**¿V o F? Sobre Procesos- Planificación**

1. **La planificación a extra largo plazo no la realiza el SO y no afecta los tiempos de respuesta de los trabajos**

FALSO

Es cierto que no la realiza el SO pero afectará los tiempos de rta de los trabajos

El extra largo plazo (Fuera del S.O. en el ámbito organizacional y político), tiene una gran influencia sobre el funcionamiento del sistema.

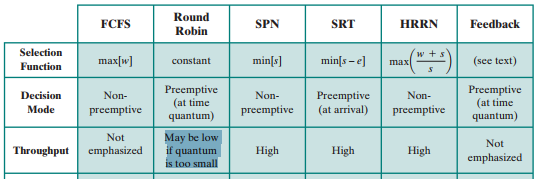
Dado que el sistema va a responder de acuerdo a esas políticas y como se planifica en el largo plazo

Consiste en una una planificación externa que se hace en el centro de cómputos y está estrechamente ligada a las políticas de funcionamiento del sistema. Es una planificación que se hace en concordancia con la política empresaria en que define el funcionamiento del centro de cómputos y se plantean las prioridades de usuarios o procesos, se organizan los programas que se van a ejecutar fuera de línea y ya encolados se los lleva a ejecución. Se realiza en el ámbito de usuarios y define las políticas de funcionamiento del sistema.

A través de procedimientos escritos se fijan las reglas que se aplicaran a los usuarios relativos al uso, seguridad, accesos, prioridades, etc.

1. **Una duración muy baja del quantum de tiempo en el algoritmo RR puede llegar a producir una baja performance en el Sistema**

Verdadero.



Stallings 8th 403

The performance of the RR algorithm depends heavily on the size of the time quantum. At one extreme, if the time quantum is extremely large, the RR policy is the same as the FCFS policy. In contrast, if the time quantum is extremely small, the RR approach can result in a large number of context switches.

Silberschatz 9th 272

1. **A pesar que el procesador sólo puede ejecutar una instrucción en cada instante de tiempo, gracias a la Multiprogramación, es posible que dos procesos se completen en un tiempo menor que si se ejecutaran de forma secuencial, incluso en un equipo con un solo procesador.**

Verdadero.

The operating system picks and begins to execute one of the jobs in memory. Eventually, the job may have to wait for some task, such as an I/O operation, to complete. In a non-multiprogrammed system, the CPU would sit idle. In a multiprogrammed system, the operating system simply switches to, and executes, another job. When that job needs to wait, the CPU switches to another job, and so on. Eventually, the first job finishes waiting and gets the CPU back. As long as at least one job needs to execute, the CPU is never idle.

Silberschatz 9th 19

1. **En un sistema, los procesos recién creados entran en una cola de espera. Entran en la cola de Ready sólo cuando se ve que la utilización de la CPU cae por debajo del 20%. Entonces se puede deducir que planificador a largo plazo utiliza este sistema.**

Falso.

En ningún lado hace referencia a un 20%

1. **En un sistema con un diagrama de 7 estados, un proceso podría pasar directamente del estado Ready al estado Exit**

Verdadero.

Ready -> Exit: For clarity, this transition is not shown on the state diagram. In some systems, a parent may terminate a child’ process at any time. Also, if a parent terminates, all child processes associated with that parent may be terminated.

Stallings 8th 119

1. **Solo cuando un proceso termina puede devolver un valor de estado a su padre, pidiendo que lo elimine**

Verdadero. "Un proceso termina cuando ejecuta su última instrucción y pide al sistema operativo que lo elimine usando la llamada al sistema exit(). En ese momento, el proceso puede devolver un valor de estado a su proceso padre (a través de la llamada al sistema wait() )". [Silberchatz-Galvin, 7ª ed. Esp., Pg. 85].

1. **Una falla en el mecanismo de sincronización de procesos concurrentes puede generar el fenómeno de inversión de prioridades**

Verdadero.

A simple example of priority inversion occurs if a lower-priority task has locked a resource (such as a device or a binary semaphore) and a higher-priority task attempts to lock that same resource. The higher-priority task will be put in a blocked state until the resource is available.

Stallings 8th 455

1. **La ejecución muy frecuente de los planificadores de corto y largo plazo es la principal característica de los mismos**

FALSO.

In terms of frequency of execution, the long-term scheduler executes relatively infrequently and makes the coarse-grained decision of whether or not to take on a new process and which one to take. The medium-term scheduler is executed somewhat more frequently to make a swapping decision. The short-term scheduler, also known as the dispatcher, executes most frequently and makes the fine-grained decision of which process to execute 1next.

Stallings 8th 400

1. **La llamada al sistema exec() o cualquiera que cumpla con la función de cambio de imagen no puede devolver un valor al proceso que la solicita**

Verdadero (Aunque Falso, hilando fino).

Because the call to exec() overlays the process’s address space with a new program, the call to exec() does not return control **unless an error occurs**.

Silberschatz 9th 117

1. **En un algoritmo de planificación de corto plazo sin desalojo (non-preemptive), no es posible que un proceso pase del estado RUNNING al estado READY.**

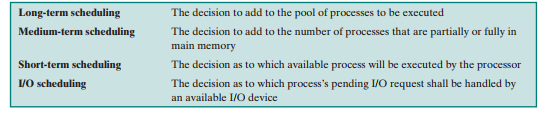
Falso.

A process may voluntarily release control of the processor. An example is a background process that performs some accounting or maintenance function periodically.

Stallings 8th 119

1. **El planificador a corto plazo (Short term scheduler) es quien toma también las decisiones de planificación de swapping de procesos.**

Falso.



Stallings 8th 396

1. **En el caso de los ULT (User Level Threads) el código para crear y destruir los threads son rutinas del Kernel**

Falso.

The threads library contains code for creating and destroying threads, for passing messages and data between threads, for scheduling thread execution, and for saving and restoring thread contexts.

Stallings 8th 165

1. **El planificador de corto plazo (Short Term Scheduler) es invocado solamente cuando se produce una interrupción de reloj.**

Falso.

The short-term scheduler is invoked whenever an event occurs that may lead to the blocking of the current process or that may provide an opportunity to preempt a currently running process in favor of another. Examples of such events include:

• Clock interrupts

• I/O interrupts

• Operating system calls

• Signals (e.g., semaphores)

Stallings 8th 400

1. **Una de las tareas del procesador al tratar una interrupción es asignar al contador de programa (Program Counter) la dirección de inicio del programa de manejo de interrupción (interrupt handler).**

Verdadero.

If an interrupt is pending, the processor does the following:

1. It sets the program counter to the starting address of an interrupt handler program.

2. It switches from user mode to kernel mode so that the interrupt processing code may include privileged instructions.

Stallings 8th 138

1. **La planificación por prioridades preemptive puede adolecer de problemas de starvation.**

Verdadero.

One problem with a pure priority scheduling scheme is that lower-priority processes may suffer starvation. This will happen if there is always a steady supply of higher-priority ready processes.

Stallings 8th 403

1. **Un procesador que ejecuta código de usuario, sólo podrá realizar un mode switch si se produce un system call.**

Falso (También en interrupciones o traps).

If an interrupt is pending, the processor does the following:

1. It sets the program counter to the starting address of an interrupt handler program.

2. It switches from user mode to kernel mode so that the interrupt processing code may include privileged instructions.

Stallings 8th 138

1. **El Busy Waiting se produce siempre que un proceso ejecuta un ciclo y tiene como desventaja que empeora la performance del sistema.**

Falso.

The term busy waiting, or spin waiting, refers to a technique in which a process can do nothing until it gets permission to enter its critical section but continues to execute an instruction or set of instructions that tests the appropriate variable to gain entrance.

Stallings 8th 212

1. **El PCB (Process Control Block) es mantenido en un almacenamiento secundario (generalmente disco) además de la localización en Memoria Central.**

Verdadero

El S.O. solo tiene que traer una parte de un proceso en particular. De ese modo que, en un momento dado, una parte del PCB del proceso puede estar en la en la memoria central y el resto en la memoria secundaria. Por lo tanto las tablas de procesos deben mostrar la ubicación de cada segmento o pagina del proceso tanto en memoria como en el disco.

Hay una tabla principal de procesos que administra el S.O. con una entrada para cada proceso.

(fuente: notas)

1. **La existencia de un planificador a corto plazo solo tiene sentido en los sistemas con multiprogramación**

VERDADERO

Sólo en este caso hay varios procesos en el sistema entre los que repartir la CPU

(fuente: neetzel)

Multiprogramacion: un modo de operación que permite la ejecución intercalada de dos o mas programas en un único procesador

*Planificador a corto plazo = dispatcher = short term schedule*

1. **El algoritmo de planificación de HRRN proporciona como efecto en los procesos, un buen balance entre procesos cortos y largos**

Verdadero.

While shorter jobs are favored (a smaller denominator yields a larger ratio), aging without service increases the ratio so that a longer process will eventually get past competing shorter jobs.

Stallings 8th 413

1. **La diferencia fundamental entre competencia y cooperación de procesos es que en la primera la sincronización es realizada solamente por los procesos, mientras que en la segunda necesariamente debe participar el SO**

FALSO

En competencia y cooperación por compartición: debe participar el SO en la sincronización.

En cooperación por comunicación: la sincronización es entre los procesos.

En competencia de procesos, los procesos no se conocen y compiten por recursos, debe haber un acceso controlado a los recursos compartidos ( no deberían ser

accedidos por más de un proceso al mismo tiempo), Para resolver estos problemas de competencia entre procesos, se utilizan dos mecanismos : la sincronización y la comunicación

En la cooperación de procesos por compartición: hay compartición de valores y aunque cada proceso no es explícitamente consciente de los demás procesos, es consciente de la necesidad de mantener la integridad de datos. Para mantener la integridad de datos se sincroniza el acceso a la sección critica

En la cooperación de procesos por comunicación: los procesos se conocen entre si, tienen primitivas de comunicación a su disposición. La comunicación es una manera de sincronizar las actividades.

**¿verificarlo?**

1. **Un proceso está formado por los siguientes 3 componentes: porción de un programa ejecutable, datos asociados que necesita el programa (variables, espacio de trabajo, buffers, etc) el contexto de ejecución del programa y de estos, el componente más esencial son los datos asociados que necesita el programa**

Falso.

We can think of a process as consisting of three components:

• An executable program

• The associated data needed by the program (variables, workspace, buffers, etc.)

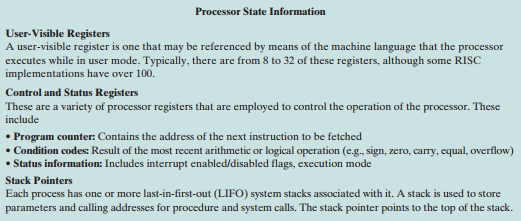
• The execution context of the program

This last element is essential. The execution context , or process state , is the internal data by which the OS is able to supervise and control the process.

Stallings 8th 64

1. **Aceptando que dentro del PCB encontramos información respecto a 3 categorías: identificación del proceso, información del estado del procesador e información de control del proceso, los registros de códigos de condición (signo, cero, acarreo, igual, etc) se encuentran dentro de la categoría información del estado del proceso**

Verdadero.



Stallings 8th 130

1. **Entre la categorías de implementación de hilos ULT y KLT, los KLT presentan respecto a los ULT la ventaja de que pueden ejecutar en cualquier SO**

Falso.

ULTs can run on any OS. No changes are required to the underlying kernel to support ULTs. The threads library is a set of application-level functions shared by all applications.

Stallings 8th 167

1. **El algoritmo de planificación SJF (Shortest Job First) se puede implementar para la cola de nuevos, pero no para la cola de listos.**

Verdadero (porque se necesita saber el largo).

Silberschatz 9th 268

1. **Un proceso se puede definir como una porción de un programa cargado en memoria central**

Falso.

Un proceso es una porción de programa en Memoria central más su PCB o su contexto de ejecución

Stallings 8th 109

1. **Siempre es necesario que una parte de la imagen de un proceso resida en memoria central**

Verdadero.

Siempre es necesario que una parte de la imagen de un proceso resida en memoria central para su ejecución

Stallings 8th 109

1. **El PCB contiene solo la información del identificador y de control de proceso**

Falso.

Stallings 8th 130

1. **El planificador a medio plazo decide qué procesos entran en el sistema**

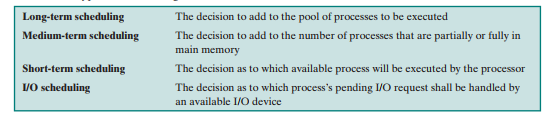
Falso.



Stallings 8th 396

1. **El planificador a largo plazo se encarga de decidir qué proceso tomará la CPU**

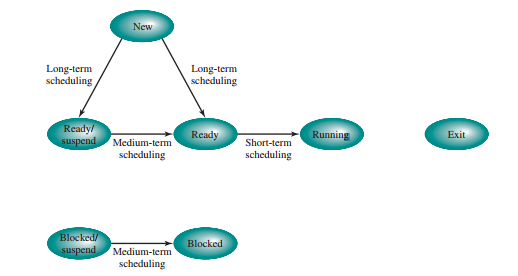
Falso.



Stallings 8th 396

1. **El planificador a medio plazo afecta a procesos creados**

Falso.

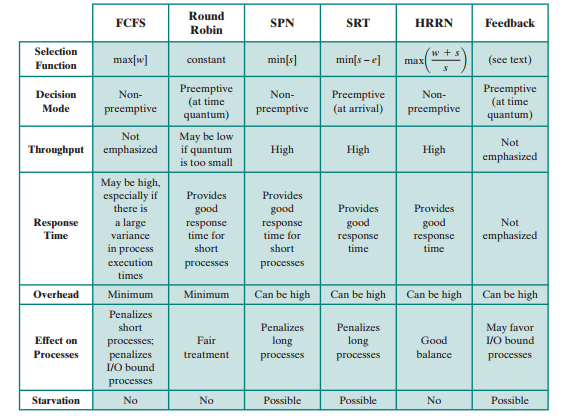


Stallings 8th 397

1. **Únicamente se produce starvation por el uso inadecuado de herramientas de**

**sincronización.**

Falso (en planificacion).



Stallings 8th 403

1. **Es imposible implementar Feedback con Round Robin en cada nivel en un sistema**

**operativo tipo Unix ya que se desconoce el largo de los procesos.**

Falso (no conocer el largo no afectaría a RR).

If we have no indication of the relative length of various processes, then none of SPN, SRT, and HRRN can be used.

Stallings 8th 413

1. **El algoritmo VRR, a diferencia del RR, hace más justa la distribución de tiempo de procesador para los procesos IO bound**

VERDADERO

Dado que los procesos IO Bound utilizan poco tiempo de procesador en cada ejecución con respecto a los procesos CPU Bound, se emplea un esquema que utiliza una cola de listos FCFS de mayor prioridad que la cola RR. A esta cola se dirigen los procesos una vez que luego de haberse bloqueado a la espera de un evento, este se produce. De esta forma se obtiene una mayor equidad, dado que los procesos en dicha cola, ejecutan por un periodo igual a un cuanto menos el tiempo que habían ejecutado en la cola RR

1. **El short term scheduler se ocupa entre otras cosas de balancear la carga de procesos I/O Bound y CPU Bound**

Falso.

It is important that the long-term scheduler select a good process mix of I/O-bound and CPU-bound processes.

Silberschatz 9th 113

1. **El cambio de proceso (process switch) puede ocurrir al tiempo que el s.o ha ganado el control desde el proceso en ejecución**

Verdadero.

A process switch may occur any time that the OS has gained control from the currently running process.

Stallings 8th 137

1. **Todos los elementos que forman parte de un proceso están guardados en el bloque de Control de Proceso**

Falso, la imagen del proceso no está guardada en el BCP, sino que está almacenada en segmentos de memoria aparte del Sistema Operativo

1. **El Bloque de control de procesos se almacena en la zona de datos del propio proceso**

Falso.

Es una estructura de datos del sistema operativo, que utiliza para gestionar los procesos, por tanto se almacena en la zona de datos del sistema operativo.

Stallings 8th 133

1. **El identificador de un proceso forma parte de los datos que se almacenan en la imagen de memoria del proceso**

Falso.

PCB

Stallings 8th 130

1. **La política SJF (primero el más corto) está pensada para su uso con procesos interactivos**

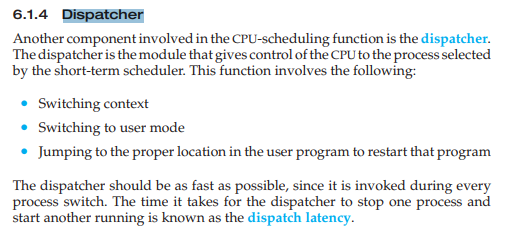
Falso, está pensada para sistemas con procesos cuya longitud es conocida de antemano

1. **El uso de prioridades dinámicas produce mayor sobrecarga del sistema operativo que el de prioridades estáticas**

Cierto, el sistema debe entrar a controlar cada cierto tiempo

1. **Una de las funciones que realiza el dispatcher es el de guardar el estado del proceso actual en su PCB y restaurar el estado del proceso siguiente a ejecutar.**

Verdadero (context switch).



Silberschatz 9th 265

1. **En un sistema que utiliza RR como política de planificación de procesos, si existen procesos que ejecutan solamente operaciones atómicas entonces el short term scheduler degenera en FCFS, ya que no se va a poder interrumpir el proceso en ejecución hasta que finalice.**

Falso.

Entre cada operacion atomica el short-term-scheduler va a cambiar de proceso si es que el quantum terminó.

1. **Un proceso suspendido siempre está a la espera de un evento.**

Falso.

• Blocked/Suspend: The process is in secondary memory and awaiting an event.

• Ready/Suspend: The process is in secondary memory but is available for execution as soon as it is loaded into main memory.

Stallings 8th 123

1. **El algoritmo de planificación SRT es el que mejor turn around time posee para cualquier proceso que se encuentre corriendo en el sistema.**

Falso, SRT beneficia a los procesos cortos y no a los largos por lo que puede haber inanición y el turn around de un proceso ser eterno.

1. **El so mantiene tablas para administrar los procesos**

VERDADERO

toda la información relativa en un proceso, distinta del contenido de su propio espacio de dirección se almacena en una tabla del Sistema Operativo llamada **tabla de procesos o PCB**, la cual consta de un arreglo (o lista enlazada) de estructuras, una por cada proceso existente en ese momento.

(fuente:notas)

1. **Cuando un proceso se bloquea es automaticamente removido a memoria secundaria, donde espera hasta que ocurra el evento por el que espera.**

**FALSO**

no siempre ocurre que se swapeen a memoria virtual los procesos, ya que si tengo suficiente memoria disponible para alocarlos, no es necesario que ello ocurra

**Existen cuatro tipo de planificaciones:**

1. Una planificación a extra largo plazo que consiste en planificaciones externas que se hacen en el centro de cómputo y están determinadas por las políticas del mismo.
2. Una planificación a largo plazo que se encarga de determinar y organizar los nuevos programas que van a ser admitidos al sistema.
3. Una a mediano plazo encargada del intercambio de los procesos, entre la memoria principal y la virtual.
4. Una planificación a corto plazo o “dispatcher”, que determina cuál es el próximo proceso que va a tomar la CPU.

De estas cuatro categorías podemos decir que las tres últimas se tratan de planificaciones internas al procesador, pero si hablamos de planificación interna tenemos que tener en cuenta también otra categoría, que sería la planificación de Entrada / Salida.

En lo que hace al desarrollo de ejercicios de planificación de procesos, interesa netamente la planificación a corto plazo, o sea, aquella de la que se encarga el dispatcher. El planificador a corto plazo se ejecuta cuando ocurre un suceso que puede conducir a la interrupción del proceso actual o que ofrece la oportunidad de expulsar al proceso actual a favor de otro respondiendo a políticas de uso del Procesador. La planificación de CPU consiste en decidir a cual de los procesos situados en la cola de preparados para la ejecución se le debe asignar CPU