

Sistemas Operativos

Práctica de planificación de procesos.

Introducción

El objetivo de la práctica es comprender el funcionamiento de las distintas políticas de planificación del procesador en sistemas mono-procesador. Al finalizar se tiene que conocer las ventajas e inconvenientes de cada una de las políticas analizadas.

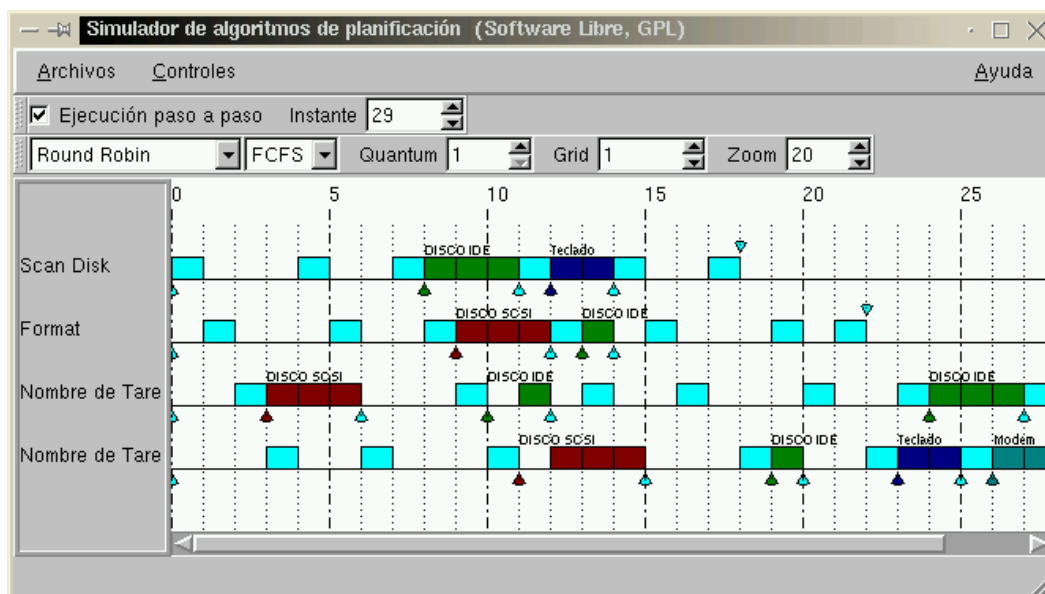
Se ha creado una aplicación gráfica que permite simular las distintas políticas de planificación. Esta aplicación carga un fichero de descripción del sistema, en el que se describen las características de las tareas que lo forman y luego, ya desde el entorno gráfico, se puede someter el conjunto de tareas a las distintas políticas de planificación.

El programa de simulación funciona sobre Linux, si bien están disponibles los programas fuentes y se puede compilar sobre otras muchas plataformas.

Las políticas de planificación que se va a utilizar durante la práctica, y que se encuentran implementadas en el simulador son: *First Come First Serve (FCFS)*, *Shortest Job First (SJF)*, *Shortest Remaining Time First (SRTF)*, *Round Robin*, *Prioridades*, *Prioridades Expulsivas* y *Prioridades expulsivas + Round Robin*. Se puede encontrar una breve descripción de cada una de estas políticas de planificación en el apéndice A.

Descripción del programa

Toda la actividad del programa se realiza desde la ventana principal:



Desde el menú de "Archivos" se puede cargar el fichero de descripción de tareas (más adelante veremos su formato), fichero con la extensión ".def". Éste es un fichero de texto plano en el que se describe el comportamiento de las tareas que componen el sistema. El fichero se ha de editar desde un editor de texto.

Se pueden tener varias ventanas principales abiertas, teniendo en cada una de ellas un algoritmo de planificación distinto. Para obtener una nueva ventana se puede utilizar **Ctrl-N** o desde el menú "Archivos->Nueva ventana".

El programa permite dos modos de visualización: **completa** o **paso a paso**, seleccionable desde un "botón" situado en una de las barras de herramientas. Con la ejecución paso a paso, el programa nos irá mostrando la situación del sistema hasta el **instante** que le indiquemos (localizado en la misma barra de herramientas). De esta forma se puede "jugar" a acertar cuál será la siguiente decisión que tomará el algoritmo de planificación.

El algoritmo de planificación empleado se elige en la primera lista desplegable de la segunda barra de herramientas. Todos los algoritmos que aquí aparecen son algoritmos de planificación a excepción del primero, etiquetado como "**Descripción**". Con la descripción se puede ver gráficamente cuál es el comportamiento de cada tarea por separado.

La política de gestión de recursos se elige desde la siguiente lista desplegable. Sólo se han implementado tres políticas: *FCFS*, *SJF* y prioridades fijas sin expulsión.

Si el algoritmo elegido es el *Round Robin* o el Prioridades con *Round Robin*, entonces se puede establecer el valor del *quantum* del sistema con el widget: Quantum .

Los dos últimos *widgets* de esta barra permiten ajustar los parámetros de visualización.

Debajo de las barras de herramientas, se encuentra el área de dibujo. En la parte izquierda aparecen los nombres de las tareas y a la derecha se encuentra el cronograma de cada tarea.

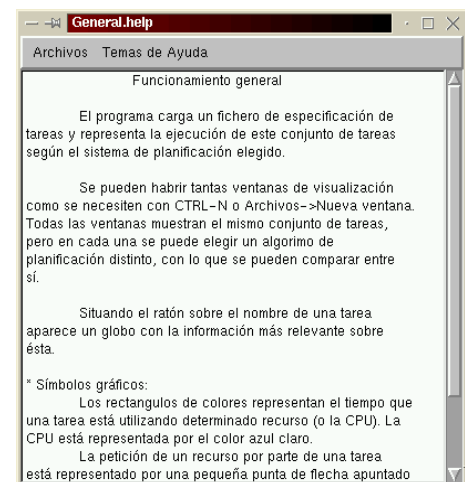
Ayuda

Con la tecla F1 (o desde el menú "Ayuda->Temas de Ayuda") aparece una ventana con ayudas tanto sobre el propio funcionamiento del programa como sobre los algoritmos de planificación disponibles. Se pueden abrir tantas ventanas de ayuda como se necesiten.

Situando el ratón sobre el nombre de las tareas, aparece (al cabo de un segundo) un pequeño globo de ayuda con la información más relevante sobre la tarea en cuestión. El globo de ayuda está visible durante 6 segundos.

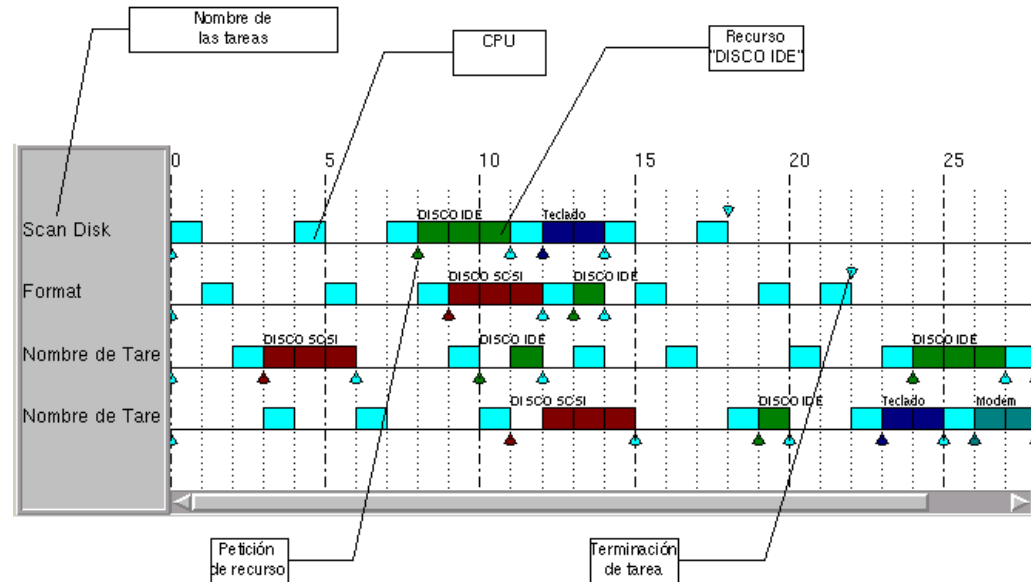
Símbolos del cronograma

Los rectángulos de colores representan el tiempo que una tarea está utilizando determinado recurso (o la CPU). La CPU está representada por el color azul claro. Cada recurso tiene un color distinto.



La petición de un recurso por parte de una tarea está representado por una pequeña punta de flecha apuntado hacia arriba debajo de los rectángulos. Y sobre el rectángulo aparece el nombre del recurso.

La terminación, o salida del recurso, está representado por una pequeña flecha apuntando hacia abajo encima de la tarea.



Trabajo de laboratorio

Ejercicio 1:

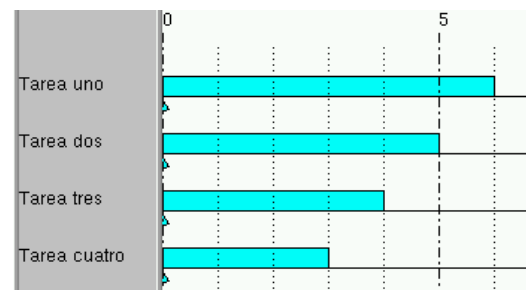
Dado el siguiente conjunto de tareas (fichero "solo_cpu.def"):

```
# Este conjunto de tareas no utiliza recursos
TAREA "Tarea uno" INICIO=0 PERIODO=0
PRIORIDAD=2 [CPU, 6]

TAREA "Tarea dos" PRIORIDAD=2 INICIO= 0
[CPU , 5 ]

TAREA "Tarea tres" PRIORIDAD=1 INICIO=0
[CPU , 4 ]

TAREA "Tarea cuatro" PRIORIDAD=1 INICIO=0
[CPU, 3]
```



1. Simúlalo paso a paso con cada una de las políticas de planificación intentando determinar con una unidad de tiempo de antelación cual será la siguiente tarea que se ejecute.
2. ¿Qué algoritmo de planificación ofrece un mejor tiempo de respuesta medio?
3. Calcula los tiempos de retorno de medio para cada algoritmo.
4. ¿Cual es el valor del *quantum* más pequeño que hace que el algoritmo *Round Robin* se comporte como el *FCFS* para este conjunto de tareas?

Ejercicio 2:

Para el conjunto de tareas del "*dos_tareas.def*":

```

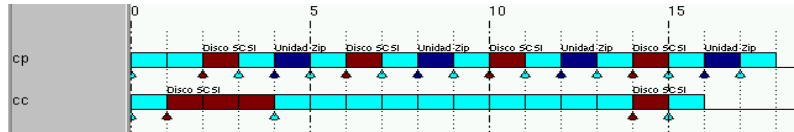
RECURSO "Disco SCSI"
RECURSO "Teclado"
RECURSO "Unidad Zip"

# Los recursos se tiene que declarar antes de utilizarlos
# Esta tarea copia un fichero del disco SCSI a Zip.

TAREA "cp" INICIO=0 PERIODO=30
[CPU, 2] [1, 1] [CPU, 1]
[3, 1] [CPU, 1] [1, 1] [CPU, 1]
[3, 1] [CPU, 1] [1, 1] [CPU, 1]
[3, 1] [CPU, 1]

# El compilador.... sobre
# todo necesita mucha CPU.
TAREA "cc"
[CPU, 1] [1, 3] [CPU, 10] [1, 1] [CPU, 1]

```



1. Calcula el tiempo de retorno del programa "cp" según cada política de planificación.
2. ¿Qué clase de problema se ve agudizado con las políticas no expulsivas?

Ejercicio 3:

Dado el siguiente conjunto de tareas "*ES_bound.def*":

```

RECURSO "R1"
RECURSO "R2"
RECURSO "R3"
RECURSO "R4"

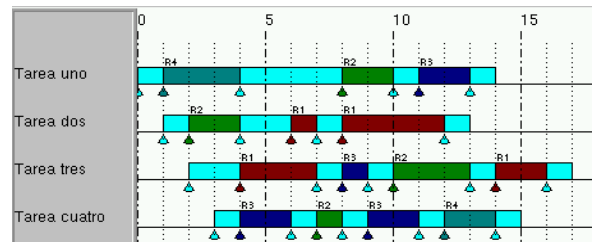
TAREA "Tarea uno" INICIO = 0 PERIODO = 0
PRIORIDAD=1
[CPU, 1] [4, 3] [CPU, 4] [2, 2] [CPU, 1] [3, 2] [CPU, 1]

TAREA "Tarea dos" PRIORIDAD=2 INICIO= 1
[CPU, 1] [2, 2] [CPU, 2] [1, 1] [CPU, 1] [1, 4] [CPU, 1]

TAREA "Tarea tres" PRIORIDAD=3 INICIO=2
[CPU, 2] [1, 3] [CPU, 1] [3, 1] [CPU, 1] [2, 3] [CPU, 1] [1, 2] [CPU, 1]

TAREA "Tarea cuatro" PRIORIDAD=4 INICIO=3
[CPU, 1] [3, 2] [CPU, 1] [2, 1] [CPU, 1] [3, 2] [CPU, 1] [4, 2] [CPU, 1]

```



1. Encuentra todos los instantes de tiempo, para cada una de las políticas de planificación de CPU, en los que todos los procesos están siendo atendidos por el sistema (en otras palabras, ningún proceso está en ninguna cola de espera).
2. En caso de no existir multiprogramación (ejecución secuencial de un proceso tras otro), cual sería el tiempo de respuesta de la "Tarea cuatro".
3. ¿Cual es la principal diferencia entre los resultados producidos por el *FCFS* y el *SJF*?

Ejercicio 4:

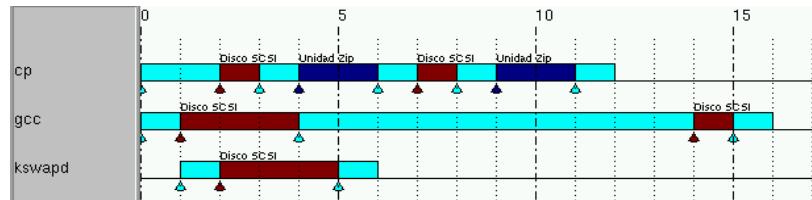
Dado el siguiente conjunto de tareas "*tareas_disco.def*":

```

RECURSO "Disco SCSI"
RECURSO "Teclado"
RECURSO "Unidad Zip"
#Esta tarea copia un
# fichero del disco SCSI a la
# unidad Zip.
TAREA "cp" INICIO=0 PRIORIDAD=1
TAREA "cp" INICIO=0 PRIORIDAD=1
[CPU, 2] [1, 1]          [CPU, 1] [3, 2] [CPU, 1] [1, 1] [CPU, 1] [3, 2] [CPU, 1]

# El compilador.... sobre todo necesita mucha CPU.
TAREA "gcc" INICIO=0 PRIORIDAD=1
[CPU, 1] [1, 3] [CPU, 10] [1, 1] [CPU, 1]

# El programa que mantiene la memoria virtual en disco
TAREA "kswapd" INICIO=1 PRIORIDAD=0
[CPU, 1] [1, 3]          [CPU, 1]
```



1. Compara para que políticas de CPU, de entre las disponibles, existen diferencias entre utilizar *FCFS*, *SJF* o *Prioridades* en la planificación de los recursos. Razona el por qué de la decisión tomada por el planificador de recursos en cada caso.
2. Calcula el tiempo total de espera en cola al acceder al *Disco SCSI* para cada una de las políticas de planificación de recursos, y variando la política de CPU entre las que muestran diferencias en la planificación de recursos.

Apéndice A

First Come, First Served (FCFS). Primero en Llegar, Primer en ser Servido

El planificador FCFS asigna la utilización de la CPU o el recurso a los procesos en el mismo orden de llegada.

Shortest Job First (SJF). Primero el trabajo más corto

El planificador SJF asigna el procesador al proceso que requiere menos tiempo de cómputo durante la siguiente ejecución. No se considera todo el tiempo de cómputo que el proceso va a necesitar para finalizar, sino el tiempo de procesador necesario hasta la próxima vez que se suspenda. No es una política de planificación expulsiva: aunque llegue una tarea más corta que la que está en ejecución, no la interrumpirá.

Shortest Remaining Time First (SRTF). Primero los trabajos más próximos a terminar

Muy parecido al SJF, pero en lugar de considerar el tiempo de cómputo, se comparan los tiempos que restan por ejecutar: tiempo de cómputo menos tiempo ya ejecutado. Es una política de planificación expulsiva.

Round Robin. Turno Rotatorio

A cada proceso se le asocia un "quantum" de tiempo. Si un proceso no ha abandonado la CPU (para hacer alguna operación de E/S) al finalizar su quantum, entonces el proceso es enviado al final de la cola de preparados y el procesador es asignado a otro proceso con un quantum completo.

Prioridades

A cada proceso se le asigna una prioridad (normalmente es el usuario el que lo hace). Cuando el procesador queda libre, se elige el proceso con mayor prioridad de entre los que están en la cola de procesos preparados.

Si llega a la cola de preparados un proceso con más prioridad que el proceso que está en el procesador, el nuevo proceso esperará en la cola de procesos activos hasta que el que está utilizando en el procesador lo abandone voluntariamente.

Prioridades Expulsivas

La política de planificación es idéntica a la anterior, sólo se diferencia en que si llega a la cola de preparados un proceso con más prioridad que el proceso que está en ejecución, el nuevo proceso pasa a ocupar la CPU y el proceso que estaba en ejecución pasa a la cola de preparados. Cuando existen varios procesos preparados con la máxima prioridad, se planifican en orden de llegada (FCFS).

Prioridades Expulsivas + Round Robin

Se comporta de forma similar a las *prioridades expulsivas*, pero cuando existen varios procesos con la máxima prioridad se utiliza la política *Round Robin* entre ellos.