


















Nombre:

Profesor:

Preguntas teóricas							Ejercicios			Nota
1	2	3	4	5	6	7	8	9	10	

Las siguientes preguntas fueron realizadas a ChatGPT. Para cada caso, se pide que analice la respuesta provista.

	[Nº 1] ¿Cuáles son las desventajas de usar hilos a nivel usuario, si se los compara con hilos a nivel kernel?
	<p>Las desventajas de usar hilos a nivel usuario, en comparación con hilos a nivel kernel, son las siguientes:</p> <ol style="list-style-type: none">Falta de soporte para multiprocesadores: Generalmente no se benefician de la asignación directa a CPUs, lo que limita el aprovechamiento total del rendimiento en sistemas multiprocesador.Bloqueo de llamadas al sistema: Puede resultar en bloqueo de hilos, especialmente si uno realiza una operación bloqueante, todos los demás hilos asociados a ese proceso también se bloquearán.Latencia de planificación: La planificación puede tener una latencia más alta, ya que cuando ocurre una interrupción de planificación, el proceso debe realizar una transición de modo usuario a modo kernel para ejecutar el planificador del espacio del kernel, que luego selecciona el siguiente hilo para ejecutar.
	Evalúe la respuesta provista. Encuentre cual(es) aspecto(s) están incorrectos/incompletos y explique porqué.
	[Nº 2] ¿Podrías ordenar las siguientes técnicas de administración de memoria usando como criterio ordenador el overhead?: Particionamiento fijo, Buddy System, Segmentación paginada.
	<p>Ordenando las técnicas de administración de memoria de menor a mayor overhead, tendríamos:</p> <ol style="list-style-type: none">Buddy System: Puede incurrir en cierta sobrecarga debido a la necesidad de mantener el seguimiento de los bloques y realizar operaciones de división y combinación.Segmentación paginada: La administración de segmentos y páginas puede requerir más recursos y operaciones adicionales para la traducción de direcciones y el seguimiento de los límites de los segmentos.Particionamiento fijo: Puede llevar a un desperdicio de memoria debido a la fragmentación interna y la rigidez en la asignación de memoria.
	Evalúe la respuesta provista. En caso de diferir en algún aspecto de la misma, justifique el motivo. En caso contrario, indique en cuál posición ubicaría al esquema de Particionamiento dinámico.
	[Nº 3] En un contexto de un sistema de altas prestaciones y poca experiencia en el desarrollo de software, ¿Convendría usar syscalls bloqueantes o no bloqueantes?
	<p>Convendría utilizar syscalls bloqueantes, ya que son más fáciles de manejar y comprender para desarrolladores con poca experiencia.</p>
	Considerando la respuesta como adecuada, elabore en la justificación explicando con mayor detalle (considere proveer algún ejemplo con pseudocódigo), teniendo en cuenta los posibles cambios de estado del proceso.
	[Nº 4] ¿Por cual motivo convendría, al operar concurrentemente sobre un mismo archivo, usar locks en lugar de semáforos mutex convencionales?
	<p>Existen algunas consideraciones que podrían favorecer el uso de locks sobre semáforos mutex en esta situación:</p> <ol style="list-style-type: none">Claridad del código: El uso de locks sobre archivos puede hacer que el código sea más claro y expresivo, ya que muestra explícitamente que el objetivo es controlar el acceso a un archivo específico.
	Provea al menos un motivo más para contribuir la respuesta provista. Indique si las respuestas cambiarían si se usaran locks sugeridos u obligatorios.
	[Nº 5] Si dos procesos quedan en un interbloqueo, ¿Podría uno de ellos romper la espera circular?
	<p>Sí, en ciertas situaciones, uno de los procesos involucrados en el interbloqueo puede romper la espera circular y resolver la situación de interbloqueo. Para que esto suceda, se debe aplicar alguna estrategia de prevención o resolución de interbloqueos. Una de las técnicas más comunes para resolver un interbloqueo es implementar un algoritmo del banquero.</p>
	Evalúe la respuesta. En caso de considerarla correcta explique cuál principio sustentaría dicha afirmación, en caso contrario explique cuál aspecto de la respuesta es incorrecto.

	[N° 6] ¿Sería posible lograr los beneficios de los esquemas de memoria virtual (como paginación) sin la existencia de direcciones lógicas?
	<i>No, no sería posible lograr los beneficios de los esquemas de memoria virtual, como la paginación, sin la existencia de direcciones lógicas. La memoria virtual, en esencia, es una abstracción que permite que el espacio de direcciones que ve un proceso en ejecución (direcciones lógicas) sea diferente del espacio físico real de memoria. Esto permite que los programas se ejecuten en un espacio de direcciones más grande y contiguo, independientemente de cómo se distribuye físicamente en RAM.</i>
	Evalúe la respuesta. En caso de considerarla correcta, explique la relación de las direcciones lógicas con los fallos de página. En caso de considerarla incorrecta, explique cuál aspecto de la respuesta está equivocado.
	[N° 7] En un SO moderno, tomando como ejemplo Linux, ¿Podría un proceso ejecutar una instrucción privilegiada?
	<i>No, en un sistema como Linux, los procesos en modo de usuario no pueden ejecutar instrucciones privilegiadas directamente. Las mismas están restringidas al modo de kernel, y cualquier intento de ejecutar una instrucción privilegiada desde el modo de usuario resultaría en interrupción.</i>
	Se pide: indique si considera la respuesta correcta o incorrecta. En caso de considerarla correcta, explique que ocurriría luego de la interrupción y evalúe si no sería más eficiente que el SO valide la instrucción. En caso contrario, justifique explicando porqué sería posible, indicando qué requisitos serían necesarios.
	[N° 8] Dados los siguientes pseudo-códigos, cada uno perteneciente a un hilo de un mismo proceso en ejecución: <div><div>LISTA; // Variable global (P: productor, C: Consumidor)</div><div><div>P: while(TRUE){ agregar(LISTA, new_value()) }</div><div>C: while(TRUE){ e = sacar(LISTA) }</div></div></div> Modifíquelo de manera tal que el productor pueda agregar elementos libremente, mientras que el consumidor retire los mismos cuando exista al menos uno en la lista, evitando condiciones de carrera. Utilice solamente semáforos.
	<i>Aquí está el pseudo-código modificado:</i> <div><div>tLista LISTA; semaphore lleno = 0; semaphore vacio = 1;</div><div><div>while (TRUE) wait (vacio); agregar_en_lista (LISTA, new_value()) signal (lleno)</div><div>while (TRUE) wait (lleno) e = sacar (LISTA) signal (vacio)</div></div></div>
	Analice dicha solución y responda ¿Se han cumplido los requisitos pedidos?¿Existe algún problema relacionado a la concurrencia? En caso de ser necesario, provea un nuevo pseudo-código con las correcciones necesarias.
	[N° 9] Se desea copiar un archivo de 11 KiB de un sistema de archivos FAT32 con clusters de 4KB a otro formateado con EXT2, el cual cuenta con 10 punteros directos, un indirecto simple y un indirecto doble (y para el cual los punteros son de 64 bits y que el tamaño de bloque es de 1 KiB). ¿Es posible copiar dicho archivo? En caso afirmativo, indique cuántos bloques en total serían necesarios para almacenar la copia.
	<i>Para almacenar el archivo de 11 KiB en el sistema de archivos EXT2, necesitaríamos:</i> <ul style="list-style-type: none">• 10 bloques para los punteros directos (apuntando a 10 bloques de datos).• 1 bloque adicional para el puntero indirecto simple (que apunta a 10 bloques más).• 1 bloque más para el bloque de punteros indirecto doble• 10 bloques de punteros indirectos simples (que apuntan a 10 bloques de datos adicionales). <i>Por lo tanto, el total es de 22 bloques en el sistema de archivos EXT2 para almacenar la copia del archivo de 11 KiB.</i>
	Se pide: explique si la solución es correcta. En caso negativo, señale el error y provea la respuesta correcta.
	[N° 10] Dado un planificador de corto plazo de tipo RR con un Q=2, indique cuántas veces tuvo que ejecutar el SO, considerando la siguiente traza: P1: CPU(3), I/O(1), CPU(1); - P2: CPU(1), I/O(1), CPU(3) Asuma que los dos procesos se encuentran listos para ejecutar en el orden: P1, P2. Asuma que cada involucramiento del SO implica 1 unidad de tiempo.
	<i>Para determinar cuántas veces tuvo que ejecutar el SO en el planificador Round Robin, necesitamos contar el número de cambios de contexto que ocurren cuando se realiza el cambio de proceso. Los cambios de contexto ocurren cada vez que un proceso se bloquea por una operación de entrada/salida (I/O) o cuando se agota su quantum de tiempo y se debe cambiar al siguiente proceso en la cola. En total, hubo 5 veces donde tuvo que involucrarse el SO</i>
	Se pide: evalúe si la respuesta es correcta, realizando un diagrama de gantt.