



Nota:

Apellido y Nombre	Profesor	Tomé conocimiento de la nota: (Sólo aplazos)

Preguntas teóricas					Ejercicios	
1	2	3	4	5	1	2

A) Teoría: Explícitamente defina como VERDADERA o FALSA cada una de estas afirmaciones justificando su respuesta en no más de 3 líneas.

Peter está configurando un servidor en el cual corren procesos CPU bound constantemente. Al revisar el uso de la CPU nota que pasa el 60% del tiempo ociosa. Entonces, decide subir el grado de multiprogramación, y al revisar la CPU horas más tarde se da cuenta de que ahora el 80% del tiempo está ociosa.

1) Es muy probable que se trate de un caso de thrashing (hiperpaginación).

Llega el momento de decidir qué estrategia para combatir Deadlocks se va a implementar.

2) Para aprovechar los recursos al máximo y que nunca ocurran Deadlocks debe elegir la estrategia de Prevención mediante la alocaación todos los recursos juntos.

Para formatear el disco, debe elegir entre utilizar una única partición que contenga los archivos comunes y el área de swapp, o utilizar dos particiones separadas con posibles formatos distintos.

3) Luego de analizar elige tener una única partición, para mejorar la performance ante la ocurrencia de page faults

Luego necesita definir qué biblioteca de hilos ofrecerá el sistema (KLT/ULT), dado que sólo puede elegir una. Recordando la información leída anteriormente sobre los procesos, Peter revisa la especificación del único procesador que posee y decide que para tener poco overhead:

4) Lo mejor en este caso es elegir la biblioteca de ULTs

Por último, como los hilos aparte de compartir archivos comparten algunas variables globales de tipo contador, decide despreocuparse con respecto al algoritmo de planificación:

5) Sabe que un algoritmo tan sencillo como FIFO no producirá inanición ni problemas de concurrencia en este caso.

B) Práctica: resuelva los ejercicios justificando las respuestas

1) Para probar el servidor con una alta carga de procesos, Peter supervisa una simulación en la que corren dos tipos de hilos, que representan átomos de hidrógeno y átomos de oxígeno, simulando la formación de H2O (agua). La cantidad de hilos que ejecutan concurrentemente es manejada por un proceso, que se ocupa de crearlos ininterrumpidamente, por lo que es posible tener varias instancias de cada hilo. El pseudo-código de “Hidrógeno” y “Oxígeno” es el siguiente:

Hidrógeno	Oxígeno
insertar_atomo_en(COLA);	H1 = sacar_atomo(COLA); H2 = sacar_atomo(COLA); unir_este_atomo_con(H1, H2);

Cada átomo de oxígeno puede comenzar la unión **sólo** si hay dos átomos de hidrógeno disponibles en la variable global COLA. H1 y H2, también son variables globales, por lo que deben ser utilizadas cuidadosamente.

Sincronice la ejecución de estos hilos utilizando semáforos, para que permitan la “formación de agua” de forma ininterrumpida.

2) Se dispone de un SO con paginación, memoria virtual (algoritmo de reemplazo clock), usando un disco con dos particiones, una con datos comunes (1gb) y otra utilizada para intercambio (swapp, 512gb). Teniendo 8gb de RAM y sabiendo que el programa más pequeño ocupa al menos 4Mb en memoria, indique:

- El formato de las direcciones lógicas y físicas de este sistema
- El formato de una entrada de la tabla de páginas, sabiendo que no se puede usar para la seguridad más de 3 bits
- ¿Cual es el principio teórico en el que se basa el éxito de la memoria virtual? Proponga un breve ejemplo de pseudo-código

El tiempo de duración del examen final será de 90' a contar desde el momento de comienzo del mismo. Si el alumno por algún motivo comenzara más tarde sólo podrá utilizar el tiempo remanente. Utilice hojas separadas para la teoría / ejercicios.



Nota:

Teoría:

- 1) Verdadero. Es el ejemplo clásico de la curva de Thrashing.
- 2) Falso. Evasión es mejor opción para aprovechar los recursos sin que ocurran Deadlocks porque va otorgando recursos por solicitudes individuales
- 3) Falso. Utilizar dos particiones (si están en dos discos diferentes, mejor aún) permite usar FS diferentes. Usar un FS con asignación contigua para el disco de swapp suele ser recomendable.
- 4) Verdadero. No existe multiprocesamiento para aprovechar, y las causas de bloqueo (E/S) son pocas porque los procesos son CPU bound. Con UTLs se eliminan los cambios de contexto.
- 5) Verdadero. En FIFO no hay inanición ni problemas de concurrencia con variables compartidas porque el proceso no es interrumpido nunca y no puede dejar el procesador con una variable contador a "medio modificar"

Practica:

1.

Hidrogeno	Oxígeno
w(mutex); insertar_atomo_en(COLA); s(mutex); s(h);	w(h); w(h); w(mutex); H1 = sacar_atomo(COLA); H2 = sacar_atomo(COLA); unir_este_átomo_con(H1, H2); s(mutex);

mutex = 1; h = 0; → Como H1 y H2 son globales pero solo para los hilos Oxigeno, se podría usar un mutex para H1 y H2 aparte, para no bloquear a COLA mientras se une

2.

Páginas de 4 Mb = 2^{22}

- Direcciones físicas: Disponemos de 8 gb (RAM) = 2^{33} bytes. => 11 bits para el frame, 22 bits para el offset
- Direcciones lógicas: Disponemos de 512 gb (área de swapp) = 2^{39} bytes. => 17 bits para la página, 22 bits para el offset

Formato entrada tabla de páginas:

# FRAME (11 bits)	PROTECCIÓN (3 bits)	P (presencia, 1 bit)	M (modificación, 1 bit)	U (uso, 1 bit)
-------------------	---------------------	----------------------	-------------------------	----------------

El principio teórico es el principio de localidad, bajo el cual durante un período de tiempo las referencias de un proceso se encuentran confinadas a un subconjunto reducido de sus páginas.

Ejemplo pseudo-código: int contador = 0; while (contador < 10) { printf("Valor: %d", contador); }