

Sistemas Operativos I - Recuperatorio - 15/11/2018

- Nombre:
- DNI:
- Firma (**al entregar y ante un docente**):

Notas:

- Los alumnos que obtuvieron una calificación distinto de M (Mal) en el examen de scripting pueden obviar resolver el ejercicio 1).
 - Se solicita realizar el examen con letra clara y legible.
1. Codifique un script que reciba un archivo de texto por parámetro y un valor numérico N , e implemente un algoritmo simple para resumir el texto. Básicamente, el script debe generar un nuevo archivo como salida que elimine todas las frases del archivo que tengan más de N palabras.
El archivo de entrada consta de líneas no vacías, donde cada línea representa una frase del texto completo. Cada línea (esto es, frase) contiene más de una palabra, separadas por un caracter especial (a elección). El script también debe informar por la pantalla la cantidad de líneas del archivo de entrada y del producido como salida.
 2. Considere una feria electrónica donde se exhibe una demo de un videojuego nuevo. A lo largo del día concurren a probar el videojuego jugadores (*threads*), los cuales deben sincronizarse como sigue. Cuando un jugador llega para jugar, se sienta en un área de espera. Cuando hay en dicha área 4 o más jugadores, se les permite a 4 de ellos jugar. Siempre que un jugador se aproxime a la consola, espera a los restantes 3 jugadores para empezar a jugar. Los 4 jugadores pueden abandonar el juego en cualquier momento, pero el juego en sí termina cuando los 4 abandonan. Luego de esto, se les permite a 4 nuevos jugadores en espera a comenzar un nuevo juego.
 - a) Programe una solución evitando *race conditions*, *busy waits*, *deadlocks*, y *maximizando concurrencia*. El código *debe estar comentado*, especialmente las estructuras concurrentes. Debe usar arreglos planos para las estructuras compartidas entre threads (no se permite Vector o listas sincronizadas).
 - b) Explique por qué su solución está libre de deadlocks, relacionando con las condiciones vistas en teoría de acuerdo a los procesos/hilos y recursos a partir de resolver el punto a).
 3. Considere el siguiente conjunto de procesos:

Proceso	Tiempo de CPU requerido (en ms)	Petición de E/S
P1	6	t=4
P2	1	No realiza
P3	2	t=1
P4	2	No realiza
P5	5	t=3

Asuma que en el tiempo cero todos los procesos se encuentran en la cola de listos. Algunos procesos realizan peticiones de E/S en cierto tiempo t (relativo al tiempo de inicio de ejecución del proceso). Asuma que una petición de E/S lleva 2 ms.

- a) Grafique diagramas de Gantt ilustrando la ejecución utilizando el algoritmo SJF sin desalojo, con dos variantes: E/S *sincrónica* (la CPU espera a que la E/S finalice) y *asincrónica* (la petición de E/S se resuelve en paralelo con la CPU).
 - b) Calcule el tiempo promedio de *turnaround* para los procesos de acuerdo a los dos algoritmos graficados.
4. Considere un sistema con direcciones lógicas de 32 bits y físicas de 64 bits. Calcule la *eficiencia* (memoria asignada a un proceso dividida la memoria asignada del proceso más la necesaria para almacenar su tabla), considerando un esquema de paginación con páginas de 4 KB y tabla de páginas de un solo nivel, y procesos de tamaño:
- a) 4 KB
 - b) 4 MB
 - c) 512 MB

Puntaje individual (a completar por la cátedra)

1	2		3		4			Nota final
	a	b	a	b	a	b	c	