

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

INF239 SISTEMAS OPERATIVOS
Semestre 2024-2
Laboratorio 5

Usted debe entregar sus respuestas en un archivo con nombre *codigo.odt* (LibreOffice Writer), por ejemplo si su código es 20201916, el archivo debe ser *20201916.odt*

1) (10 puntos) Evaluación de algoritmos

Se tiene los siguientes procesos que llegan todos al mismo tiempo

Proceso	Tiempo de servicio
A	20
B	13
C	19
D	21
E	9

Tabla 1

a) (2 puntos) Haciendo uso del simulador, determine qué algoritmo minimiza el tiempo medio de espera. Justifique su respuesta presentando *screenshots* de las simulaciones. En el caso de *round robin* elija el *quantum* más apropiado.

b) (2 puntos) Considerando su respuesta de la parte a), ésta se puede generalizar o es una respuesta para el caso particular presentado en la Tabla 1. Justifique su respuesta presentando *screenshots* que apoye su conclusión.

c) (2 puntos) Haciendo uso del simulador, determine qué algoritmo minimiza el tiempo medio de retorno. Justifique su respuesta presentando *screenshots* de las simulaciones. En el caso de *round robin* elija el *quantum* más apropiado.

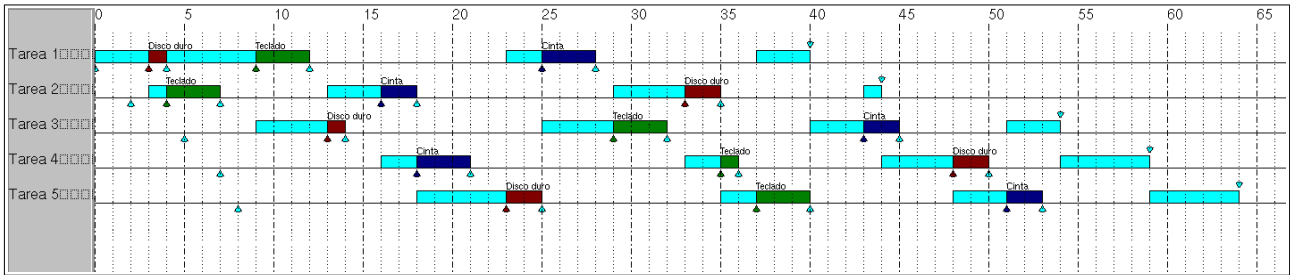
d) (2 puntos) Considerando su respuesta de la parte c), ésta se puede generalizar o es una respuesta para el caso particular presentado en la Tabla 1. Justifique su respuesta presentando *screenshots* que apoye su conclusión.

e) (2 puntos) Si a la Tabla 1 se agregan tiempos de llegada, como se muestra en la Tabla 2, el algoritmo que minimiza el tiempo medio de espera y el algoritmo que minimiza el tiempo medio de retorno, ¿se ven alterados? Justifique su respuesta presentando *screenshots* de las simulaciones.

Proceso	Tiempo de servicio	Tiempo de llegada
A	20	0
B	13	3
C	19	5
D	21	7
E	9	9

Tabla 2

2) (6 puntos) Dada la siguiente salida



a) (2 puntos) Escriba la descripción del archivo que produce esta salida.

b) (2 puntos) ¿Qué algoritmo se ha usado? ¿Hay poca información para determinar el algoritmo? ¿Pueden ser más de un algoritmo, si es así que otros algoritmos pueden ser?

c) (2 puntos) ¿Hay multiprogramación? Justifique su respuesta valiéndose del gráfico. En caso que no haya multiprogramación indique qué sería necesario para que lo haya, emplee el gráfico para ello.

3) (4 puntos) Andrew S. Tanenbaum y Herbert Bos en el libro *Modern Operating System*, definen el efecto *convoy*, que se puede presentar en el algoritmo *First-come/First-server*, con el siguiente ejemplo:

Unfortunately, first-come, first-served also has a powerful disadvantage. Suppose there is one compute-bound process that runs for 1 sec at a time and many I/O-bound processes that use little CPU time but each have to perform 1000 disk reads to complete. The compute-bound process runs for 1 sec, then it reads a disk block. All the I/O processes now run and start disk reads. When the compute-bound process gets its disk block, it runs for another 1 sec, followed by all the I/O-bound processes in quick succession.

The net result is that each I/O-bound process gets to read 1 block per second and will take 1000 sec to finish. With a scheduling algorithm that preempted the compute-bound process every 10 msec, the I/O-bound processes would finish in 10 sec instead of 1000 sec, without slowing down the compute-bound process much.

Elabore el archivo *convoy.def* de forma que la salida del simulador muestre el efecto *convoy*. Los procesos deben incluir carga de CPU y carga de I/O. Acompañe en esta respuesta *screenshots* de las salidas del simulador.

Coloque comentarios apropiados explicando ¿cómo se manifiesta el efecto *convoy* en la simulación? En sus comentarios indique en cada proceso el tiempo de espera.

Recuerde:

Tiempo de retorno, es el tiempo transcurrido desde que llegó hasta que complete su ejecución.

Tiempo de espera, es el tiempo que el proceso estuvo en la cola de listos esperando CPU.

Prof. Alejandro T. Bello Ruiz