

Laboratorio de Sistemas Operativos

Primer examen parcial

Fecha de entrega: Miércoles, 12 de noviembre de 2008

Planificación de procesos

En un sistema monoprocesador, los procesos aparentan ser ejecutados en forma paralela. En realidad una sola CPU debe ser compartida por los procesos, que ejecutan una cantidad de instrucciones durante un determinado intervalo de tiempo antes de ser desalojados para dar paso a otros procesos. Su problema consiste en simular la tarea de planificación de procesos en un sistema operativo. Para ello, deberá implementar las primitivas necesarias para permitir la planificación de procesos utilizando colas de prioridad.

Entrada del programa

Los datos del simulador se leerán por entrada estándar, y consistirán en una serie de comandos (DEFINE, PROCESS, START, CLEAN y EXIT), que indican la definición de un parámetro de simulación, los datos para un proceso, el inicio de la simulación, y el comienzo de otra simulación, respectivamente. Una vez que se lea el comando START (es decir que ha terminado la lectura de los datos), se deberá comenzar la simulación. A continuación se describe el uso de estos comandos.

Comando	Descripción	Ejemplo
DEFINE	Define un parámetro para la simulación. Los únicos parámetros que se pueden definir son el número de colas de prioridad a utilizar, la estrategia de planificación (FIFO o RR) para cada cola y el cuanto (necesario solamente si se utiliza Round-Robin) para cada cola.	#Utilizar tres colas de prioridad DEFINE queues 3 #Definir la estrategia para cada cola DEFINE scheduling 1 RR DEFINE scheduling 2 RR DEFINE scheduling 3 FIFO #Definir quantum (prioridad) para cada cola #cola DEFINE quantum 1 5 DEFINE quantum 2 4 DEFINE quantum 3 2
PROCESS	Anuncia los datos un proceso: su nombre, su prioridad (la cola en la cual debe ser ubicado) y su tiempo de ejecución.	#A: prioridad: 1 llegada: 0 tiempo: 10 PROCESS A 1 0 10 #B: prioridad: 2 llegada: 5 tiempo: 4 PROCESS B 2 5 4
START	Anuncia el fin de la lectura de los datos y el inicio de la simulación. No especifica ningún parámetro.	START
CLEAN	Anuncia que se debe inicializar el simulador para comenzar otra simulación.	CLEAN
EXIT	Indica que el simulador debe terminar.	EXIT

Se puede suponer que la entrada del programa es válida, es decir que los comandos y los parámetros se dan en el formato especificado y que su orden es correcto (por ejemplo, no se podrá establecer el quantum o la estrategia de planificación de una cola que no existe, ni se podrá anunciar la llegada de un proceso a una cola que no existe).

Tampoco se tendrá el comando para iniciar la simulación antes de definir todos los parámetros.

Suponga también que la planificación entre colas se realiza con una estrategia round-robin, y que la llegada de un proceso de mayor prioridad al que se está ejecutando no implica que se deba interrumpir el proceso que tiene la CPU. De esta forma, se garantiza que no existirá inanición para los procesos de menor prioridad. También recuerde que primero se deberá procesar la llegada de un nuevo proceso a la cola que le corresponde. Es decir, si en el mismo momento en que llega un nuevo proceso se retira un proceso de la CPU (y ambos tienen la misma prioridad), primero deberá entrar a la cola el nuevo proceso y luego se pondrá en la cola el proceso que salió de la CPU.

Salida del programa

Para cada simulación, del programa deberá imprimir la secuencia de ejecución de los procesos (en términos de los cuantos asignados), y el tiempo promedio de espera de los procesos. Opcionalmente se podrá imprimir otra información, como el número de colas de prioridad, la estrategia de planificación y el quantum para cada cola, etc.

Ejemplo de entrada

La siguiente es una secuencia de comandos de entrada. Para este caso se utiliza una sola cola de prioridad, que se planifica con Round Robin y un quantum de 5 unidades de tiempo.

```
#Definir los parametros de simulación.
#Las líneas que comienzan por '#' son comentarios y deben ser ignorados
#Esta simulación define una sola cola de prioridad con Round Robin.
# El simulador debe soportar varias colas de prioridad, con FIFO o RR.
DEFINE queues 1
DEFINE scheduling 1 RR

#Quantum para la cola 1: 5 unidades de tiempo
DEFINE quantum 1 5

#Las líneas en blanco también son ignoradas!

#Ahora definir los procesos...
#En este caso los dos llegan a la misma cola (solo hay una!): la cola 1.

#Primer proceso (p1), prioridad (cola) = 1, t llegada = 5, t. ejecución 10.
PROCESS p1 1 5 10

#Segundo proceso (p2), prioridad (cola) = 1, t llegada = 5, t. ejecución 8.
PROCESS p2 1 0 8

#Inicia la simulación...
START

#limpiar para comenzar otra simulación...
CLEAN

DEFINE .....
DEFINE ....

PROCESS ...
....
START
EXIT
```

Ejemplo de salida

Para la primera simulación que se muestra en el ejemplo de entrada, la salida del programa deberá ser:

SIMULACIÓN #: 1

(opcional) Número de colas de prioridad: 1

(opcional) Planificación de las colas:

(opcional) 1: Round Robin, $q=5$

(opcional) Número de procesos: 2

Secuencia de ejecución:

p2 (5), p1(5), p1(3 OK), p2 (5 OK)

Tiempo promedio de espera: 4

El formato de la salida es libre, sin embargo para cada simulación deberá contener como mínimo la secuencia de ejecución y el tiempo promedio de espera.