



Nota:

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

Preguntas teóricas					Ejercicios	
1	2	3	4	5	1	2

- A) **Teoría:** Defina explícitamente como VERDADERA o FALSA cada una de estas afirmaciones justificando brevemente.
- 1) Implementar segmentación paginada permite combinar la ventaja de ambos esquemas, eliminando tanto la fragmentación interna como externa.
- 2) Si se copian archivos de un fs de tipo unix a otro de tipo FAT, inevitablemente se perderá cierto tipo de meta-información.
- 3) Una incorrecta configuración del planificador de largo plazo podría generar thrashing (sobrepaginación)
- 4) Una transición entre el estado Running y el estado Ready sólo es posible en algoritmos con quantum.
- 5) En un programa donde existan condiciones de carrera, seguramente se producirán interbloqueos en el corto o largo plazo

B) **Práctica:** Resuelva los ejercicios justificando las respuestas

1) Un sistema maneja su memoria virtual utilizando paginación por demanda, con direcciones lógicas de 16 bits, y asignando hasta 4 frames por proceso, con sustitución local. El proceso A realiza las siguientes referencias: A000h, 0300h, 3500h, 1000h, 4100h y A001h

Página	0	1	2	3	4	5	6	7	8	9
Frame	5	-	-	9	7	1	-	-	-	-

- Sabiendo que todas las referencias son válidas y que la primera (A000h) no produce un fallo, indique la traza de ejecución, señalando los fallos de página que ocurren, para los siguientes algoritmos:
- a) LRU (sabiendo que los frames se cargaron en orden ascendente)
- b) Clock (sabiendo que todos los bits de uso están en cero, y el puntero está en el frame 9)
- 2) Scaloni está pensando en el próximo partido. Para ello pone un equipo con 4 defensores, 3 volantes y 3 delanteros. La mayoría de las veces no lo atacan, el 80% de las veces la pelota circula y casi siempre la pelota entra al arco contrario, pero por si acaso, la AFA contrató a Peter para simular, utilizando el algoritmo que corresponda. Si los procesos se encuentran donde muestra la flecha, ¿está el equipo en estado seguro?

Defender() { Asignar(defensor, 2); => if(nos_estan_atacando) { Asignar(defensor, 2); Asignar(volante, 2); } }	PasarLaPelota() { Asignar(volante, 2); => if(la_pelota_no_circula) { Asignar(volante, 1); Asignar(delantero, 1); } }	PegarleAlArco() { Asignar(delantero, 2); => if(la_pelota_no_entra) { Asignar(defensor, 2); Asignar(volante, 2); } }
--	---	--

Nota: "Asignar" es una función bloqueante, que asigna la cantidad de recursos indicada, hasta finalizar el proceso.