistema operativo, incorporadas en un módulo que ya conocemos, llamado **planificador de procesos**. Como ya hemos adelantado, **este se encarga de decidir qué procesos** (que estén listos para ser ejecutados), **deben iniciarse y en qué orden deben hacerlo**. El planificador de procesos debe realizar su trabajo asegurando el máximo aprovechamiento del sistema, es decir, debe dar servicio a todos los procesos que estén en ejecución en un momento dado y no dejar a ninguno de ellos fuera sin ejecutarse. Para ello el planificador puede utilizar distintos algoritmos de planificación:

- **Primero en llegar (FCFS o FIFO):** Este tipo de planificador **es uno de los más simples y hará que se vayan ejecutando los procesos según se creen**, sin ningún otro criterio.



- **Prioridad al más corto:** Este tipo de planificador ordenará los procesos según el tiempo de ejecución que necesiten e irá pasando a ejecución priorizando el que menos tiempo necesite. Con esta planificación, se produce lo que se conoce como inanición, ya que, si hay un proceso que necesita mucho tiempo de ejecución y siempre entran otros que necesiten menos, nunca se ejecutará.
- **Prioridad al más largo:** Este tipo de planificador ordenará los procesos según el tiempo de ejecución que necesiten e irá pasando a ejecución priorizando el que más tiempo necesite. Al igual que con la anterior, con esta planificación también se produce inanición, ya que, si hay un proceso que necesita muy poco tiempo de ejecución y siempre entran otros que necesiten más tiempo, nunca se ejecutará.
- **Round Robin:** Este tipo de planificación tiene un tiempo de ejecución llamado quantum, que es el tiempo que dejará ejecutarse a cada proceso, sacándolos de ejecución una vez pasado el quantum de tiempo. De este modo, todos los procesos se ejecutarán poco a poco.
- **Planificación por prioridad:** Este tipo de planificación utiliza la prioridad de ejecución que tienen los procesos, ejecutando antes el más prioritario. También puede producir inanición a procesos con poca prioridad.
- **Planificación de colas múltiples:** Este tipo de planificador es una mezcla de todos los anteriores.

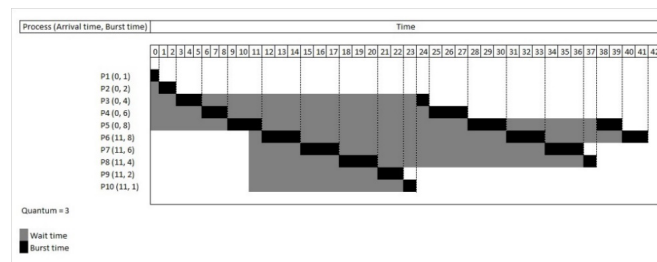


Fig. 6. Planificación de procesos con Round Robin.

/ 8. Caso práctico 2: “¿Qué planificador es mejor utilizar en el sistema operativo?”

Planteamiento: Una vez Pilar y José conocen qué son los planificadores de procesos del sistema operativo, tienen algo más claro cómo funciona la gestión de los procesos en el mismo.

Una de las cosas que más ha llamado la atención de nuestros amigos ha sido la cantidad de planificadores que existen. «Yo pensaba que para gestionar esto solo habría una forma y que todos los sistemas operativos lo hacían igual», le dice sorprendido José a Pilar, a lo que ella le responde que no tenía ni idea de que esto funcionase así.

Ahora, la duda que les asalta es cuál de estos planificadores será mejor para la gestión de procesos.

Nudo: ¿Qué piensas al respecto? ¿Crees que alguno de estos planificadores será mejor que los demás para la gestión de los procesos en el sistema operativo?

Desenlace: No nos cabe duda de que la gestión de procesos que lleva a cabo el sistema operativo es una tarea extremadamente complicada, ya que tiene que decidir cuál entra a ejecutarse en cada momento y cuál sale de



Fig. 7. Sistema operativo Windows.



ejecución para dejar entrar al nuevo. Ninguno de los planificadores que existen es totalmente perfecto, ya que, por ejemplo, muchos provocarán inanición, lo que hará que ciertos procesos no se ejecuten nunca.

Lo que hacen los diseñadores de los sistemas operativos es combinar los diferentes planificadores para intentar mejorar lo máximo posible esta gestión. Por este motivo, cada sistema operativo hace una gestión diferente de los procesos, ya que sus diseñadores intentan hacer la «mejor mezcla» para que la gestión de procesos sea lo más óptima posible.

/ 9. Cambios de contexto

Cuando el planificador de procesos decide sacar un proceso del estado 'Ejecutándose' y volver a ponerlo en el estado 'Listo' (lo cual ocurre muy frecuentemente), el sistema operativo debe ser capaz de guardar el contexto del proceso actual y restaurar el contexto del proceso que el planificador ha elegido ejecutar.

Esto significa que deben guardarse todos los datos de la situación del proceso que se estaba ejecutando. Este cambio de contexto se producirá también cuando hay una interrupción, es decir, cuando un proceso que se estaba ejecutando necesita un recurso y produce la interrupción para que se le proporcione.



Fig. 8. Guardando el estado de un proceso.

Cada vez que se produzca un cambio de contexto, el sistema operativo debe guardar:

- Estado en el que se encontraba el proceso.
- Estado del procesador, debiendo guardar todos los valores que tenían los diferentes registros del procesador en el momento del cambio de contexto.
- Información de gestión de memoria, debiendo guardarse el espacio de memoria que tenía reservado el proceso.
- Contador de programa, que es el encargado de indicar por dónde va ejecutándose el proceso, lo cual nos permitirá, más adelante, seguir ejecutando la instrucción del proceso desde donde se quedó.
- Puntero de pila, que se encarga de apuntar a la parte superior de la pila de ejecución del proceso.

Podemos decir que el cambio de contexto es «tiempo perdido» o tiempo no aprovechado, debido a que el procesador no hace un «trabajo útil» durante ese tiempo.

Una vez el proceso vuelva a entrar al estado 'Ejecutándose', todos los datos que han sido salvados se deberán volver a restaurar para que el proceso pueda seguir ejecutándose como si nada hubiese pasado.



Audio 1. "Recordatorio importante sobre el cambio de contexto"

<https://bit.ly/32ljxD>





/ 10. Resumen y resolución del caso práctico de la unidad

A lo largo de esta unidad, hemos visto qué son los procesos, cómo están compuestos y cómo los maneja el sistema operativo para ejecutar las aplicaciones o programas.

Hemos estudiado cómo está compuesto el diagrama de estados de un proceso, por el cual deberá pasar obligatoriamente en su ciclo de vida, viendo todos y cada uno de los posibles estados que se pueden dar y qué condiciones deben cumplirse para pasar de un estado a otro de forma correcta.

También hemos aprendido cómo podemos gestionar los procesos, es decir, qué operaciones podemos hacer con un proceso.

Además, hemos realizado un buen repaso de los planificadores de procesos que posee el sistema operativo: qué son y qué ejecuta cada uno de ellos.

Por último, hemos visto qué son los cambios de contexto y cómo afectan a los procesos cuando se dan en el sistema operativo.

Resolución del Caso práctico de la unidad

Ya sabemos que los procesos no son más que programas en ejecución, por lo que podemos deducir que lanzar un proceso significa lo mismo que ejecutar un programa.

Volviendo al problema que tienen que resolver nuestros amigos, en el entorno de trabajo virtual, deberán poder ejecutar varias aplicaciones o procesos que ya están disponibles. Como están desarrollando dicho entorno en Java, no vamos a tener problema para ello, ya que Java nos permite ejecutar procesos directamente desde código (como veremos en la siguiente unidad), por lo que podemos abrir cualquier programa o aplicación que tengamos instalada y disponible.

Gracias a esto, la gestión del entorno de trabajo virtual será mucho más sencilla que si tuviéramos que crear las funcionalidades desde cero.

/ 11. Bibliografía

Estado de un proceso y PCB. (21 de octubre de 2014). [Entrada de Blog]. Sistemas Operacionales. Recuperado de: <https://sistemasoperativosincca.wordpress.com/estado-de-un-proceso-y-pcb/>

Gómez, O. (2016). Programación de Servicios y Procesos Versión 2.1. Recuperado de: https://github.com/OscarMaestre/ServiciosProcesos/blob/master/_build/latex/ServiciosProcesos.pdf

Moreno, J. (28 noviembre de 2015). ¿Cómo se gestiona una CPU o Unidad de Control de Proceso (UCP)? MyFPschool. Recuperado de: <http://myfpschool.com/como-se-gestiona-una-cpu-o-unidad-de-control-de-proceso-ucp/>

Planificador, en Wikipedia, la enciclopedia libre. Recuperado de: <https://es.wikipedia.org/wiki/Planificador>

Planificador de procesos, en Wiki de Sistemas Operativos. (S. f.). Recuperado 8 de julio de 2020 de: https://1984.lsi.us.es/wiki-ssoo/index.php/Planificador_de_procesos

Proceso (informática), en Wikipedia, la enciclopedia libre. Recuperado de: [https://es.wikipedia.org/wiki/Proceso_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Proceso_(inform%C3%A1tica))

Round-robin scheduling, en Wikipedia, la enciclopedia libre. Recuperado de: https://en.wikipedia.org/wiki/Round-robin_scheduling

Sánchez, J. M. y Campos, A. S. (2014). Programación de servicios y procesos. Madrid: Alianza Editorial.