PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

INF239 SISTEMAS OPERATIVOS
Semestre 2024-1
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) (12 puntos) Evaluación de algoritmos

Se tiene los siguientes procesos que llegan todos al mismo tiempo

Proceso	Tiempo de servicio
A	15
B	8
C	14
D	16
E	4

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.

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 1.0. 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.

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 1.0. Justifique su respuesta presentando "*screenshots*" que apoye su conclusión.

e) (4 puntos) Si a la tabla 1.0 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	15	0
B	8	2
C	14	4
D	16	6
E	4	8

Tabla 2

2) (4 puntos) *Andrew S. Tanenbaum* y *Herbert Bos* en el libro *Modern Operating System*, definen el efecto convoy, presente 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.

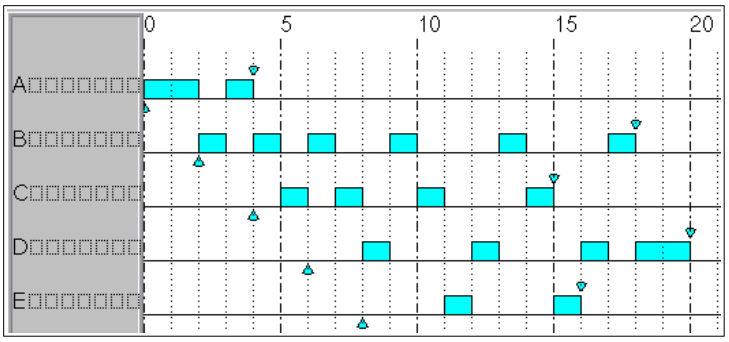
Elabore el archivo *convoy.def* de forma que la salida del simulador muestre el efecto convoy. Los procesos deben incluir carga de CPU y carga de I/O. Acompañe en esta respuesta "*screenshots*" de las salidas del simulador.

3) (4 puntos) Al crear un archivo para la siguiente tabla 3 (***RR***, ***q*** =1):

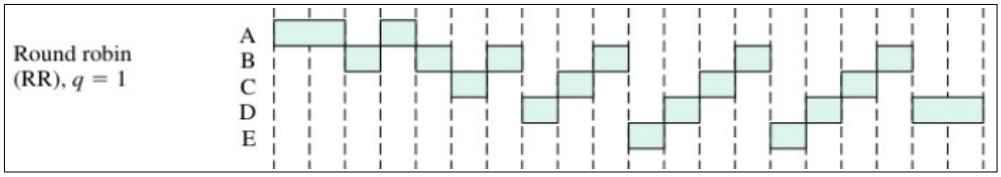
Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2

Tabla 3

Se obtuvo la siguiente salida en el simulador



Mientras que en el libro texto se encuentra la siguiente salida



Explique qué decisión está tomando el algoritmo del simulador para mostrar una salida diferente. Justifique su respuesta con "*screenshots*" que apoyen su respuesta.

Prof. Alejandro T. Bello Ruiz