La planificación de la CPU también necesita un criterio para determinar a qué proceso se le asigna CPU. Para esto existen algoritmos de planificación.

Estos algoritmos pueden ser:

Apropiativos si el proceso en ejecución puede ser interrumpido y llevado a la cola de listos (se desperdicia CPU en los cambios de contexto).

No apropiativos si el proceso está en ejecución hasta que termine o se bloquee por algún evento.

Algunos de los algoritmos son: FIFO (primero en llegar, primero en ejecutarse), SJF (el proceso más chico primero), RR (los procesos se ejecutan por un quantum, que puede ser fijo o variable. cuando termina el quantum, se expulsa al proceso de la CPU y se coloca al final de la cola de listos), PRIORIDADES (los procesos tienen prioridades, se ejecutan en base a ellas), SRTF (versión apropiativa de SJF), VRR (igual que RR, pero cuando un proceso vuelve de E/S, tiene prioridad).

Los algoritmos pueden estar parametrizados para modificar la planificación.

Los procesos pueden ser de 2 tipos:

Batch (no hay usuarios esperando una respuesta, se pueden usar algoritmos no apropiativos)

Interactivos (tienen interacción con el usuario, conviene apropiativos).

También pueden ser independientes (si no afecta ni puede ser afectado por otros) o cooperativos (si afecta o es afectado por otros).

Los procesos siempre son creados por otro proceso. El proceso padre de todos nace ya creado ya que, al iniciar el SO, el loader lo carga. Cuando se crea un proceso, se crea la PCB y se le asigna un PID único. Hay 2 modelos para la relación padre-hijo: compiten por igual por CPU o el padre espera a que termine el hijo para ejecutarse. En Unix el espacio de direcciones del proceso hijo es un duplicado del proceso padre. Se crean procesos usando la SC fork() y se carga el programa en el espacio de direcciones con execve(). El fork copia el código para el hijo y dentro hay una condición para saber de quién es el código, si retorna 0, se ejecutó el hijo y si se retorna algo mayor a 0, el hijo se creó correctamente. Si retorna algo negativo, hubo un error.

Para finalizar los procesos se utiliza exit y el control del proceso se devuelve al SO.