PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

SISTEMAS OPERATIVOS

2da práctica (tipo a)
(Primer semestre de 2014)

Horario 0781: prof. V. Khlebnikov

Duración: 1 h. 50 min.
Nota: No se puede usar ningún material de consulta.
La presentación, la ortografía y la gramática influirán en la calificación.
Puntaje total: 20 puntos

Pregunta 1 (2 puntos – 10 min.) (*MOS4E, Chapter 2, Problem 44*) Five jobs are waiting to be run. Their expected run times are 9, 6, 3, 5 and *X*. In what order should they be run to minimize average response time? (Your answer will depend on *X*.)

Pregunta 2 (3 puntos – 15 min.) (*MOS4E, Chapter 2, Problem 45*) Five batch jobs. *A* through *E*, arrive at a computer center at almost the same time. They have estimated running times of 10, 6, 2, 4, and 8 minutes. Their (externally determined) priorities are 3, 5, 2, 1, and 4, respectively, with 5 being the highest priority. For each of the following scheduling algorithms, determine the mean process turnaround time. Ignore process switching overhead.

(a) Priority scheduling.
(b) First-come, first served (run in order 10, 6, 2, 4, 8).
(c) Shortest job first.

Assume that only one job at a time runs, until it finishes. All jobs are completely CPU bound.

Pregunta 3 (2 puntos – 10 min.) (*MOS4E, Chapter 2, Problem 46*) The CTSS (the M.I.T. Compatible TimeSharing System) designers realized that it was more efficient to give CPU-bound processes a large quantum once in a while, rather that giving them small quanta frequently (to reduce swapping). On the other hand, giving all processes a large quantum would mean poor response time. Their solution was to set up priority classes. Processes in the highest class were run for one quantum. Processes in the next-highest class were run for two quanta. Processes in the next one were run for four quanta, etc. Whenever a process used up all the quanta allocated to it, it was moved down one class.

A process running on CTSS needs 30 quanta to complete. How many times must it be swapped in, including the very first time (before it has run at all)?

Pregunta 4 (2 puntos – 10 min.) (*MOS4E, Chapter 2, Problem 47*) Consider a real-time system with two voice calls of periodicity 5 msec each with CPU time per call of 1 msec, and one video stream of periodicity 33 ms with CPU time of 11 msec. Is this system schedulable?

Pregunta 5 (2 puntos – 10 min.) (*MOS4E, Chapter 2, Problem 49*) The aging algorithm with $a = 1/2$ is being used to predict run times. The previous four runs, from oldest to most recent, are 40, 20, 40, and 15 msec. What is the prediction of the next time? (Consider that the first prediction, after the first run, is equal to the run.)

Pregunta 6 (2 puntos – 10 min.) (*MOS4E, Chapter 2, Problem 51*) In the dining philosophers problem, let the following protocol be used: An even-numbered philosopher always picks up his left fork before picking up his right fork; an odd-numbered philosopher always picks up his right fork before picking up his left fork. Will this protocol guarantee deadlock-free operation?

Pregunta 7 (7 puntos – 35 min.) (*The Little Book of Semaphores, Section 4.3.1*) Solution to the mutual exclusion problem using weak semaphores (Morris's solution):

```
 1      room1 = room2 = 0
 2      mutex = Semaphore(1)
 3      t1 = Semaphore(1)
 4      t2 = Semaphore(0)

 1      mutex.wait()
 2          room1 += 1
 3      mutex.signal()

 5      t1.wait()
 6          room2 += 1
 7          mutex.wait()
 8          room1 -= 1

10          if room1 == 0 :
11              mutex.signal()
12              t2.signal()
13          else :
14              mutex.signal()
15              t1.signal()

17      t2.wait()
18          room2 -= 1

20          # critical section

22          if room2 == 0 :
23              t1.signal()
24          else :
25              t2.signal()
```

a) (2 puntos – 10 min.) Explique cómo trabaja el código si existe un solo hilo de ejecución.

b) (4 puntos – 20 min.) Explique cómo trabaja el código si aparecen otros hilos cuando el primer hilo llega a la línea 8.

c) (1 punto – 5 min.) ¿Cómo se demuestra que no hay inanición?

La práctica ha sido preparada por VK en LibreOffice Writer.

Profesor del curso: (0781) V. Khlebnikov

Pando, 6 de mayo de 2014