AYUDANTÍA T1

Germán Leandro Contreras Sagredo
Ricardo Esteban Schilling Broussaingaray

IIC2333 [2018-2] - Sistemas Operativos y Redes

INTRODUCCIÓN

INTRODUCCIÓN

Los objetivos de esta ayudantía son:

- Entender el funcionamiento de la tarea 1.
- Resolver las dudas que surjan durante la explicación de lo pedido.

SCHEDULER

SCHEDULER

¿Qué es un scheduler?

- Encargado de decidir qué proceso poner a ejecución en la CPU.
- Puede ser una selección arbitraria (no recomendable), o basarse en criterios.

SCHEDULER

¿Qué criterios utilizar? Dependerá del tipo de scheduler:

- *Preemptive*: Hace uso de interrupciones para decidir cuándo cesar la ejecución de un proceso.
- *Non-Preemptive*: Esperan a que el proceso termine su ejecución (ya sea voluntariamente, por I/O o por finalización).

SCHEDULER

En particular, trabajaremos con un scheduler interactivo, i.e. que busca minimizar el tiempo de ejecución de un conjunto de procesos.

ROUND ROBIN

ROUND ROBIN

Implementaremos este scheduler, que es interactivo y *preemptive* (es decir, interrumpe a los procesos, no espera a que estos cedan la CPU).

ROUND ROBIN

Y, ¿cómo funciona?

1. Atiende a los procesos por orden de llegada (o creación) y los ordena en una cola.
2. Deja que cada proceso ejecute un máximo de tiempo antes de interrumpirlos para darle el paso al siguiente. A este tiempo le llamamos *quantum*.
3. Es justo, ya que a todos los procesos les otorga la oportunidad de ejecutar.

ROUND ROBIN

Además de las características antes mencionadas, en la simulación consideraremos lo siguiente con respecto a los procesos:

- Tienen periodos en los que hacen uso de la CPU (CPU-burst o ráfaga) y otros en los que esperan un ingreso por I/O (I/O-burst).
- En la secuencia de input, los tiempos A_i reflejan una ráfaga y los tiempos B_j reflejan una espera de por un input (que simularemos como simple espera).

ROUND ROBIN

Además de las características antes mencionadas, en la simulación consideraremos lo siguiente con respecto a los procesos:

- Además, los procesos poseen estados. Estos son: `READY`, `RUNNING`, `WAITING` y `FINISHED`.
- Los procesos ubicados en la cola se encuentran en estado `READY` y, al ingresar a la CPU para ejecutar, pasan a estar en estado `RUNNING`.
- Pasan a estado `WAITING` una vez que terminan una ráfaga y posteriormente viene un tiempo de espera. **No ingresan a la cola hasta haber terminado este intervalo de tiempo.**
- Pasan a estado `FINISHED` al terminar de ejecutar la última ráfaga.

ESTADÍSTICAS

ESTADÍSTICAS

Durante la simulación, además de indicar los cambios de estado de los procesos, **debe** obtener una estadística general para cada proceso.

Veremos en qué consiste cada uno de estos.

ESTADÍSTICAS

Los siguientes datos tienen relación con los accesos de los procesos a la CPU:

- Número de usos de la CPU: Corresponde a la cantidad de veces que el proceso ingresa a la CPU para ejecutar, es decir, la cantidad de veces que pasa a estado `RUNNING`.
- Número de bloqueos: O bien el número de interrupciones. Corresponde a la cantidad de veces que el scheduler decide sacar al proceso de la CPU por el consumo del quantum.

ESTADÍSTICAS

A continuación, se presentan distintas métricas de tiempo que también debe calcular a lo largo de la simulación:

- *Turnaround time*: Tiempo total que le toma al proceso terminar su ejecución (pasar a estado `FINISHED`) desde que ingresa a la cola por primera vez.
- *Response time*: Tiempo que le toma al proceso ser atendido por primera vez una vez que ingresa a la cola en estado `READY`.
- *Waiting time*: Tiempo total que el proceso estuvo en espera (ya sea en la cola esperando a ser atendido o en estado `WAITING`).

ESTADÍSTICAS

Un ejemplo de los tiempos antes mencionados:

```
[t = 12] El proceso GERMY ha sido creado.
[t = 15] El proceso GERMY ha pasado a estado RUNNING.
[t = 17] El proceso GERMY ha pasado a estado WAITING.
[t = 18] El proceso GERMY ha pasado a estado READY.
[t = 19] El proceso GERMY ha pasado a estado RUNNING.
[t = 20] El proceso GERMY ha pasado a estado FINISHED.
```

En base al ejemplo anterior, tenemos que:

- *Turnaround time*

$$T_{\text{Término}} - T_{\text{Llegada}} = 20 - 12 = 8$$

- *Response time*

$$T_{\text{Atendido}} - T_{\text{Llegada}} = 15 - 12 = 3$$

- *Waiting time*

$$\sum_{i}(T^{i}_{\texttt{RUNNING}} - T^{i}_{\texttt{READY}}) + \sum_{j}(T^{j}_{\texttt{READY}} - T^{j}_{\texttt{WAITING}})$$

$$= ((15 - 12) + (19 - 18)) + ((18 - 17)) = 5$$

CASOS ESPECIALES

CASOS ESPECIALES

Existen casos especiales en que no necesariamente hay una única forma de manejarlos. Para facilitar la corrección y estandarizar las simulaciones, se ha determinado qué considerar en cada uno de estos.

CASOS ESPECIALES

- Si el quantum se agota al mismo tiempo que la ráfaga actual de un proceso, cuenta como bloqueo. No obstante, el proceso pasa directamente a estado `WAITING` o `FINISHED`, no pasa de forma intermedia a estado `READY`.
- Su proceso puede llegar en tiempo $t = 0$. Debe asegurarse de que su programa no se caiga en este caso.
- Si existe un único proceso en el sistema, este funciona de la misma forma. Es decir, sigue siendo interrumpido por el scheduler y saliendo de la CPU si agota su quantum.

CASOS ESPECIALES

- Si un proceso nuevo llega a la cola al mismo tiempo que otro pasa de estado `WAITING` a `READY`, arbitrariamente ingresa primero el proceso nuevo y luego el que haya estado en espera.
- Si dos procesos pasan de estado `WAITING` a `READY` al mismo tiempo, vuelven a ingresar a la cola **en el mismo orden** que entraron a `WAITING`. Notar que no es posible que hayan ingresado a este estado al mismo tiempo.

CASOS ESPECIALES

Además de lo anterior, hay que considerar que cuando no existe ningún proceso en ejecución, se hace uso de uno especial llamado idle que “ejecuta” en la espera.

Es ideal considerarlo para entender dentro de la simulación lo que está pasando (es decir, imprimir sus cambios de estado). No obstante, **no se debe considerar dentro del archivo de las estadísticas.**

PRECAUCIONES, CONSEJOS, OTROS

PRECAUCIONES Y CONSEJOS

- Al pedir un tiempo de ejecución de su tarea menor a 30 segundos, no buscamos que sea muy eficiente con una gran cantidad de procesos. Con esta restricción simplemente esperamos que su implementación sea lo suficientemente razonable para no quedarse pegada en casos simples.
- Lean bien el enunciado y modelen el problema según lo pedido, será más fácil para ustedes después simular el comportamiento del scheduler.
- Usen punteros, ocupan menos espacio y es menos probable que incurran en errores.

TESTS

Se incluye un único test en conjunto con su resultado esperado, el que les servirá para ver que su simulación se comporta de la forma esperada.

Es importante que verifiquen que sus resultados coincidan, porque este será parte de los tests de evaluación.

DUDAS, CONSULTAS

¿Dudas, consultas?

FIN