CC4302 Sistemas Operativos – Examen – Semestre Primavera 2025 – Profs.: Mateu, Arenas

Pregunta 1

El tipo *Portero* sirve para evitar que entren a una tienda más de *MAX* personas simultáneamente. Se le pide que defina el tipo *Portero* y programe las siguientes funciones:

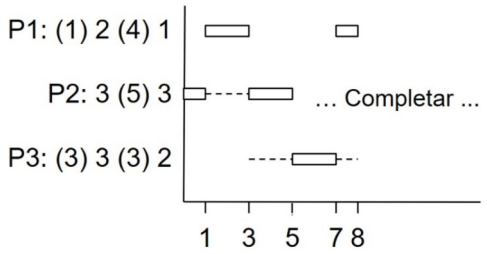
- *void iniPortero(Portero *p, int max)*: Inicializa el portero *p* que permite que ingresen simultáneamente hasta *max* threads.
- *void entrar(Portero *p)*: Solicita al portero *p* ingresar a la tienda. Si ya hay *max* personas en el interior debe esperar.
- *void salir(Portero *p)*: Notifica al portero *p* la salida de la tienda. Si hay threads en espera se hace pasar al thread que lleva más tiempo esperando.

Para resolver este problema **Ud. debe usar los mutex y condiciones de pthreads**. No puede usar otras herramientas de sincronización como semáforos. Además su solución debe evitar cambios de contexto inútiles como que un thread se despierte solo para darse cuenta que debe continuar esperando, y por lo tanto debe usar el patrón *request*.

Pregunta 2

a.- (4 ptos) Resuelva nuevamente el problema del portero pero esta vez con herramientas nativas de nThreads, es decir, utilice operaciones como START_CRITICAL, setReady, suspend, schedule, etc. Ud. debe definir el tipo *Portero* y reprogramar las mismas funciones de la pregunta 1. No puede implementar esta API en términos de otras herramientas de sincronización pre-existentes en nThreads como semáforos, mutex o condiciones. Use el estado WAIT_PASAR.

b.- (2 ptos) En el diagrama parcial se puede apreciar las decisiones de scheduling para 3 procesos. Para cada proceso se indica la duración de la ráfaga y la duración de su estado de espera entre paréntesis.



La línea punteada indica el estado READY. Identifique qué estrategia es la utilizada. ¿Es esta estrategia de scheduling preemptive o non preemptive? Complete el diagrama.

Pregunta 3

i. ¿Tiene sentido usar spin-locks en una máquina monocore? Explique. Responda entonces cómo implementaría la exclusión mutua en un núcleo moderno de Unix para una máquina monocore.

ii. Ud. debe elegir una herramienta para garantizar la exclusión mutua en una sección crítica de un módulo del núcleo de Linux. Considere una máquina multicore. ¿Bajo qué condiciones sería más eficiente usar un spin-lock y cuando sería más eficiente un semáforo?

iii. ¿Qué es lo más importante que perderían los usuarios si los computadores modernos no ofrecieran paginamiento? Explique considerando qué pasa cuando se “caen” aplicaciones como chrome, word, windows media player, waze, whatsapp, etc. debido a problemas con los punteros.

iv. Compare ventajas y desventajas de usar un disco tradicional para paginamiento versus un solid state drive (SSD). En su comparación considere número de page faults atendidos por segundo, velocidad (en MB/seg.) para cargar en memoria un proceso que fue llevado a disco (estado swap) y tiempo de vida del dispositivo de paginamiento.