

# Sistemas Operativos

## 1º Rec 1er Parcial 2C2023 – Resolución

Aclaración: La mayoría de las preguntas o ejercicios no tienen una única solución. Por lo tanto, si una solución particular no es similar a la expuesta aquí, no significa necesariamente que la misma sea incorrecta. Ante cualquier duda, consultar con el/la docente del curso.

### Teoría

1. Si bien una interrupción puede producirse en cualquier instante del ciclo de instrucción, esta será atendida únicamente en la fase posterior a la de ejecución, que está dedicada a el tratamiento de interrupciones.
2. El planificador de mediano plazo influye mucho en el grado de multiprogramación, ya que puede quitar parcialmente procesos de memoria para liberarla y alojarlos en disco, pero si pensamos en un sistema operativo con un diagrama de 5 estados, sin planificador de mediano plazo, el que termina influyendo en el mismo es el de largo plazo, ya que enviara “procesos a memoria” (NEW → READY) o los quitara (x→EXIT)
3. a) Falso, si son ULTs SIN JACKETING, con que un hilo se bloquee, el proceso también se bloqueara.  
b) Falso, ya que los KLTs son conocidos por el sistema operativo, si uno de estos se bloquea, no bloqueara al resto, como sí sucede con los ULTs.
4. a) Falso, no comparten la memoria dinámica, solamente compartirán el código que posean y con respecto a la memoria estática, tendrá una copia de la que tenía el proceso que lo creó.  
b) Verdadero, todo cambio de procesos implica al menos dos cambios de modo de ejecución, ya que es el sistema operativo quien realiza este cambio, por lo que debe de pasar de Modo Usuario → Modo Kernel → Modo Usuario
5. a) Falso, dado que un interbloqueo involucra al menos 2 entidades (procesos o hilos) dado que por definición una entidad posee recursos que la otra requiere y viceversa, en conclusión, en este contexto nunca podría ocurrir.  
b) Verdadero, en el unico caso que un proceso podria sufrir inanición bajo este contexto, es teniendo un sistema en el cual el tratamiento de deadlock no verifique la cantidad de instancias de los recursos que el proceso requiere para su ejecución, de manera tal que pueda querer utilizar más de los que existan y el sistema operativo no mandar a ejecutar dicho proceso.

### Práctica

- 1) Los algoritmos que permiten obtener la traza de ejecución propuesta son:

ULT's con SJF, y los procesos (o KLT's) con un VRR de  $Q=3$

Los instantes que validan esto son los siguientes:

- ❖  $T=6$ , U1 teniendo aún ráfagas de CPU para ejecutar, fue desalojado, además, K2 ya se encontraba en la cola de ready, pero dado que el proceso A no finalizó su quantum, tuvo mayor prioridad.
- ❖  $T=12$ , K2 ya había regresado de la cola de bloqueados por su IO, sin embargo, K1 no completó su Quantum, por ende tuvo mayor prioridad.
- ❖  $T=14$ , similar a  $T=12$ , el proceso A no finalizó su Quantum
- ❖ Para determinar el SJF basta con verificar siempre ejecuto el UTL con menor rafaga de CPU, podría pensarse HRRN, pero dado que U2 sufrió starvation hasta que U1 finalizó, se descarta esta opción.

No podemos determinar con exactitud si el SJF es con o sin desalojo, pero sí podemos afirmar que no implementa jacketing.

2)  $\text{pies[]} = \{0, 2\}$ , toque = 0, volver = 0

La secuencia a realizar es: PD ida, toque, PD vuelta, PD cambiarSentido, PD ida, toque, PD vuelta, PD cambiarSentido, cambiarPieDominante.

Luego repetir la secuencia pero en lugar del PD utilizar PI.

Pisada (1)
<pre>while(1){     wait(toque)     toque(id_pieDominante)     signal(volver)     signal(pies[id_pieDominante])x2     wait(toque)     toque(id_pieDominante)     signal(volver)     cambiarPieDominante() }</pre>

Pie Izquierdo (1)
<pre>while(1){     wait(pies[0]) x2     ida(sentido)     signal(toque)     wait(volver)     vuelta(sentido)     cambiarSentido(&amp;sentido)     signal(pies[1]) }</pre>

Pie Derecho (1)
<pre>while(1){     wait(pies[1]) x2     ida(sentido)     signal(toque)     wait(volver)     vuelta(sentido)     cambiarSentido(&amp;sentido)     signal(pies[0]) }</pre>

3)

- El estado del sistema es seguro. La única secuencia posible es P4-P2-P5-P1-P3.
- No se concedería la petición porque P4 no los declaró en necesidades máximas.