de ejecución debe estar por terminar, debe favorecerse para ayudarlo a que termine y abandone el sistema tan pronto como sea posible

Cuanto tiempo más necesitará el proceso para terminar.- Los tiempos de espera promedio pueden reducirse al mínimo ejecutando primero aquellos procesos que requieran los menores tiempos de ejecución para terminar. Lamentablemente, casi nunca se sabe con exactitud cuanto tiempo durará un proceso.

Planificación no apropiativa y apropiativa

Una disciplina de planificación es *no apropiativa* si una vez que le ha sido asignado el CPU a un proceso, ya no se le puede arrebatar. En los sistemas no apropiativos, los trabajos largos retrasan a los cortos, pero el tratamiento para todos los procesos es más justo. Los tiempos de respuesta son más predecibles porque los trabajos nuevos de alta prioridad no pueden desplazar a los trabajos en espera.

Una disciplina de planificación es *apropiativa* si al proceso se le puede arrebatar el CPU. La planificación apropiativa es importante en aquellos sistemas de planificación en los cuales los procesos de alta prioridad requieren atención rápida. En los sistemas interactivos de tiempo compartido, la planificación apropiativa es importante para garantizar tiempos de respuesta aceptables.
La apropiación tiene un precio. El *cambio de contexto* implica un gasto extra. Para que la técnica de apropiación sea efectiva deben mantenerse varios procesos en el almacenamiento principal de manera que el siguiente proceso se encuentre listo cuando quede disponible el CPU. Conservar en el almacenamiento principal programas que no esten en ejecución también implica gasto extra.
Al hacer el diseño de un mecanismo de planificación apropiativa hay que tomar en cuenta la arbitrariedad de casi todos los sistemas de prioridades. Sin embargo es necesario analizar y evaluar cada mecanismo de planificación antes de ser implantado. La sencillez puede ser atractiva, pero si el mecanismo no se puede hacer sencillo, debe tratarse al menos de hacerlo efectivi y significativo.

Prioridades
Las prioridades pueden ser asignadas en forma automática por el sistema, o bien se pueden asignar externamente. Pueden ganarse o comprarse. Pueden ser estáticas o dinámicas. Pueden asignarse en forma racional, o de manera arbitraria en situaciones en las que un mecanismo del sistema necesita distinguir entre procesos pero no le importa cual de ellos es en verdad más importante.

Prioridades estáticas.- No cambian, son fáciles de llevar a la práctica e implican un gasto extra relativamente bajo. No responden a cambios en el ambiente que podrían hacer necesario un ajuste de prioridades. Es decir se mantienen constantes mientras dura el proceso.

Prioridades dinámicas.- Responden a los cambios. La prioridad inicial asignada a un proceso tiene una duración corta, después de lo cual se ajusta a un valor más apropiado. Este esquema es más complejo e implica más gasto extra que los esquemas estáticos, pero este queda justificado por el aumento de sensibilidad del sistema. Cambian en respuesta a los cambios de las condiciones del sistema.

Prioridades compradas.- Un sistema operativo debe proporcionar un servicio competente y razonable a una gran comunidad de usuarios, pero también debe manejar las situaciones en las cuales un miembro de la comunidad necesite un trato especial. Un usuario con un trabajo urgente puede estar dispuesto a pagar extra, esto es, *comprar prioridad*, por un nivel más alto de servicio. Este pago extra es obligatorio debido a que puede ser necesario arrebatar recursos con otros usuarios que también pagan. Si no hubiera un pago extra, entonces todos los usuarios pedirían un nivel más alto de servicio.

TIPOS DE PLANIFICACION

Planificación a Plazo Fijo.- Se programan ciertos trabajos para terminarse en un tiempo específico. Los trabajos pueden tener un gran valor si son entregados a tiempo y carecer de él si son entregados fuera del plazo, por lo que algún usuario puede estar dispuesto a pagar extra para asegurar que sus trabajos sean entregados a tiempo. Este tipo de planificación es compleja por varias razones que debemos considerar:

- El usuario debe informar por adelantado las necesidades precisas de recursos de la tarea. Esta información no suele estar disponible.
- El sistema debe ejecutar la tarea en un plazo determinado sin degradar el servicio a otros usuarios.
- El sistema debe planificar cuidadosamente sus necesidades de recursos dentro del plazo. Esto puede ser difícil por la llegada de nuevos procesos que impongan demandas impredecibles al sistema.
- Si hay muchas tareas a plazo fijo activas al mismo tiempo podría ser necesario la utilización de métodos de optimización para cumplir con los plazos.
- La administración intensiva de recursos requerida por ésta planificación puede ocasionar un gasto extra sustancial. Aunque los usuarios esten dispuestos a pagar una cuota alta por los servicios recibidos, el consumo neto de los recursos del sistema puede ser tan alto que el resto de la comunidad puede sufrir degradación del servicio.

Planificación Primeras Entradas Primeras Salidas (PEPS o First Input First Output FIFO).- Es la disciplina más simple. Los procesos se despachan de acuerdo a su tiempo de llegada a la cola de procesos listos. Es una disciplina no apropiativa. Es justa en el sentido formal pero es injusta con los procesos cortos que tienen que esperar a que los trabajos largos se ejecuten o bien que los trabajos importantes tienen que esperar a que se terminen los de menor importancia. Es la más predecible de las planificaciones. Sin embargo no es útil en la planificación para usuarios interactivos porque no puede garantizar buenos tiempos de respuesta.

Planificación por turno (Round Robin [RR]).- Es una disciplina <u>apropiativa</u>. Los procesos se despachan en forma PEPS, pero se les asigna una cantidad limitada de tiempo de CPU conocida como *cuanto o quantum*. Si un proceso no termina antes de que expire su tiempo de CPU, se le quitará el CPU y éste se le asignará al siguiente proceso en espera, el proceso desposeído se colocará al final de la cola de procesos listos. Es efectiva en ambientes de tiempo compartido en los que se necesita garantizar tiempos de respuesta razonables para usuarios interactivos.
El gasto extra debido a la apropiación es bajo gracias a eficientes mecanismos de cambio de contexto y a la asignación de suficiente memoria para que los procesos residan en la memoria al mismo tiempo.

Quantum.- La determinación del tamaño de quantum es vital para lograr una buena utilización del sistema y tiempos de respuesta razonable. Un tamaño de quantum muy grande hará que cualquier disciplina apropiativa se aproxime a su contraparte no apropiativa. Un quantum muy pequeño puede desperdiciar tiempo de CPU al obligar a un excesivo cambio de contexto entre procesos. Por lo que se debe de elegir lo bastante grande para que la mayoría de las solicitudes triviales terminen en un quantum. Ejemplo: en un sistema limitado de E/S, el quantum es lo bastante grande para que la mayor parte de los procesos puedan realizar una petición de E/S antes de que expire su quantum.

Planificación por Prioridad del Trabajo más Corto Primero (Shortest-job-first SJF).- Es una disciplina <u>no apropiativa</u> utilizada sobre todo para trabajos por lotes. Según esta disciplina se ejecuta primero el trabajo (o proceso) en espera que tiene el menor tiempo estimado de ejecución hasta terminar. Reduce al mínimo el tiempo promedio de espera pero los trabajos largos pueden verse sometidos a largas esperas.
El problema obvio con SJF es que exige conocer con exactitud el tiempo que tardará en ejecutarse un trabajo o proceso, y esa información no suele estar disponible; lo mejor que se puede hacer es basarse en los tiempos de ejecución estimados por el usuario.

Planificación del Tiempo Restante mas Corto (Shortest-remaining-time-scheduling SRT).- Es la contraparte <u>apropiativa</u> de SJF. En SRT, el proceso con el menor tiempo estimado de ejecución para terminar es el primero en ejecutarse, incluyendo los procesos nuevos. Un proceso en ejecución puede ser despojado por un proceso nuevo con un tiempo estimado de ejecución mas pequeño; implica un gasto extra mayor que SJF, pero proporciona un mejor servicio a los trabajos nuevos cortos. Reduce más los tiempos promedio de espera de todos los trabajos pero los trabajos largos pueden sufrir retrasos mucho mayores que en SJF.

Planificación por Prioridad de la Tasa de Respuesta mas Alta (Highest-response-ratio-next HRN).- Corrige algunos defectos de SJF, particularmente la excesiva predisposición contra los trabajos largos y el favoritismo de trabajos cortos nuevos. Es un disciplina <u>no apropiativa</u> en la cual la prioridad de cada trabajo no sólo es función del tiempo de servicio, sino también del tiempo que ha esperado el trabajo para ser atendido. Cuando un trabajo

obtiene el procesador, se ejecuta hasta terminar. Las prioridades dinámicas se calculan con la siguiente formula:

$$\text{Prioridad} = \frac{\text{Tiempo de espera} + \text{Tiempo de servicio}}{\text{Tiempo de servicio}}$$

Como el tiempo de servicio aparece en el denominador, los trabajos cortos tendrán preferencia. Como el tiempo de espera aparece en el numerador, los trabajos largos que han esperado también tendrán un trato favorable.
El tiempo de respuesta del sistema para el trabajo si este se inicia de inmediato se representa por esta suma:

$$\text{Tiempo de Respuesta} = \text{Tiempo de espera} + \text{Tiempo de servicio}$$