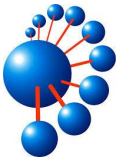

Simulador de planificación de procesos

Sistemas Operativos (2022-2)

Integrantes: Marco Antonio Aguayo Solís
Dazhi Enrique Feng Zong
Pablo Ignacio Zapata Schifferli
Profesora: Cecilia Hernández R.
Fecha: 27 de noviembre, 2022

Descripción del simulador

El simulador consiste en M hebras de kernel que van ejecutando N procesos cada ronda. Cada proceso que es creado empieza con prioridad 5 y un tiempo de ejecución entre a y b. Cada ronda imprime en la consola el estado de cada proceso en la runqueue activa, junto a su tiempo de ejecución. Al ejecutar un proceso, se imprime en la consola la CPU encargada, si es terminado se notifica. Si no alcanzó a terminarse, se actualiza su tiempo de ejecución y prioridad, luego es agregado a la runqueue expirada.

Se implementaron dos runqueues, cada una es un vector de 10 colas de enteros. Ambas van alternándose el trabajo de activa y expirada. Para acceder a cada cola, ya sea para agregar o sacar un proceso, se requiere obtener un lock asociado a ella.

Las hebras tipo E y G fueron implementadas como funciones y cada una tiene asociada una hebra de kernel. Cada hebra E va ejecutando los procesos de la runqueue activa por orden de prioridad. Cuando una hebra E termina de procesar la runqueue activa, hace un wait() en una variable de condición vcE. Si todas las hebras E están esperando, se hace un signal a la variable de condición vcG para activar la hebra G. La hebra G intercambia los roles entre la runqueue activa y expirada y añade más procesos, luego hace un broadcast a vcE para que las hebras E sigan ejecutando procesos.

En el main:

1. Se pide como entrada N, M, a y b. Luego se crea todo (runqueues, locks, etc.)
2. Se crean N procesos que se agregan a la runqueue activa. Además de añadir el tiempo y luego imprimir la información (cola, tiempo, proceso).
3. Se crea el vector de hebras T y se le pasa de parámetro la función de la hebra E y la CPU.
4. Adicionalmente, se crea la hebra G (con su función).

Pseudocódigos

- **int defPrioridad(int prioridadActual, int nuevoTiempo)**
  Si nuevoTiempo/quantum > 3: la nueva prioridad disminuye en uno, en otro caso aumenta en uno. La prioridad es un valor entre 0 y 9.

```
cvE, cvG ← variables de condición
me, mg ← mutex
```

- **hebra_E(int h):**

```
cola ← 0
while 1:
    bloquear locks[activa][cola]
    if runqueue[activa][cola] no está vacía:
        sacar un proceso de la cola ejecutarlo por el mínimo entre el quantum y el tiempo que necesita
        desbloquear locks[activa][cola]
        sleep()

        si el proceso no completo su ejecución:
            tiempo[proceso] ← random entre [a, b]
            nuevaPrioridad ← defPrioridad(cola, tiempo)
            bloquear locks[expirada][nuevaPrioridad]
            agregarlo a la cola expirada correspondiente
            desbloquear locks[expirada][nuevaPrioridad]

    `   else:
        desbloquear locks[activa][cola]
        si está en la última cola:
            si todas las hebras E están esperando:
                cvG.signal()

            cvE.wait(me)
        cola ← (cola +1)%10
```

- **hebra_G():**

```
while 1:
    cvG.wait(mg)
    me.lock()
    intercambiar las colas expirada y activa
    agregar procesos nuevos a la cola activa, misma cantidad de los que finalizaron en la ronda anterior
    me.unlock()
    cvE.broadcast()
```

Supuestos y limitaciones

El quantum es igual a uno, por lo que valores altos de a y b harán que los procesos siempre disminuyan su prioridad, valores bajos de a y b provocarán el efecto contrario. Con a = 1, b = 5, el simulador funciona bien.

Resultados

Con N = 20, M = 3, a = 1, b = 5. El simulador pareciera hacer bien su trabajo

```
Cola
Activa Proceso(Tiempo)
 0     -
 1     -
 2     -
 3     -
 4     -
 5     0(1) 1(3) 2(5) 3(2) 4(1) 5(4) 6(4) 7(4) 8(2) 9(4) 10(1) 11(1) 12(2) 1
3(3) 14(4) 15(2) 16(2) 17(4) 18(1) 19(1)
 6     -
 7     -
 8     -
 9     -

CPU 0, Proceso 0 ejecutandose
CPU 1, Proceso 1 ejecutandose
CPU 2, Proceso 2 ejecutandose
CPU 1, Proceso 3 ejecutandose
CPU 0, Proceso 0 finalizado
CPU 0, Proceso 5 ejecutandose
CPU 2, Proceso 4 ejecutandose
CPU 0, Proceso 6 ejecutandose
CPU 1, Proceso 7 ejecutandose
CPU 2, Proceso 4 finalizado
CPU 2, Proceso 8 ejecutandose
CPU 2, Proceso 9 ejecutandose
CPU 1, Proceso 10 ejecutandose
CPU 0, Proceso 11 ejecutandose
CPU 1, Proceso 10 finalizado
CPU 1, Proceso 13 ejecutandose
CPU 0, Proceso 11 finalizado
CPU 0, Proceso 14 ejecutandose
CPU 2, Proceso 12 ejecutandose
CPU 1, Proceso 15 ejecutandose
CPU 2, Proceso 16 ejecutandose
CPU 0, Proceso 17 ejecutandose
CPU 0, Proceso 18 ejecutandose
CPU 1, Proceso 19 ejecutandose
CPU 1, Proceso 19 finalizado
CPU 0, Proceso 18 finalizado
```

```
Cola
Activa Proceso(Tiempo)
 0     -
 1     -
 2     -
 3     -
 4     1(1) 3(2) 8(2) 7(3) 6(2) 9(2) 13(2) 12(2) 14(2) 17(2) 15(2) 16(2)
 5     10(1) 11(1) 12(2) 13(2) 14(2) 15(2)
 6     2(4) 5(5)
 7     -
 8     -
 9     -

CPU 0, Proceso 1 ejecutandose
CPU 2, Proceso 3 ejecutandose
CPU 1, Proceso 8 ejecutandose
CPU 2, Proceso 7 ejecutandose
CPU 1, Proceso 6 ejecutandose
CPU 0, Proceso 1 finalizado
CPU 0, Proceso 9 ejecutandose
CPU 1, Proceso 13 ejecutandose
CPU 0, Proceso 12 ejecutandose
CPU 2, Proceso 14 ejecutandose
CPU 1, Proceso 17 ejecutandose
CPU 2, Proceso 15 ejecutandose
CPU 0, Proceso 16 ejecutandose
CPU 0, Proceso 10 ejecutandose
CPU 1, Proceso 12 ejecutandose
CPU 2, Proceso 11 ejecutandose
CPU 0, Proceso 10 finalizado
CPU 0, Proceso 14 ejecutandose
CPU 1, Proceso 13 ejecutandose
CPU 2, Proceso 11 finalizado
CPU 2, Proceso 15 ejecutandose
CPU 2, Proceso 15 finalizado
CPU 1, Proceso 2 ejecutandose
CPU 0, Proceso 5 ejecutandose
```

```
Cola
Activa Proceso(Tiempo)
 0     -
 1     -
 2     -
 3     8(1) 6(1) 16(3) 15(1) 17(2)
 4     12(3) 14(2)
 5     3(4) 9(4) 7(4) 13(5) 14(2) 12(3) 5(3) 2(1) 16(3) 17(2) 18(1) 19(1)
 6     13(5)
 7     -
 8     -
 9     -

CPU 0, Proceso 8 ejecutandose
CPU 2, Proceso 6 ejecutandose
CPU 1, Proceso 16 ejecutandose
CPU 0, Proceso 8 finalizado
CPU 0, Proceso 17 ejecutandose
CPU 2, Proceso 6 finalizado
CPU 2, Proceso 12 ejecutandose
CPU 1, Proceso 15 ejecutandose
CPU 2, Proceso 14 ejecutandose
CPU 0, Proceso 3 ejecutandose
CPU 1, Proceso 15 finalizado
CPU 1, Proceso 9 ejecutandose
CPU 0, Proceso 7 ejecutandose
CPU 2, Proceso 13 ejecutandose
CPU 1, Proceso 14 ejecutandose
CPU 1, Proceso 12 ejecutandose
CPU 2, Proceso 5 ejecutandose
CPU 0, Proceso 2 ejecutandose
CPU 2, Proceso 16 ejecutandose
CPU 0, Proceso 2 finalizado
CPU 0, Proceso 18 ejecutandose
CPU 1, Proceso 17 ejecutandose
CPU 0, Proceso 18 finalizado
CPU 2, Proceso 19 ejecutandose
CPU 1, Proceso 13 ejecutandose
CPU 1, Proceso 13 finalizado
CPU 2, Proceso 19 finalizado
```