2021 - Sistemas Operativos General

Comenzado el Saturday, 5 de June de 2021, 15:06 **Estado** Finalizado **Finalizado en** Saturday, 5 de June de 2021, 16:16 **Tiempo** 1 hora 10 minutos empleado

Navegación por el cuestionario Mostrar una página cada vez Finalizar revisión

Pregunta **1** Parcialmente correcta Puntúa como

Marcar Marcar

pregunta

Se dispone de varios pacientes que desean atenderse en un consultorio. Existen dos recepcionistas que atienden en simultáneo (empezando por recepcionista nº1), con la orden de no asignar en total más de tres turnos en simultáneo.

Se desarrolló el siguiente pseudo código:

variable_global RESP_ENCUESTA = true;

Paciente (N instancias)	Recepcionista 1 (1 instancia)	Recepcionista 2 (1 instancia)
pedir_turno()		
SIGNAL(A) WAIT(B) atenderse() SIGNAL(C) SYSCALL1() if (RESP_ENCUESTA = = true){ responder_encuesta();	while(TRUE) WAIT(D) WAIT(A) WAIT(C) asignar_turno() SIGNAL(B) SIGNAL(E)	while(TRUE) WAIT(E) WAIT(A) WAIT(C) asignar_turno() SIGNAL(B) SIGNAL(D)
SYSCALL2()		

Se pide responder lo siguiente:

- Los valores iniciales de los semáforos deberían ser A= 0 ✓ , C= 3 ✓ , B= 0
- Podrían sufrir una situación de starvation las instancias del Proceso: Paciente
- Si las llamadas a WAIT(D) y WAIT(A) en Recepcionista 1 fueran alternadas en su orden (esto es, intercambiadas), esto generaría: Comportamiento similar 🗢 🗶
- Si las llamadas a SIGNAL(B) y SIGNAL(E) en Recepcionista 1 fueran alternadas en su orden, esto generaría: Posible bloqueo **\$** X
- *****

Pregunta **2** Parcialmente correcta Puntúa como Marcar

pregunta

Se desea desarrollar un sistema operativo donde los procesos repartan el tiempo de CPU equitativamente, evitando monopolizar la CPU y evitando que los procesos cpu bound (limitados por cpu) perjudiquen a los i/o bound (limitados por i/o). No debería existir planificación de mediano plazo.

Para garantizar la estabilidad del sistema, en el mismo no debería haber más de 5 procesos en ready, 6 en ejecución y 7 bloqueados. En los estados New/Exit no debería haber más de 7 procesos en cada estado.

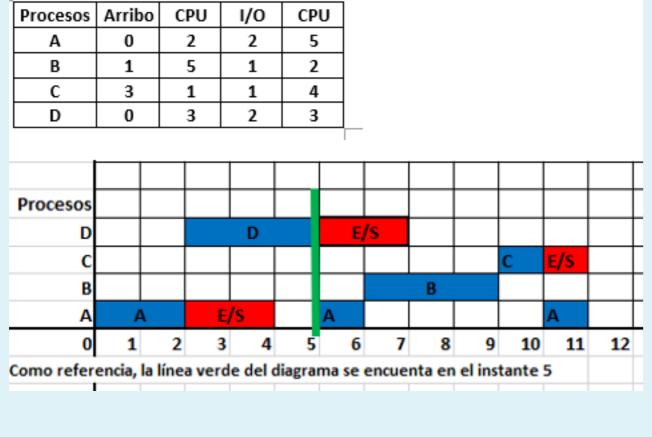
El sistema no tendrá KLTs, pero debería permitir el uso de ULTs.

Responda lo siguiente sobre dicho sistema a diseñar:

- Cantidad necesaria de estados, en el diagrama de estados de los procesos: Ready / Running / Blocked / New / Exit **\$**
- ***** X
- Nivel máximo de multiprogramación que tendrá el sistema: 18
- La información sobre planificación para el planificador de corto plazo del sistema operativo (como por ejemplo: prioridad, quantum ó estimación de ráfaga; según se necesite) deberá ser ubicada en el siguiente elemento de la imagen del proceso: Ninguno 💠 🗶

Pregunta **3** Correcta Puntúa como 1,00 Marcar

Dado el siguiente diagrama y traza de ejecución de 4 procesos:



Nota: ante simultaneidad en tiempos de llegada, se elige por orden alfabético ascendente.

a) Determine qué algoritmo de PCP (Planificador de Corto Plazo) está utilizando: VRR **+ **

Ir a...

b) En base al algoritmo seleccionado en el punto 1, determine por qué en el instante 5 selecciona a ese proceso a ejecutar: Ninguna de las opciones es válida 💠 🗸

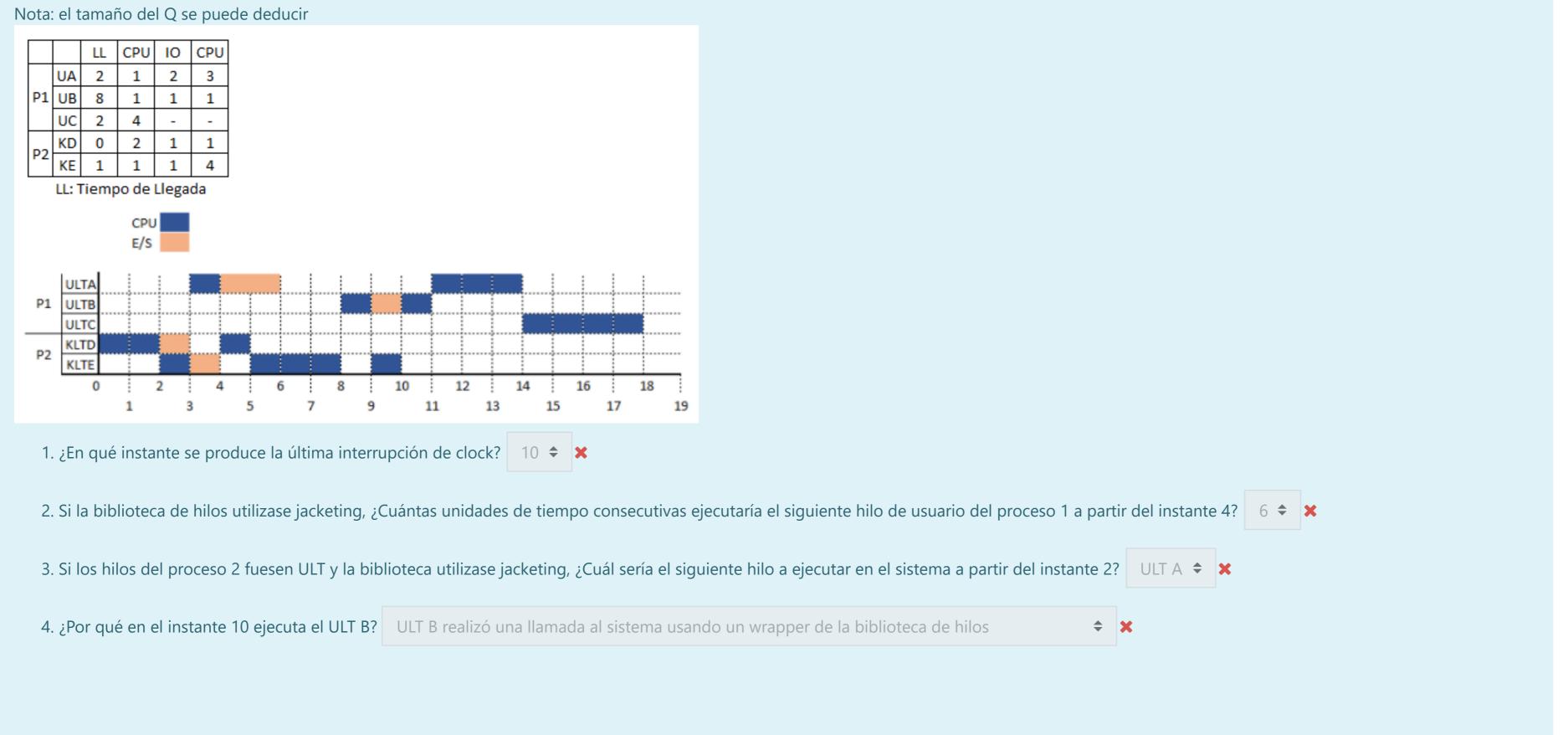
c) Determine qué proceso continuará ejecutando en el instante 11: A 🗢 🗸

d) Determine el orden en que finalizan los procesos: D/A/C/B ◆ ✓

Pregunta 4 Incorrecta Puntúa como Marcar pregunta

■ 2021 - 1C - 1er Parcial - Teoría - TT

Dada la siguiente traza de ejecución, responda las preguntas teniendo en cuenta que el planificador de corto plazo utiliza Round Robin y la biblioteca de hilos de usuario utiliza SJF con desalojo.



Finalizar revisión