

Gestionando un sistema operativo.

Los sistemas operativos de multiproceso permiten gestionar un conjunto de procesos simultáneos. La función del gestor del sistema operativo es decidir qué proceso se debe ejecutar antes. Para esto, los procesos (Pid) tienen asociada una prioridad dependiendo del número de niveles con el que trabaja el sistema operativo, es decir, si el sistema operativo tiene 5 niveles, cada proceso tendrá una prioridad del 0 al 5, siendo 0 la más alta y 5 la más baja.

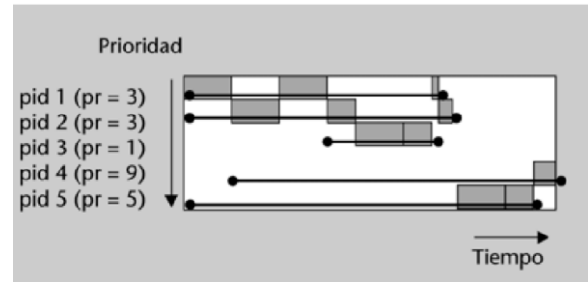
Las características principales de un sistema operativo son el número de niveles (como hemos explicado anteriormente), el número de ciclos que dedicará a cada actividad (número de veces que permite a un proceso ejecutarse) que será para todos los procesos la misma y, por último, el tiempo que tiene el sistema para gestionarse. Todas las tareas deben ejecutarse en ese periodo de tiempo.

Por otra parte, lo principal que debemos saber de los Pid es su prioridad y el tiempo que tarda en ejecutarse.

Cuando el gestor del sistema operativo debe elegir un proceso, seleccionará aquel que tenga la mayor prioridad y en caso de que dos procesos tengan la misma, se ejecutará primero el que más tiempo le quede por ejecutar y por último si los dos últimos datos son iguales, se dará prioridad al que menor Pid tenga.

Una vez seleccionado el Pid, el gestor del sistema operativo le restará el tiempo correspondiente (número de ciclos * tiempo de ciclo). Si el Pid en cuestión aún tiene tiempo por ejecutar, volverá a ser candidato en la siguiente ronda. Si por el contrario el Pid ha terminado su ejecución, el gestor del sistema operativo no lo volverá a tener en cuenta.

Recuerda que los procesos deben ejecutarse siempre dentro del tiempo total y que, si un proceso no completa el tiempo total del ciclo destinado a su desarrollo, solo restaremos el tiempo que hemos utilizado para ejecutar ese proceso.



Entrada:

La entrada estará compuesta por distintos casos de prueba, cada uno de ellos representando un sistema operativo al inicio de su ejecución.

Cada caso de prueba consta de varias líneas. La primera contiene el número N de niveles del sistema operativo y el tiempo T (en segundos) del que se quiere conocer el desarrollo de la ejecución, el número de procesos K que compiten por ser ejecutados y por último el número de ciclos C que va a emplear en cada proceso junto con el tiempo utilizado en cada ciclo (en segundos). A continuación, aparecen K líneas con la descripción de cada proceso: prioridad y tiempo de ejecución por ciclo.

Salida:

Para cada caso de prueba se escribirá una línea diciendo si se han podido ejecutar satisfactoriamente todos los procesos. En caso de que algunos de los procesos no hayan podido ejecutarse completamente, es decir, si no han agotado todo su tiempo inicial, se escribirá a continuación el número de procesos que sí se han completado.

Entrada de ejemplo:

```
10 200 3 2 25
PID1 3 100
PID2 2 50
PID3 3 200
5 200 2 2 15
PID1 4 20
PID2 0 80
20 1000 4 3 50
PID4 1 50
PID3 18 150
PID1 1 100
PID2 13 2000
15 300 3 2 20
PID2 12 50
PID3 12 50
PID1 12 100
```

Salida de ejemplo:

```
NO 1
SI
NO 2
SI
```