

Apellido y Nombre: Profesor:

A				B		NOTA	CORRECTOR	ALUMNO
1	2	3	4	5	1	2		
								(Sólo en caso de revisión de examen)

Explicitamente defina como VERDADERO o FALSO cada una de las siguientes afirmaciones FUNDAMENTANDO su respuesta en no más de 3 líneas en ambos casos.

A.1 Una de las ventajas de la encriptación asimétrica (o de clave pública / privada) sobre la encriptación simétrica es que la primera brinda un grado de seguridad mayor frente al criptoanálisis.

FALSO ambas técnicas de cifrado brindan un alto grado de seguridad frente al criptoanálisis.

A.2. El uso de memoria virtual permite siempre ejecutar un mayor número de procesos simultáneamente de manera más eficiente que si se tiene sólo memoria principal.

VERDADERO con MV un proceso no necesita estar completamente en M. Principal para ser ejecutado, por lo tanto, pueden existir mas procesos listos para ejecutar y poder tener el procesador ocupado, aunque la eficiencia podría caer por el fenómeno de thrashing.

A.3. La técnica de buffering de entrada / salida permite incrementar la eficiencia del sistema operativo y la performance individual de los procesos en un ambiente multiprogramado.

VERDADERO: Si se trabaja con MV, el SO puede enviar a disco la página del proceso que está haciendo la E/S (ya que la E/S se hace en páginas lockeadas del SO), entonces se incrementa la eficiencia del resto de los procesos al tener mas espacio en memoria para su ejecución.

2do. Las grandes diferencias de velocidades de los dispositivos con respecto al bus del sistema hacen necesario el uso del buffering, lo que permite utilizar técnicas de E/S por interrupciones o por DMA.

A.4. Una de las ventajas del uso de kernel level threads (KLTs) es que el cambio de threads (Thread switch) de un mismo proceso es más rápido que el cambio entre procesos (process switch) ya que los primeros comparten el stack (pila) del proceso.

FALSO: si bien es cierto que el Thread switch es mas rápido que el process switch por el tamaño de su pila y porque no se actualiza el "process control information" del PCB (solo se cambian los registros dentro del address space) y no es cierto que comparten la pilas, cada thread tiene su propia pila

B) Responda la siguiente pregunta en no más de 5 renglones.

Usted es el Gerente del área de Sistemas de una importante compañía de retail. Se encuentra a punto de implementar un nuevo sistema operativo que brindará servicio a los puntos de atención al público. Desde las terminales de venta se ejecutan 2 tipos de procesos: aquellos que permiten el pago de facturas de servicios (como ser la cuenta de gas o de teléfono) y los procesos de cierre que permiten contabilizar los ingresos registrados por un cajero.

Si mensualmente se supera un determinado número de transacciones de cobro de facturas, todos los empleados reciben un bono, por lo cual es importante que el punto de venta se encuentre ocupado por un mismo cliente el menor tiempo posible. Por otro lado, el proceso de cierre se ejecuta mientras que el cajero realiza el recuento de efectivo y junta sus pertenencias de la caja, que se encuentra cerrada a la atención al público. Este proceso suele demorarse 1 o 2 minutos, en promedio.

El sistema operativo que está a punto de implementar permite seleccionar cuál será el algoritmo que utilizará el planificador a corto plazo al momento de la instalación. Entre las opciones se encuentran el Round Robin con quantum igual a 4, el FIFO y el SRT. Ud. debe decidir cuál será el algoritmo que configurarán en el servidor, teniendo en cuenta ÚNICAMENTE los datos de este problema. Fundamente su respuesta.

Para que el punto de venta se encuentre ocupado por un proceso el menor tiempo, se debe tener un alto **RESPONSE TIME**.

a) El proceso de cierre va después de los procesos de facturación, por lo que se puede utilizar FIFO para que cada transacción de pago se realice sin interrupciones y que el proceso de cierre se ejecute al final, cuando la caja se haya cerrado y no haya mas procesos de facturación.

b) La solución correcta es si se usara SRT se le daria prioridad a los clientes con transacciones rápidas, minimizando su tiempo de finalización. Así estos clientes están poco tiempo. Esto en promedio haría menor el tiempo en que están en servicio los clientes. Si bien SRT produce starvation en los procesos largos (Proceso de cierre) si llegaran muchos procesos cortos (Facturación), pero este no es el caso del problema porque los procesos largos se ejecutan cuando la caja cierra y no se reciben mas pagos.

C) Resuelva en forma clara y detallada los siguientes ejercicios.

C.1. Un sistema cuenta con un disco rígido con las siguientes características: Tiempo "pista a pista": 1ms; tiempo de 1 rotación entera: 10ms; Sectores de 512 Bytes; Sectores por pista: 10; cilindros: 100; Cabezas: 2. Asumiendo que los pedidos que solicitan los procesos P1 y P2 están al mismo tiempo en el buffer de pedidos, calcule el tiempo necesario para satisfacer dichos pedidos sabiendo que el sistema operativo implementa el algoritmo F-SCAN y que en un momento dado los procesos quieren leer los archivos representados por las siguientes tablas:

Archivo 1: registros de 512 bytes

Registro	0	1	2	3	4	5
Dirección lógica	10	20	15	22	100	50

Archivo 2: registros de 1024 bytes

Registro	0	1	2	3	4	5	6	7
Dirección lógica	12/13	52/60	80/81	110/51	200/201	11/101	204/205	202/203

P1: Leer (Archivo1, 0, 2) siendo (archivo a leer, primer registro a leer, cantidad de registros a leer)

P2: Leer (Archivo2, 4, 2)

Asuma que la cabeza de lectura / escritura se encuentra en el sector físico (0, 0, 1), que la organización de los registros de los archivos es contigua y que el brazo se mueve hacia los cilindros mayores.

DATOS:

Seek time = 1ms

Rotation time = 10 ms

Sectores por pista = 10 por lo que tarda un sector = 1 ms.

Cilindros = 100

Cabezas = 2

Sectores = 512 By

En un segundo hace 100 rotaciones,

Transfer time = (cant. By)/(trps * By por Pista) = 512/(100*5120) = 1 ms por sector

Se asume que los pedidos están al mismo tiempo en el buffer, el F-SCAN degenera en SCAN

Pedidos Lógicos: 10, 20

200, 201, 11, 101

Dir. Lógica	Dir. Física
10	(0,1,1)
20	(1,0,1)
200	(10,0,1)
201	(10,0,2)
11	(0,1,2)
101	(5,0,2)

Por ser SCAN los pedidos se ordenan: 10, 11, 20, 101, 200, y 201.

Direcciones	Seek time	Rotation Delay Time	Transfer time	Observaciones
(0,0,1) → (0,1,1)	0 ms	0ms	1 ms	Están en el buffer
(0,1,2) → (0,1,2)	0 ms	0 ms	1 ms	idem
(0,1,3) → (1,0,1)	1 ms	7 ms	1 ms	
(1,0,2) → (5,0,2)	4 ms	6 ms	1 ms	
(5,0,3) → (10,0,1)	5 ms	3 ms	1 ms	
(10,0,2) → (10,0,2)	0 ms	0 ms	1 ms	
Subtotales:	10 ms	16 ms	6 ms	
Tiempo total:	10 ms + 16 ms + 6 ms = 32 ms			

Respuesta: el tiempo necesario para satisfacer dichos pedidos es de 32 ms.

- C.2. Indicando la traza de ejecución, muestre en forma CLARA de que manera se ejecutarán los siguientes procesos, considerando que lo hacen concurrentemente en un sistema multiprogramado. Detalle que procesos finalizaron y cuáles no y por qué razón. Para que se considere aprobado el punto deberá justificar su conclusión con un grafo de asignación de recursos, caso contrario el ejercicio se evaluará como incorrecto en su TOTALIDAD. Adicionalmente, considere que el sistema operativo no libera los recursos que tienen asignados los procesos cuando finalizan.

Inicialización de los semáforos:

I, B, A, C, P = 0
M, K, G = 1

P1	P2	P3	P4	P5	P6
D(I)	D(M)	U(B)	D(A)	U(C)	D(K)
U(I)	U(A)	D(C)	D(A)	D(B)	U(A)
D(P)	D(I)	D(G)	U(I)	D(M)	D(I)
U(P)	U(I)	U(G)	D(K)	D(G)	U(I)
D(K)	D(K)		U(K)	U(M)	D(M)
		U(M)	D(P)	D(K)	U(K)
			U(P)	U(G)	U(P)

instante	READY QUEUE	RUNNING	SEMAFOROS							COLA DE SEM.	Estado
			I	B	A	C	P	M	K		
t ₀	P1,P2,P3,P4,P5,P6		0	0	0	0	0	1	1	1	
t ₁	P2,P3,P4,P5,P6	P1(DI)	-1								I=[P1]
t ₂	P3,P4,P5,P6	P2(DM)						0			M=()
t ₃	P3,P4,P5,P6	P2(UA)			1						A=()
t ₄	P3,P4,P5,P6	P2(DI)	-2								I=[P1,P2]
t ₅	P4,P5,P6	P3(UB)	1								B=()
t ₆	P4,P5,P6	P3(DC)			-1						C=[P3]
t ₇	P5,P6	P4(DA)		0							A=()
t ₈	P5,P6	P4(DA)		-1							A=[P4]
t ₉	P6	P5(UC)			0						C=()
t ₁₀	P6, P3	P5(DB)	0								B=()
t ₁₁	P6, P3	P5(DM)					-1				M={P5}
t ₁₂	P3	P6(DK)						0			K=()
t ₁₃	P3	P6(UA)		0							A=()
t ₁₄	P3,P4	P6(DI)	-3								I=[P1,P2,P6]
t ₁₅	P4	P3(DG)						0			G=()
t ₁₆	P4	P3(UG)						1			G=()
t ₁₇	VACIA	P4(UI)	-2								I=[P2,P6]
t ₁₈	P1	P4(DK)					-1				K=[P4]
t ₁₉	VACIA	P1(UI)	-1								I=[P6]
t ₂₀	P2	P1(DP)				-1					P=[P1]
t ₂₁	VACIA	P2(UI)	0								I=0
t ₂₂	P6	P2(DK)					-2				K=[P4,P2]
t ₂₃	VACIA	P6(UI)	1								I=0
t ₂₄	VACIA	P6(DM)					-2				M={P6}

Resumen

PROCESO P3

TERMINA EN EL INSTANTE 16

PROCESO P1

ESTA BLOQUEADO ESPERANDO QUE SE LIBRE EL RECURSO P

PROCESO P5

ESTA BLOQUEADO ESPERANDO QUE SE LIBRE EL RECURSO M

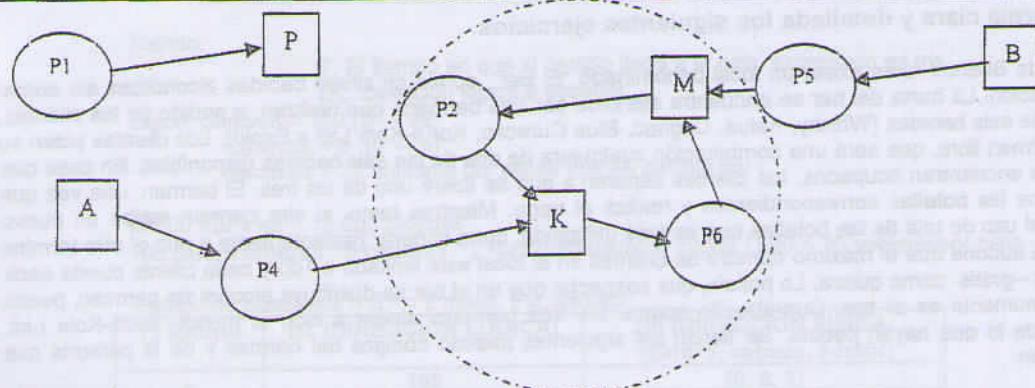
PROCESO P3

ESTA BLOQUEADO ESPERANDO QUE SE LIBRE EL RECURSO K

LOS PROCESOS P2 Y

ESTAN EN DEADLOCK

P6



OBSERVACIÓN:

Los procesos P1, P4 Y P5 ESTAN BLOQUEADOS aunque la liberación de recursos (P, B y A) depende del orden de ejecución externa o de los procesos involucrados en el conflicto, técnicamente no están involucrados en el DEADLOCK como si lo están M y K.

bC.3. Dos usuarios X e Y disponen de las siguientes cuatro herramientas: a) Algoritmo de hashing H unidireccional de generación de claves en base a un documento; b) Un algoritmo S de encriptación de claves simétricas; c) Un algoritmo A de encriptación de claves asimétricas; d) Una red de comunicaciones pública que los vincula.

X desea enviarle un documento D a Y. Ese envío debe asegurar que solo X e Y pueden leer el contenido de D, debe asegurarse a Y que el origen es X y debe asegurar la consistencia de D. Explique paso a paso las acciones que realizan los dos usuarios para lograr que Y obtenga a D con las condiciones anteriormente detalladas. En su explicación aclare como se van cumpliendo las condiciones.

Para que se considere correcto el punto deberá garantizar UNICAMENTE lo que se pide. Si se garantiza más o menos de lo pedido, se considerará el punto EN SU TOTALIDAD como incorrecto.

Condición de aprobación del examen: Deberá tener como mínimo 3 (tres) preguntas teóricas correctamente resueltas y 1 (uno) ejercicio práctico. Recuerde que el primer tema de U2 fue "The City Of Blinding Lights". Sólo se considerará como respuesta lo que esté escrito en este final.

Apellido y Nombre:.....

Profesor:.....

A					B		NOTA	CORRECTOR	ALUMNO
1	2	3	4	5	1	2			
									(Sólo en caso de revisión de examen)

- A) Explicitamente defina como VERDADERO o FALSO cada una de las siguientes afirmaciones FUNDAMENTANDO su respuesta en no más de 3 líneas en ambos casos.

A.1. Para realizar un rollback sobre un proceso que generó deadlock en el sistema es necesario llevar un backup periódico.
FALSO: backup es una copia de resguardo (seguridad), para hacer un rollback es necesario guardar el estado anterior del sistema, incluyendo el estado de los procesos.

A.2. La planificación por prioridades preemptive puede adolecer de problemas de starvation.
VERDADERO: esta planificación sufre de problemas de starvation cuando existen procesos de bajas prioridades encolados y llega un flujo constante de mayor prioridad

A.3. Un procesador que ejecuta código de usuario, sólo podrá realizar un mode switch si se produce un system call.
FALSO: también puede realizar un mode switch en otros casos como por ejemplo ante la aparición de una interrupción o un trap.

A.4. El Bussy Waiting se produce siempre que un proceso ejecuta un ciclo y tiene como desventaja que empeora la performance del sistema.

FALSO: un proceso con espera activa empeora la performance del sistema porque usa tiempo de procesador para esperar un recurso.

A.5. Una de las ventajas de los sistemas distribuidos con un servidor central de procesamiento es que los mismos tienen por característica que son siempre faulttolerant.

FALSO: los sistemas distribuidos tienen procesamiento distribuido en distintos equipos y no tienen un servidor central que por otro lado si ese servidor se "cae" deja de funcionar el sistema, entonces no es fault tolerant.

B) Resuelva en forma clara y detallada los siguientes ejercicios.

B1. En la Provincia de Buenos Aires existe un local denominado "El Bar", donde se sirven bebidas alcohólicas sin contar con la debida habilitación. La barra del bar se encuentra atendida por tres bartmans que realizan, a pedido de los clientes, tragos con mezclas de seis bebidas (Whisky, Kaiua, Cognac, Blue Curacao, KoKa-Kola Lait y Espid). Los clientes piden su consumición a un barman libre, que será una combinación cualquiera de dos de las seis bebidas disponibles. En caso que todos los bartmans se encuentren ocupados, los clientes esperan a que se libere uno de los tres. El barman, una vez que recibe el pedido, toma las botellas correspondientes y realiza el trago. Mientras tanto, si otro barman recibe un nuevo pedido que requiere el uso de una de las botellas que se está utilizando, debe esperar pacientemente a que el otro termine de hacer su trago. Se supone que el máximo número de clientes en el local está limitado a 100 y cada cliente puede pedir tantas combinaciones -gratis- como quiera. La policía, que sospecha que en el bar se distribuye alcohol sin permiso, puede entrar en cualquier momento en el bar. Cuando ello ocurre, los tres bartmans sirven a todo el mundo Koca-Kola Lait, independientemente de lo que hayan pedido. Se tienen los siguientes pseudo códigos del barman y de la persona que simulan dicha situación:

Valores iniciales de los semáforos:

Cant-cli = 100 (semáforo general)

Cant-Barman = 3 (semáforo general)

Botellas [6] = {1,1,1,1,1,1} (semáforos Mutex)

Preparado = 1 (semáforo general)

Proceso_Persona (numero_cliente)	Proceso_Barman ()
<pre> while(1) { down (cant-cli); down (cant-Barman); pide_consumicion (numero_cliente, bebida1, bebida2); Up (Pedido) Down (espera-trago); bebe(); Up (cant-Barman); Up (cant-cli); } } </pre>	<pre> while(1) { down (pedido); recibe_pedido (numero_cliente, bebida1, bebida2); if (policia) { down (KOKA LAIT); real1= "KOKA LAIT"; up (KOKA LAIT); real2= " "; } else { real1= bebida1; real2 = bebida2; } down (preparado); down (botella[real1]); down (botella[real2]); Realizar_preparado(); Up (preparado); Sirve (mezcla (real1,real2)); Up (botella[real1]); Up (botella[real2]); Up (espera-trago); } </pre>

Teniendo en cuenta los datos mencionados, se pide que incorpore los semáforos que crea conveniente, declarando el valor y el tipo de cada uno de los mismos, para que las acciones se lleven a cabo normalmente.

- B2. Se tiene un disco rigido de 48000 KB cuyo cabezal tarda 2 ms en pasar de una pista a otra. Las cabezas leen del sector más chico hacia el más grande (1,2,...,n). La configuración del disco es la siguiente: 6 pistas por cilindro, 100 cilindros y un tamaño de sector de 4KB. La cabeza se encuentra en la dirección lógica 100, ascendiendo. Los pedidos a disco que realiza un proceso en el Sistema se encuentran representados por la siguiente tabla:

T	0	1	4	4	5	7	8	8
DL	200	380	240	390	490	310	20	150

Siendo:

T: El tiempo en que el pedido llega a la cola, expresado en ms.

DL: Dirección lógica a acceder.

Disco: cilindros = 100 (0 — 99)

Cabezas = 6 (0 — 5)

Sectores = 20 sectores por pistas que se calcula por :

$$48000 \text{ kB} / 4 \text{ kB} = 12.000 \text{ bloques}$$

$$100 \text{ pistas/cabezas} * 6 \text{ cabezas} * ?\text{sectores/pistas} = 12000 / 600 = 20 \text{ sectores por pista}$$

Tiempo de búsqueda = ST = 2 mseg. Por cilindro

T: tiempo	DL (DIRECCION LOGICA)	DF (DIRECCION FISICA: cilindro, cabeza, sector)
-	100	(0, 5, 1)
0	200	(1, 4, 1)
1	380	(3, 1, 1)
4	240	(2, 0, 1)
4	390	(3, 1, 11)
5	490	(4, 0, 11)
7	310	(2, 3, 11)
8	20	(0, 1, 1)
8	150	(1, 1, 11)

B2.1. Si se sabe que se tiene un Sistema Operativo que utiliza el C – LOOK como política de planificación de disco, indique el orden en que fueron atendidos los pedidos.

T	cilindro	Pedidos atendidos	cola
0	0	-	200 (1)
2	1	200	380 (3)
4	2	240	380 (3), 240 (2), 390 (3)
6	3	380, 390	490 (4)
8	4	490	490 (4), 310 (2), 20 (0), 150 (1)
16	0	20	310 (2), 150 (1)
18	1	150	310 (2)
20	2	310	

RESPUESTA AL PUNTO B2.1

ORDEN DE ATENCIÓN: 200, 240, 380, 390, 490, 20, 150, 310.

- * B2.2. Considerando para este punto que la velocidad de rotación del disco es de 6000 RPM, y se ha cambiado el algoritmo de planificación de disco a SCAN, indique el Tiempo medio de atención de los siguientes pedidos: 400, 124, 360 considerando que la diferencia de tiempo de llegada entre pedidos es de 4 ms y que la cabeza se encuentra nuevamente en la dirección lógica 100, ascendiendo.

DISCO: 6000 REVOLUCIONES POR MINUTO → 100 REV. POR SEG. → 1 REV. EN 10 mseg.
20 sectores en 10 mseg → 1 sector = 0,5 mseg.

Seek time = ST = 2 mseg por cilindros
Rotation time = RT = 10 mseg por giro
Posición actual : (0, 5, 1)

LLEGADA	DL	DF
0	100	(0, 5, 1)
0	400	(3, 2, 1)
4	124	(1, 0, 5)
8	360	(3, 0, 1)

PLANIFICACIÓN SCAN

LLEGADA	DL	DF	destino	ST (tiempo de búsqueda)	RT (tiempo rotacional)	TL (tiempo de lectura)	TA (tiempo de atención)
0	100	(0, 5, 1)	(3, 2, 1)	-	-	-	0
Cola: (3,0,1) (1,0,5) entonces (3, 2, 2) — (3,0,1)							
0	400	(3, 2, 2)	(3, 0, 1)	6	4	0.5	10.5
8	360	(3, 0, 2)	(1, 0, 5)	0	9.5	0.5	10
Debe ir hasta el último cilindro y volver atendiendo hacia abajo (3, 0, 2) — (99, x, x) tarda 192 mseg — (99, x, 6) — (1,0, 5) en que tarda 196 ms + 2 ms en total 390 ms de ST							
4	124	(1, 0, 5)	(1, 0, 18)	390	1.5	05	392
Tiempo medio de atención:				$(10.5 + 10 + 392) / 3 = 137.5 \text{ ms}$			

Condición de Aprobación: Para aprobar este examen deberá tener como mínimo 3 (tres) preguntas teóricas bien respondidas y 2 (dos) ejercicios correctamente resueltos.

Apellido y Nombre:..... Profesor:.....

A					B		NOTA	CORRECTOR	ALUMNO
1	2	3	4	5	1	2			

(Sólo en caso de revisión de examen)

A) Explicitamente defina como VERDADERO o FALSO cada una de las siguientes afirmaciones FUNDAMENTANDO su respuesta en no más de 3 líneas en ambos casos.

A.1. En un sistema multiprocesador, deshabilitar las interrupciones de un procesador es suficiente para solucionar el problema de la sección crítica.

FALSO deshabilitar las interrupciones de un procesador no es suficiente para solucionar el problema de la sección crítica ya que no garantiza la mutua exclusión para otros procesos que pueden ejecutarse en otros procesadores.

A.2. La prepaginación reduce el número de page faults.

FALSO ya que se pueden traer páginas anticipadamente que luego no sean referenciadas por el proceso, por lo que habrá que sustituirlas provocando page fault.

A.3. Dado que la TLB (Translation Lookaside Buffer) contiene como entradas las últimas direcciones lógicas referenciadas, se reduce la frecuencia de acceso a la memoria principal para recuperar las entradas de páginas.

FALSO la TLB reduce la frecuencia de acceso a la memoria principal pero no contiene direcciones lógicas sino el número de páginas y sus marcos correspondientes

A.4. El planificador de corto plazo (Short Term Scheduler) es invocado solamente cuando se produce una interrupción de reloj.

FALSO se invoca la ejecución del planificador de corto plazo para decidir que proceso continuará en uso del procesador después de un context switch y esto ocurre frente a cualquier interrupción.

A.5. Una de las tareas del procesador al tratar una interrupción es asignar al contador de programa (Program Counter) la dirección de inicio del programa de manejo de interrupción (interrupt handler).

VERDADERO cuando se produce una interrupción se guarda el contexto del proceso en ejecución para invocar la rutina de interrupción por lo que se carga la dirección de inicio de dicha rutina y comienza la ejecución del servicio de la interrupción.

B) Resuelva en forma clara y detallada los siguientes ejercicios.

NOTA: Cada punto práctico será considerado como correcto si y sólo si todas las preguntas que lo componen están contestadas satisfactoriamente.

B1. Bono y The Edge se están preparando para su presentación en el estadio de River. Debido a ello, están supervisando el estado de los sistemas que controlan la consola de sonido, los cuales tienen un sistema operativo que utiliza páginas para la administración de memoria y cuya dirección está compuesta de 32 bits, de los cuales 20 son para el índice y 12 para el desplazamiento. El sistema cuenta con 12 frames de memoria disponibles para procesos y 4 para el sistema operativo. En todo momento, se ejecutan en la consola 2 procesos. El estado es el siguiente:

P1	P2
Read(2151)	Down(B)
Down(A)	Write(13333)
Write(1536)	Up(A)
Up(C)	Write(19000)
Read(20480)	Down(C)
Down(B)	Read(560)
Read(512)	Up(B)
Up(A)	Write(195)
Write(7321)	Down(A)
Down(C)	Read(2560)
Read(10500)	Up(C)
	Read(4096)

Estado inicial de los frames					
P1			P2		
Página	Frame	Presente	Página	Frame	Presente
0	1	1	0	7	1
1	4	1	1	2	1
2	10	1	2	6	1
3	5	0	3	4	1
4	6	0	4	4	0

Últimas referencias

P1	P2
(2-0-1)	(2-0-3)

Estado de los semáforos: A=C=0;B=1;Mutex

Con la información de la consola, conteste:

B.1.1) ¿Cuántos fallos de página se producen para cada proceso?:

- i) Si el módulo utiliza como algoritmo de víctima FIFO – Asignación fija – Alcance Global.
- ii) Si el módulo utiliza LRU – Asignación fija – Alcance Local.

B.1.2) ¿Se puede producir deadlock?

B.1.3) ¿Terminan de ejecutar? ¿Por qué?. Indique la traza de ejecución de los procesos.

B.1.4) ¿Con qué tema empezó el show de U2?

SOLUCIÓN

Datos: direccionamiento 32 bits 20 bits para indice y 12 para offset, tamaño de página 4 KBy

Traducciones de direcciones lógicas a físicas

Proceso 1		
Dir física	Page	Offset
2151	0	2151
1536	0	1536
20480	5	0
512	0	512
7321	1	3225
10500	2	2308

Proceso 2		
Dir física	Page	Offset
13333	3	1045
19000	4	2616
560	0	560
195	0	195
2560	0	2560
4096	1	0

B2. Mientras 60.000 personas cantaban sin parar, Mick se cambiaba en el backstage del escenario y Keith ya comenzaba a entonar las tercera estrofas de "This Place Is Empty". El público deliraba. Los plomos se estaban preparando para la próxima canción, la que llevaría a Mick, a Keith, a Ronnie y a Charlie al centro del campo. La lluvia no paraba de caer. El sistema que controlaba el escenario móvil estaba de 8 dispositivos de desplazamiento (DD), 7 controladores de posición (CP) y 7 servomecanismos para elevar el escenario (EE). Mick seguía cantando bajo la lluvia, mientras Mick consultaba con el equipo sobre la situación. Estaba cayendo mucha lluvia y había dudas respecto a la posibilidad de mover el escenario. El Sistema estaba ejecutando los siguientes procesos:

Proceso	Demanda Máxima declarada			Recursos Alocados		
	DD	CP	EE	DD	CP	EE
Sonido	4	3	6	1	1	0
Movimiento	0	4	4	0	2	1
Luces	4	2	2	1	1	1
Frenos	1	6	3	0	0	2
Vídeo	7	3	2	2	1	0

Mick quería que el equipo de backstage le garantizara que el escenario no se iba a tratar en el medio del camino por culpa de un deadlock, dado que dejaría al grupo en una mala posición.

B2.1) Indicarle a Mick si se podrá mover el escenario (Estado seguro) o si no se podrá (Estado inseguro). Detalle en la respuesta los pasos para determinar en qué estado se encuentra el escenario.

B2.2) Describir las estrategias de recuperación de deadlock (Deadlock recovery).

Condición de aprobación del examen: Deberá tener como mínimo 3 (tres) preguntas teóricas correctamente resueltas y 1 (uno) ejercicio práctico. Recuerde que el primer tema de U2 fue "The City Of Blinding Lights". Sólo se considerará como respuesta lo que esté escrito en este final.

Apellido y Nombre:.....

A					B			CALIFICACION	Firma del Alumno
1	2	3	4	5	1	2	3		(Solo en caso de revisión de examen)

- A) **Explícitamente defina como VERDADERO o FALSO cada una de las siguientes afirmaciones FUNDAMENTANDO su respuesta en no más de 3 líneas en ambos casos.**

A.1. Un sistema operativo puede detectar el thrashing mediante el análisis de la relación entre los porcentajes de utilización de la CPU y el grado de multiprogramación del sistema.
FALSO:

EL TRASHING ES DEBIDO A LA HIPERPAGINACION O SEA EL INTENSO INTERCAMBIO ENTRE DISCO Y MEMORIA CENTRAL. NO TIENE NADA QUE VER EL USO DE LA CPU SI EL SISTEMA TIENE DMA Y SI LA TRANSFERENCIA ES REALIZADA POR PROGRAMA O POR INTERRUPCIONES ENTONCES SOLO SE OCUPA DE TRANSFERIR LOS DATOS EN UN PEQUENO INSTANTE DE TIEMPO.

A.2. En monitores con notify y broadcast, si el proceso que ejecuta un csignal no termina su ejecución dentro del monitor, entonces son necesarios dos cambios de procesos (processeswitch).

VERDADERO: OCURRE CUANDO CAMBIA EL MODO DE USUARIO A KERNEL Y DE KERNEL A USUARIO

A.3. Los hilos de un mismo proceso comparten las variables globales; el heap, la pila (stack) y los manejadores de señales (signalhandlers), pero no comparten el contador de programa (ProgramCounter).

FALSO: PUEDE TOCAR LA PILA DEL PADRE PERO NO PUEDE TOCAR LA PILA DE EJECUCIÓN.

ADEMÁS CADA HILO TIENE SU PROPIA PILA.

LOS HILOS DE UN MISMO PROCESO COMPARTEN LOS RECURSOS ASIGNADOS AL MISMO Y SU ESPACIO DE DIRECCIONES.

A.4. La llamada al sistema (systemcall) es una forma que tienen los procesos de acceder a funciones privilegiadas.

VERDADERO: UN PROCESO USUARIO PUEDE PEDIR UN SERVICIO AL KERNEL MEDIANTE UN SYSTEM CALL O UNA INSTRUCCIÓN PRIVILEGIADA.

A.5. Si el estado de un sistema es un estado seguro, entonces no puede haber espera circular.

FALSO: EL ESTADO DE UN SISTEMA ES SEGUNDO SI HAY UNA SECUENCIA DE EJECUCIÓN QUE PERMITA TERMINAR TODOS LOS PROCESOS, INDEPENDIENTEMENTE SI HAY UNA ESPERA CIRCULAR O NO, ENTONCES EN UN ESTADO SEGUNDO SE PUEDE DAR UNA ESPERA CIRCULAR MOMENTANEAMENTE SIEMPRE QUE NO SE DEEN LAS TRES CONDICIONES RESTANTES (COFFMAN).

UNA ESPERA CIRCULAR POR SI SOLA NO ES CONDICIÓN SUFFICIENTE

B) Resuelva en forma clara y detallada los siguientes ejercicios.

- B1. Se tiene un Sistema con un procesador P-IV con 1 GB de memoria y un disco rígido de aproximadamente 8 Gigabytes. Este disco contiene físicamente 256 cilindros y el brazo tarda 256 ms en recorrer todos los cilindros, numerados del 0 al 255. El sistema tiene los siguientes pedidos de lectura de cilindro encolados en el instante 0:

94, 47, 226

El sistema recibe los siguientes pedidos de lectura de disco en los siguientes momentos:

Instante (en ms)	2	3	4	4	4
Pedido de lectura del cilindro:	38	81	198	67	124

Considerando que la cabeza del disco se encuentra actualmente en el cilindro 65 y se está moviendo hacia el cilindro externo (cilindro 255), indicar cuál es el tiempo requerido para atender los pedidos si se utilizan los siguientes algoritmos:

- a) FCFS
- b) LOOK
- c) SSTF
- d) C-SCAN

A) FCFS

POSICION	INSTANTE	ORDEN PEDIDOS Cyl	Estado de la Cola	Tiempos Parciales
65	0	94	47,226	
67	2		47,226,38	
68	3		47,226,38,81	
69	4		47,226,38,81,198,67,124	
94	29	47	226,38,81,198,67,124	29
47	76	226	38,81,198,67,124	47
226	255	38	81,198,67,124	179
38	443	81	198,67,124	188
81	486	198	67,124	43
198	603	67	124	117
67	734	124	--	131
124	791	--	--	57

B) LOOKUP

POSICION	INSTANTE	ORDEN PEDIDOS Cyl	Estado de la Cola	Tiempos Parciales
65 subiendo	0	94	226,47	0
67 sub	2		226,47,38	
68 sub	3		226,81,47,38	
69 sub.	4		124,198,226,81,67,47,38	
94sub	29	124	198,226,81,67,47,38	29
124sub.	59	198	226,81,67,47,38	30
198 sub	133	226	81,67,47,38	74
226 baja	161	81	67,47,38	28
81 baja	306	67	47,38	145
67 baja	320	47	38	14
47	340	38	—	20
38	349	—	—	9

TIEMPO TOTAL: 349 ms

C) SSTF

POSICION INICIAL	INSTANTE	ORDEN PEDIDOS Cyl	Cola	Tiempos Parciales
65	0	47	94,226	0
63	2		38,94,226	
62	3		38,81,94,226	
61	4		38,67,81,94,124,198,226	
47	18	38	67,81,94,124,198,226	18
38	27	67	81,94,124,198,226	9
67	56	81	94,124,198,226	29
81	70	94	124,198,226	14
94	83	124	198,226	13
124	113	198	226	30
198	187	226	—	74
226	215	—	—	28

TIEMPO TOTAL: 215 ms

D) C-SCAN

POSICION	INSTANTE	ORDEN PEDIDOS Cyl	Estado de la Cola	Tiempos Parciales
65 subiendo	0	94	226,47	0
67 sub	2		226,47,38	
68 sub	3		226,81,47,38	
69 sub.	4		124,198,226,81,67,47,38	
94sub	29	124	198,226,81,67,47,38	29
124sub.	59	198	226,81,67,47,38	30
198 sub	133	226	81,67,47,38	74
226 sub	161	Max	81,67,47,38	28
255 invertir	190		38,47,67,81	29
0	190	38	47,67,81	0
38	228	47	67,81	38
47	237	67	81	9
67	257	81	—	20
81	271	—	—	14

TIEMPO TOTAL: 271 ms

B2. Se tiene un sistema cuyo algoritmo de reemplazo de páginas es el del reloj modificado y en el cual la siguiente tabla muestra la ocupación actual de los marcos con páginas (por ejemplo el marco 0 contiene actualmente la página 17, el marco 1 contiene la página 32, etc.).

FrameNumber	0	1	2	3	4	5	6	7
Page Number	17	32	41	5	7	13	2	20
Bit Uso	1	0	0	0	0	1	1	0
Bit Modificación	1	0	1	1	1	1	1	1

Si el puntero del reloj apunta al marco 4 y la secuencia de pedidos de página es la siguiente:

32 (lectura), 14 (escritura), 15 (escritura), 2 (lectura), 18 (lectura), 14 (escritura)

Contestar:

- ¿Qué páginas serán reemplazadas de acuerdo a la secuencia de pedidos?. Debe indicar la secuencia de reemplazo.
- Indicar la ocupación de los marcos y el valor de los bits de uso y modificación de manera similar a la presentada en el enunciado de este ejercicio al finalizar la atención de los pedidos.

SOLUCIÓN

				PEDIDOS					
Uso	Modif	Frame	Page	Write	Read	W	R	R	W
1 0	1	0	17		14	15	2	18	14
0 1 0	0 1	1	32	✓	32	32		32	
0 1	1 0	2	41		40	41		18	
→ 0	1	3	5		5	5		5	
0 1	1 0 1	4	7		14	14		14	✓
1 0	1	5	13		13	13		13	
1 0 1	1	6	2		2	2	✓	2	
0 1	1	7	20		20	15		15	
				PF	PF			PF	

EXPLICACIÓN:

1er PEDIDO: PAGINA 32 (ESCRITURA)

EL PUNTERO DEL RELOJ APUNTA AL Frame #3

SE OBSERVA QUE LA PAGINA 32 ESTA EN MEMORIA, POR LO TANTO SE PUEDE ATENDER EL PEDIDO Y SE ACTUALIZA LA TABLA:

NÚMERO DE FRAME:	1
NÚMERO DE PAGE:	32
NÚMERO DE BIT DE USO:	1
NÚMERO DE BIT DE MODIFICACIÓN:	1

2do PEDIDO: PAGINA 14 (LECTURA) SE OBSERVA QUE LA PAGINA 14 NO ESTA EN MEMORIA, POR LO TANTO SE DEBE CARGARLA PRODUCIENDOSE EL PRIMER PAGE FAULT.

EL PUNTERO DEL RELOJ APUNTA AL Frame #4 Y TIENE EL BIT DE USO EN CERO, ENTONCES SE PUEDE REEMPLAZARLA PREVIO GRABARLA EN DISCO PUES EL BIT DE MODIFICACIÓN ESTA EN UNO, O SEA QUE FUE MODIFICADA.

SE ACTUALIZA LA TABLA:

NÚMERO DE FRAME:	4
NÚMERO DE PAGE:	14
NÚMERO DE BIT DE USO:	1
NÚMERO DE BIT DE MODIFICACIÓN:	0

Y EL PUNTERO DEL RELOJ APUNTA AL Frame #5

3er PEDIDO: PAGINA 15 (ESCRITURA) SE OBSERVA QUE LA PAGINA 15 NO ESTA EN MEMORIA, POR LO TANTO SE DEBE CARGARLA PRODUCIENDOSE EL SEGUNDO PAGE FAULT.

SE PONE EL BIT DE USO DEL FRAME #5 EN CERO, EL BIT DE USO DEL FRAME #6 EN CERO Y CUANDO LLEGA AL FRAME#7 SE OBSERVA QUE SE LO PUEDE REEMPLAZAR PREVIO GRABARLÁ EN DISCO PUES EL BIT DE MODIFICACIÓN ESTA EN UNO, O SEA QUE FUE MODIFICADA Y SE CARGA LA PAGINA 15.

LA TABLA QUEDA:

NÚMERO DE FRAME:	0	1	2	3	4	5	6	7	EL PUNTERO
NÚMERO DE PAGE:	17	32	41	5	14	13	2	15	QUEDA
NÚMERO DE BIT DE USO:	1	1	0	0	1	0	0	1	APUNTANDO
NÚMERO DE BIT DE MODIFICACIÓN:	1	1	1	1	0	1	1	0	AL FRAME #0

Y EL PUNTERO DEL RELOJ APUNTA AL Frame #0

4to PEDIDO: PAGINA 2 (LECTURA) SE OBSERVA QUE LA PAGINA 2 ESTA EN MEMORIA, POR LO TANTO SE PUEDE ATENDER EL PEDIDO Y SE ACTUALIZA LA TABLA SOLO EL BIT DE USO

NÚMERO DE FRAME:	6
NÚMERO DE PAGE:	2
NÚMERO DE BIT DE USO:	1
NÚMERO DE BIT DE MODIFICACIÓN:	1

5to PEDIDO: PAGINA 18 (LECTURA) SE OBSERVA QUE LA PAGINA 18 NO ESTA EN MEMORIA, POR LO TANTO SE DEBE CARGARLA PRODUCIENDOSE EL TERCER PAGE FAULT.

SE BUSCA CON EL PUNTERO DEL RELOJ DONDE SE LA PUEDE UBICAR

LA TABLA ACTUAL ES:

NÚMERO DE FRAME:	0	1	2	3	4	5	6	7	EL PUNTERO
NÚMERO DE PAGE:	17	32	41	5	14	13	2	15	QUEDA
NÚMERO DE BIT DE USO:	1	1	0	0	1	0	1	1	APUNTANDO
NÚMERO DE BIT DE MODIFICACIÓN:	1	1	1	1	0	1	1	0	AL FRAME #0

EL #FRAME 0 TIENE EL BIT DE USO = 1 SE CAMBIA EL BIT DE USO = 0 Y SE AVANZA;
EL #FRAME 1 TIENE EL BIT DE USO = 1 SE CAMBIA EL BIT DE USO = 0 Y SE AVANZA;



EL #FRAME 2 TIENE EL BIT DE USO = 0 LO QUE INDICA QUE ESTA PAGINA SE PUEDE REEMPLAZAR POR PAGINA 18, ENTONCES SE GRABA LA PAGINA 41 POR ESTAR MODIFICADA Y SE CARGA LA PAGINA 18.
LA TABLA QUEDA:

NÚMERO DE FRAME:	0	1	2	3	4	5	6	7	EL PUNTERO QUEDA APUNTANDO AL FRAME #3
NÚMERO DE PAGE:	17	32	18	5	14	13	2	15	
NÚMERO DE BIT DE USO:	1	1	1	0	1	0	1	1	
NÚMERO DE BIT DE MODIFICACIÓN:	1	1	0	1	0	1	1	0	

6to PEDIDO: PAGINA 14 (ESCRITURA) LA PAGINA ESTA EN MEMORIA, POR LO TANTO SE PUEDE ATENDER EL PEDIDO Y SE ACTUALIZA LA TABLA:

NÚMERO DE FRAME:	4
NÚMERO DE PAGE:	14
NÚMERO DE BIT DE USO:	1
NÚMERO DE BIT DE MODIFICACIÓN:	1

FINALMENTE LA TABLA QUEDA:

NÚMERO DE FRAME:	0	1	2	3	4	5	6	7	EL PUNTERO QUEDA APUNTANDO AL FRAME #3
NÚMERO DE PAGE:	17	32	18	5	14	13	2	15	
NÚMERO DE BIT DE USO:	1	1	1	0	1	0	1	1	
NÚMERO DE BIT DE MODIFICACIÓN:	1	1	0	1	1	1	1	0	

Y SE REEMPLAZARON LAS PAGINAS:

NÚMERO DE PAGE:	7	20	41
NÚMERO DE FRAME:	4	7	2

- ✓ 3. Se dispone de un disco de 30 MB de capacidad formateado para que trabaje con un sistema de archivos tipo UNIX cuyas características se describen a continuación:

1. Tamaño de bloque 512 bytes
2. Tamaño de la dirección de bloque: 4 bytes
3. Número de i-nodos: 500
4. Campos del i-node:
 - Atributos del archivo (496bytes)
 - 2 punteros directos
 - 1 puntero indirecto simple
 - 1 puntero indirecto doble

Se pide:

- a) ¿Qué tamaño máximo podrá tener un archivo en este sistema de archivos?
- b) ¿Es posible crear enlaces físicos (Hard Links) en este sistema de archivos?
- c) ¿Qué ventajas tiene la alocación indexada respecto de la alocación contigua?

SOLUCIÓN:

¿Qué tamaño máximo podrá tener un archivo en este sistema de archivos?

$$T_{MAX} = [2 + 1 \times (512 \div 4) + 1 \times (512 \div 4)^2] = 8.455.168 \text{ bytes o } 8.06 \text{ Meby}$$

¿Es posible crear enlaces físicos (Hard Links) en este sistema de archivos?

SI, PORQUE EL I-NODO PROVEE BYTES PARA ATRIBUTOS DEL ARCHIVO, COMO SER EL CONTADOR DE LOS LINKS

¿Qué ventajas tiene la alocación indexada respecto de la alocación contigua?

PERMITE ACCEDER DE MANERA NO SECUENCIAL A LOS REGISTROS MAS RÁPIDOS: NO ES NECESARIO MANTENER EL ORDEN DE ACCESO.

Condición de Aprobación: Para aprobar este examen deberá tener como mínimo 3 (tres) preguntas teóricas bien respondidas y 2 (dos) ejercicios correctamente resueltos.

Apellido y Nombre:

A					B			CALIFICACION	Firma del Alumno
					1	2	3		(Solo en caso de revisión de examen)
1	2	3	4	5	1	2	1	1	

A) Explicitamente defina como VERDADERA o FALSA cada una de las siguientes afirmaciones, FUNDAMENTANDO su respuesta en no más de 3 renglones.

1. En un file system la organización más compleja es la denominada pila.
2. En un sistema usa un diagrama de Procesos de siete Estados que utiliza memoria virtual se elimina la necesidad de usar swapping explícito para enviar los procesos de Bloqueado a Suspendido.
3. El diseño de Microkernel impone interfaces no uniformes en los requerimientos hechos por los Procesos.
4. En un sistema de memoria virtual las políticas de LRU y RELOJ son superiores a la de FIFO. Pro son más complejas y sufren mayor overhead que la de FIFO.
5. Con DMA el procesador ejecuta más lentamente, pero para la transferencia de varias palabras, el DMA es mucho más eficiente que la Entrada -Salida Programada o Manejada por Interrupciones.

B) Resuelva en forma clara y detallada los siguientes ejercicios.

Ejercicio 1.

En un Depósito hay un montacargas para distribuir la mercadería en los dos sectores del 1er piso. El empleado de depósito carga un paquete y lo coloca en el montacargas y los empleados de los sectores A y B los retiran. Siempre deben retirar 3 paquetes el empleado del sector A y luego 1 paquete el empleado del sector B, 3 del A y 1 del B y así sucesivamente. El ascensor puede cargar hasta 20 paquetes e inicialmente está vacío.

```

D           A           B
Do while (T)Do while (T)Do while (T)
{           {           {
traer_paquete()    retirar_paq_montacargas()    retirar_paq_montacargas()
Cargar_paq_montacargas()Llevar_paquete_al_sector()    Llevar_paquete_al_sector()
}           }           }

```

1.1 Realizar la sincronización con semáforos de los tres procesos. D, A y B

D⇒| 20 | ⇒ AAABAAAB

```

D           A           B
Do while (t) Do while (t) Do while (t)
{
traer_paquete() P(a) P(b)
P(vacio)     P(lleno) P(b)
P(mutex)      P(mutex) P(b)
Cargar_paq_montacargas()    retirar_paq_montacargas()    P(lleno)
V(mutex)      V(mutex) V(mutex)
V(lleno)       V(vacio)    retirar_paq_montacargas()
}           V(b)          V(mutex)
              Llevar_paquete_al_sector() V(vacio)
                           } V(a)
                           V(a)
                           V(a)
Llevar_paquete_al_sector()
}
vacio=20 lleno=0 a=3 b=0 mutex

```

1.2 Hacer una prueba de escritorio donde el orden de ejecución sea D DDDD A B A para mostrar que la sincronización funciona. Utilice la siguiente tabla

Sem1	Sem2	Sem..	Sem..	SemN	Proceso D	Proceso A	Proceso B	Colas
vacio	lleno	mutex	a	b	D	A	B	Colas
20	0	1	3	0				
19					P(vacio)			
			0		P(mutex)			
			1		V(mutex)			
			1		V(lleno)			
18					P(vacio)			
			0		P(mutex)			
			1		V(mutex)			
			2		V(lleno)			
17					P(vacio)			
			0		P(mutex)			
			1		V(mutex)			
			3		V(lleno)			
16					P(vacio)			
			0		P(mutex)			
			1		V(mutex)			
			4		V(lleno)			
15					P(vacio)			
			0		P(mutex)			
			1		V(mutex)			
			5		V(lleno)			
				2	P(a)			
4					P(lleno)			
			0		P(mutex)			
			1		V(mutex)			
16					V(vacio)			
			1		V(b)			
			0		P(b)			
			-1		P(b)	B \rightarrow Qb (Bloqueado)		
				1	P(a)			
3					P(lleno)			
			0		P(mutex)			
			1		V(mutex)			
17					V(vacio)			
			0		V(b)	B \leftarrow Qb (Ready)		

Ejercicio 2.) Dado un programa que debe leer totalmente un archivo secuencial de 1350 registros de 128 bytes cada uno, grabados sobre un disco cuyos sectores son de 512 bytes, se pide:

¿Cuántas operaciones de E/S deben lanzarse durante la ejecución del programa?

- a) Si el archivo se encuentra en un disco de 1G que utiliza FAT 16 y no está particionado..
- b) Si el archivo se encuentra en un sistema de inodos con punteros de 16 bits y tamaño del bloque de 1 k. con 30 punteros directos , dos indirectos simples y un indirecto triple.

a) en 512 bytes entran 4 registros de 128 bytes
 $512/128 = 4$

$$1 \text{ Kb} = 1024 \text{ bytes} \quad 128 \\ 2^10 = 1024 \text{ bytes}$$

tamaño del bloque para FAT 16 $2^{16} * X = 2^{30} \Rightarrow X = 16\text{K}$
en un bloque de 16 k entran 128 registros

Cant regen 1 bloque

128 reg —— 1 bloque
1350 reg —— $X = 1350/128 = 10,546875 \Rightarrow$ se necesitan 11 bloques .

Como la FAT está en memoria no se necesita un acceso a disco para leerla \Rightarrow

solo necesito 11 accesos a disco para leer todo el archivo.

b) Entran 8 registros en un bloque de 1K $\Rightarrow 16 \text{ bits} = 2 \text{ bytes}$

Cantidad de punteros en un bloque = $1024 \text{ bytes}/2 = 512 \text{ punteros}$

Cantidad de bloques necesarios para leer todo el archivo

8 registros —— 1 bloque

1350 registros —— $X = 1350/8 = 168,75 \Rightarrow$ el archivo está contenido en 169 bloques

Suponiendo que el inodo está en memoria los 30 primeros bloques son de acceso directo \Rightarrow 30 lecturas en disco.
Faltan leer 169 - 30 = 139 bloques

La primera indirección simple tiene 512 punteros a bloque \Rightarrow con esta puedo leer el resto del archivo.

Se necesita un acceso a disco para leer el bloque de punteros de indirección simple y 139 bloques más que son los que se necesitan para finalizar la lectura del archivo.

Cantidad total de accesos a disco = $30 + 1 + 139 = 170$ accesos a disco.

Ejercicio 3. - Dada la siguiente matriz de acceso, se pide que la convierta en una lista global, lista de capacidades y lista de accesos.

	F1	F2	F3	Imp. láser	D1	D2	D3	D4
D1	read		execute			switch		
D2		read		print				switch
D3	read write		execute	admin			switch	
D4		read	read write		switch			

Condición de Aprobación: Para aprobar este examen deberá tener como mínimo 3 (tres) preguntas teóricas bien respondidas y 2 (dos) ejercicios correctamente resueltos.



Apellido y Nombre:

A					B			CALIFICACION	Firma del Alumno
					1	2	3		(Solo en caso de revisión del examen)
1	2	3	4	5	1	2	1	1	

1.- El PCB (Process Control Block) es mantenido en un almacenamiento secundario (generalmente disco) además de la localización en Memoria Central.

Rta: Verdadero (pag 127)

2.- Las primitivas send bloqueante y receive bloqueante es la forma mas natural para la programación de las tareas concurrentes.

Rta: Falso send no bloqueante y receive bloqueante es lo mas natural (pag 243)

3.- La política de limpieza (clearingpolicy) en la administración de Memoria Central (M.C.) es utilizada solo para grabar en disco, las páginas que fueron modificadas en M.C.

Rta: Verdadero (pag 370)

4- Los conflictos involucrados en un Deadlock se deben a 2 o más procesos y no a la necesidad de recursos que necesitan esos procesos

Rta: Falso se debe a los recursos que necesitan los procesos (pag 266)

5.- En el caso de los ULT (UserLevelThreads) el código para crear y destruir los threads son rutinas del Kernel

Rta: Falso son rutinas de la biblioteca de threads (pag 162)º

B) Resuelva en forma clara y detallada los siguientes ejercicios.

Ejercicio 1.

En un Depósito hay un montacargas para distribuir la mercadería en los dos sectores del 1er piso. El empleado de depósito carga un paquete y lo coloca en el montacargas y los empleados de los sectores A y B

Don Mateo tiene en su peluquería lugar para atender a tres clientes al mismo tiempo, y un salón de espera en el que se pueden acomodar 5

clientes en un sofá. En caso de que todas las ubicaciones estén llenas, se cuenta con un lugar donde los clientes pueden esperar parados. En

ningún caso la cantidad de clientes puede superar 25 en total dentro de la peluquería, ya que no se lo habilita la municipalidad. Si el cliente

puede entrar tomará un lugar en el sofá o se quedará parado (si el sofá está completamente ocupado). Cuando uno de los peluqueros está

libre, atenderá a un cliente que se encuentre esperando en el sofá, y como se liberó una posición en el sofá, si hay alguien esperando de pie

se ubicará en la posición liberada. Cuando se termina de cortar el pelo al cliente, éste debe pasar por la caja a pagar (que es administrada por

Don Mateo).

Se pide que resuelva la sincronización utilizando semáforos,

Dentro de la solución presentada deberán estar los valores iniciales de los semáforos, el protocolo de sincronización que debe usar cada

proceso para acceder a las regiones críticas, cuáles son los recursos y cuáles los procesos (todo esto a nivel teórico).

NOTA: Si ud. realiza una codificación en algún lenguaje o pseudocódigo será considerada incorrecta sin evaluar si la solución funciona o no.

Cliente Don Mateo

P(e) P(p)

entrar(); P(m);

P(a) cobrar();

sentarse(); V(e)

P(co) V(m)

V(a)

cortarse();

V(p)

V(co)

Valores iniciales de semáforos:

e=25 (semáforo para ingreso al local)
a = 5 (semáforo para lugares en el sofá)
co = 3 (semáforo para lugares de anteción)
p = 0 (semáforo para habilitar a Mateo a cobrar)
m = 1 (mutex)

Ejercicio 2.

Un programa lee una matriz de 64 columnas por 20 filas, con tipo de dato Integer (ocupa 2 bytes), de la siguiente manera (no se toman en cuenta los encabezados del programa, ni las definiciones de variables):

```
forColumnas := 0 to 63  
    forFilas := 0 to 19  
        write matrix ( Filas, Columnas )  
    next;  
next;
```

El tamaño de la memoria asignada donde debe ejecutar el proceso es de 256 bytes, divididos en páginas de 128 bytes. El

área de código del proceso ocupa una página, y está siempre cargado en memoria, quedando la otra página libre para cargar los datos desde la memoria virtual.

1) Se pide que indique la cantidad de fallos de página que generará dicho proceso (toda carga de página deberá ser considerada fallo de página, sin importar como estaba la página anteriormente). Rta: 1280 fallos

2) Efectúe los cambios necesarios al programa anterior para que disminuyan la cantidad de fallos. En caso de no haber

nigún cambio que reduzca la cantidad de fallos, expréselo claramente, y justifique. En caso de realizar cambios, recalcule el punto anterior para el nuevo programa.

```
forFilas := 0 to 19  
    forColumnas := 0 to 63  
        write matrix ( Filas, Columnas )  
    next;
```

20 fallos

Ejercicio 3.

Se posee un disco rígido con las siguientes características físicas:

2 platos

300 pistas por plato

18 sectores por pista

Formateado con Interleave de 0

La operación de escritura o lectura de un sector demora 1 ms.

Tiempo para pasar entre pistas 1 ms

El disco cuenta con un buffer de 1 sector y el tiempo necesario para enviar la información desde el buffer hasta el controlador es de 18 ms.

Tiempo punta a punta 10 ms.

En un determinado momento se encuentran encoladas las siguientes peticiones (Plato-Pista-Sector):

P1-30-12 / P0-200-9 / P1-25-0 / P1-88-15 / P1-202-2 / P0-72-4

Se pide que efectúe el orden de ejecución y los tiempos demorados en cada solicitud para el método SSTF, teniendo en cuenta que recién

termina de leer en P1-80-3, y las cabezas se encuentran en sentido ascendente.

ORDEN SSTF: POSICIÓN ACTUAL P1-80-3, P1-88-15, P0-72-4, P1-30-12, P1-25-0, P0-200-9, P1-202-2

ATENCIÓN PEDIDO DESDE P1-80-3, A P1-88-15,

DE PISTA 80 A 88 SON 8 ms, MIENTRAS TANTO SE MOVIÓ DEL SECTOR 4 AL 12, EL BUFFER NO SE DESCARGÓ POR LO QUE GIRA

UNA VUÉLTA QUE SON 18ms MAS. DEL SECTOR 12 AL 15 SON 3 ms , LECTURA DEL SECTOR 15 TARDA 1 ms MAS TIEMPO TOTAL

DE ESTA OPERACIÓN ES 30ms

ATENCIÓN PEDIDO DESDE P1-88-15, A P0-72-4,

DE PISTA 88 A 72 SON 16 ms, MIENTRAS TANTO SE MOVIÓ DEL SECTOR 16 AL 14, DEL SECTOR 14 AL 4 TARDÓ 8ms, LECTURA

DEL SECTOR 4 TARDA 1 ms, TIEMPO TOTAL DE ESTA OPERACIÓN ES 25ms

ATENCIÓN PEDIDO DESDE P0-72-4, A P1-30-12,

DE PISTA 72 A 30 SON 42 ms, MIENTRAS TANTO SE MOVIÓ DEL SECTOR 5 AL 11, DEL SECTOR 11 AL 12 TARDÓ 1ms, LECTURA

DEL SECTOR 12 TARDA 1 ms, TIEMPO TOTAL DE ESTA OPERACIÓN ES 44ms

ATENCIÓN PEDIDO DESDE P1-30-12, A P1-25-0,



DE PISTA 30 A 25 SON 5 ms, MIENTRAS TANTO SE MOVIÓ DEL SECTOR 13 AL 0, EL BUFFER NO SE DESCARGÓ POR LO QUE GIRA UNA VUELTA QUE SON 18ms LECTURA DEL SECTOR 0 TARDA 1 ms, TIEMPO TOTAL DE ESTA OPERACIÓN ES 24ms

ATENCIÓN PEDIDO DESDE P1-25-0, A P0-200-9,
DE PISTA 25 A 200 SON 175ms, MIENTRAS TANTO SE MOVIÓ DEL SECTOR 1 AL 14., DEL SECTOR 14 AL 9 TARDÓ 13ms, LECTURA
DEL SECTOR 0 TARDA 1 ms, TIEMPO TOTAL DE ESTA OPERACIÓN ES 189ms

ATENCIÓN PEDIDO DESDE P0-200-9, A P1-202-2
DE PISTA 200 A 202 SON 2ms, MIENTRAS TANTO SE MOVIÓ DEL SECTOR 10 AL 12., EL BUFFER NO SE DESCARGÓ POR LO QUE GIRA UNA VUELTA QUE SON 18ms, DEL SECTOR 12 AL 2 TARDA 8ms, LECTURA DEL SECTOR 2 TARDA 1 ms, TIEMPO TOTAL DE ESTA OPERACIÓN ES 29ms

TIEMPO TOTAL = $30 + 25 + 44 + 24 + 189 + 29 = 341$ ms, PERO ESTE TIEMPO ES SIN CONTAR QUE EL ÚLTIMO PEDIDO SIGUE EN EL BUFFER. PARA ENVIAR EL PEDIDO HABRÍA QUE ADICIONAR 18 ms CON LO CUAL EL TIEMPO TOTAL SERÍA 359 ms

Apellido y Nombre:

A					B					CALIFICACION	Firma del Alumno		
					1	2	3						
1	2	3	4	5	1	a	b	c	a	b	c	d	e

- A) Explicitamente defina como VERDADERA o FALSA cada una de las siguientes afirmaciones, FUNDAMENTANDO su respuesta en no más de 3 renglones.

1.- El control de la planificación del disco, en un sistema basado en la política de planificación de procesos por prioridades, es externo al software de control administrativo del disco

Rta: Verdadero (pag 491)

2.- La técnica "Bloqueo de Registro" en un File System no interactúa con el hardware de Memoria Virtual cuando esta es usada

Rta: Verdadero (pag 543)

3- La planificación RR (Round Robin) es particularmente efectiva en sistemas de tiempo compartido de propósito general y en sistemas de procesamiento transaccional.

Rta: Verdadero (pag 406)

4.- Lo bueno de la memoria virtual es que no se necesitan usar direcciones absolutas, para traducir las direcciones.

Rta: Es falso porque justamente las direcciones se traducen de virtual a real.

5.- Si procesos estrechamente relacionados ejecutan en paralelo (en distintos procesadores), el bloqueo por sincronización puede llegar a ser alto, menos process switching pueden llegar a ser necesarios, y la performance se incrementará.

Rta: Falso: El bloqueo por sincronización disminuirá, por el parallelismo, por el multiprocesamiento.

- B) Resuelva en forma clara y detallada los siguientes ejercicios.

Ejercicio 1. Considera que estos 4 procesos se están ejecutando concurrentemente, y que en el Sistema existen 4 semáforos inicializados en 1. Dada la siguiente secuencia de ejecución se pide determinar los posibles estados del sistema, justificando su respuesta en cada caso. Si existiera la posibilidad de que se produzca deadlock indique entre qué procesos y con qué recursos.

P1	P2	P3	P4
P(S)	P(X)	P(U)	P(T)
P(T)	P(U)	P(S)	P(X)
P(S)	P(X)
.....	V(S)	V(X)
V(T)	V(U)	V(U)	V(T)
V(S)	V(X)		

RESPUESTA:

Hay diferentes secuencias de ejecución que producen resultados distintos, basta con explicar alguna .

- a) Si ejecuta el P1 ,P2,P3;P4 en forma completa hasta bloquearse

S	T	U	X	P1	P2	P3	P4	O	Estado
1	1	1	1						
0				P(S)					
0				P(T)					
-1				P(S)				P1→Qs	BLOQ
				P(X)					
		0		P(U)					
		0		P(U)				P2→Qx	BLOQ
		-1		P(X)				P3→Qu	BLOQ
		-1		P(U)				P4→Qi	BLOQ
		-1							

El Proceso P1 y P2 padecen inanición ya que no hay suficientes recursos y no hay espera circular → no se cumplen las condiciones del Deadlock
En este caso,

- b) otra forma de ejecución es que cada uno ejecute una instrucción y pase a otro proceso.

S	T	U	X	P1	P2	P3	P4	O	Estado
1	1	1	1						
0				P(S)					

		0	P(X)			
	0		P(U)			
-1		P(T)		P1→Q1	BLOQ	
			P(U)	P2→Qu	BLOQ	
			P(S)	P3→Qs	BLOQ	
			P(X)	P4→Qx	BLOQ	

Si se produce esta secuencia y de esta forma → hay deadlock.

Ejercicio 2. Un programa debe leer una tabla de una base de datos, que tiene 20000.- (veinte mil) registros con la siguiente estructura (toda en memoria que trabaja con paginación, y el tamaño de página es de 1024 bytes).

Código (8bytes)

Descripción (60 caracteres)

Dirección (35 caracteres)

Código postal (6 caracteres)

Se pide:

- a. Indique la cantidad de páginas necesarias para cargar todos los registros en memoria.

RESPUESTA:

Tamaño de cada registro

$$TR = (8+60+35+6) \text{ bytes} = 111 \text{ bytes}$$

En cada página entran $1024/111 = 9,23$ registros.

Como no se puede empezar en una pag y terminar el reg en otra → Hay 9 reg por pagina

Como el archivo tiene 20000 registros y entran 9 por página → la cantidad de páginas necesarias es:

$$\text{CanLPag} = 20000/9 = 2222,2 \text{ pag} \rightarrow \text{necesito } 2223 \text{ pag.}$$

- b. Indique cual es el porcentaje de desperdicio (si hay), de la memoria.

Cada página tiene 9 registros.

Memoria ocupada en la pag. = $9 * 111 \text{ bytes} = 999 \text{ bytes}$

Desperdicio por página = $1024 \text{ bytes} - 999 \text{ bytes} = 25 \text{ bytes}$

Desperdicio de las 2222 pag = $2222 * 25 \text{ bytes} = 55550 \text{ bytes}$

$$\text{Can de reg en la pag 2223} = 20000 - 2222 * 9 = 20000 - 19998 = 2$$

Desperdicio en la pag.2223=1024-(can de reg que hay en la pag 2223)*111bytes

Desperdicio en la pag.2223=1024 bytes - $2 * 111 \text{ bytes} = 802 \text{ bytes}$

$$\text{DESPERDICIO TOTAL} = 55550 \text{ bytes} + 802 \text{ bytes} = 56352 \text{ bytes}$$

Memoria total ocupada = $1024 * 2223 \text{ bytes}$

Memoria total ocupada = 2276352 bytes

$$2276352 \text{ bytes} \quad 100\%$$

$$56352 \text{ bytes} \quad X = 56352 \text{ bytes} * 100\% / 2276352 \text{ bytes} = 2,47\%$$

- c. Que cambios se podrían realizar para disminuir el desperdicio al mínimo sin cambiar el tamaño de página, y recalcule los dos puntos anteriores. (Todas las respuestas deben estar justificadas):

Se podría agrandar el registro agregando un campo con blancos, de tal manera que no dejen desperdicio en la página

Ejercicio 3. Dado el siguiente código conteste las preguntas, considerando que las funciones inicializar-semáforo, P() y V() ya están codificadas y sirven para inicializar, solicitar y liberar semáforos respectivamente. Por otro lado, la función actualizar-registro guarda en un archivo el parámetro recibido y la función leer-registro consulta el valor que actualmente tenga ese archivo.

El archivo mencionado solo almacena un único valor y cada vez que la función actualiza-registro es invocada, se reemplaza con el nuevo valor recibido.

- a. ¿Cuántos procesos se generan al ejecutar este programa?

Cuando se ejecuta este programa se pueden generar 1, 2 o 3 procesos.

1 al ejecutarse el programa y los otros dos con los dos fork que hay dentro.

- b. ¿Qué ocurre si el planificador pone en la cola de listos a un proceso que no puede capturar un semáforo?

Queda bloqueado en la cola de ese semáforo hasta que otro lo habilite.

- c. ¿Qué proceso logra actualizar el archivo primero? Esto puede variar o siempre será igual en sucesivas ejecuciones?

Lo hace 1º H1 ya que es el único que tiene el semáforo habilitado, siempre va a ser así.

- d. ¿Es posible predecir observando el código cuál será el orden de ejecución de cada proceso? De ser posible indique cuál es ese orden, y en caso contrario explique el por qué.

Si se ejecuta H1,H2,P de acuerdo como se van habilitando los semáforos.

- e. En ciertas ocasiones se ha observado que las salidas de pantalla no son las esperadas. Por ejemplo, a veces aparecen mensajes idénticos en forma continua, lo cual no parece lógico en primera instancia. ¿Puede determinar cuál puede ser el motivo de estas variaciones esporádicas?

La limpieza del buffer flush (stdout).

```

int main()
{
    pid_t x, y;
    inicializar_semaforo( a, 0 );
    inicializar_semaforo( b, 1 );
    inicializar_semaforo( c, 0 );

    x=fork() ; && ejecuta el padre y genera un hijo H1
    if (x)
        y=fork() ; && esta parte la ejecuta solo el padre
        y genera un hijo H2

    if ((x) && (y))
    {
        /* esta parte la ejecuta solo el padre P*/
        for(;;)
        {
            P(a);
            Actualizar_registro (getpid ());
            V(b);
            printf("El valor registrado es: %d\n", leer_registro ());
            flush ( stdout );
        }
    }

    else
        /* esta parte la ejecuta H1*/
        for(;;)
        {
            P(b);
            Actualizar_registro (getpid ());
            V(c);
            printf("El valor registrado es: %d\n", leer_registro ());
            flush ( stdout );
        }
    }

    else
        /* esta parte la ejecuta H2*/
        for(;;)
        {
            P(c);
            Actualizar_registro (getpid ());
            V(a);
            printf("El valor registrado es: %d\n", leer_registro ());
            flush ( stdout );
        }
}

```

SIGUE EN LA OTRA COLUMNA

OBSERVACIÓN: Las primitivas P() y V() son equivalentes a Down() y Up() o Wait() y Signal()



Apellido y Nombre: Profesor:

A					B		NOTA	CORRECTOR	ALUMNO
1	2	3	4	5	1	2			(Sólo en caso de revisión de examen)

A): Responda los siguientes puntos, FUNDAMENTANDO su respuesta en no más de 4 renglones.

1. Referido al algoritmo de Round Robin para scheduling de CPU de corto plazo ¿Es posible que se presente inanición?

Rta: Escrito.

2. ¿En qué casos todos los hilos de usuario (ULTs) de un proceso comparten el estado del Proceso?

Rta: Cuando el proceso pasa al estado Suspendido o al estado Terminado. Pág. 158, 1er. párrafo.

3. Si un proceso, ejecutando código de un procedure de un monitor, ejecuta una operación WAIT sobre una variable condición, el proceso espera "dentro" del monitor en una cola de procesos bloqueados asociada a dicha variable.

F: si el proceso queda bloqueado pero como "ejecutando" dentro de un monitor, ningún otro proceso podrá acceder a la variable condición (sobre la cual el proceso original hizo el WAIT) para hacer el SIGNAL correspondiente que despierte al proceso dormido. Recordar que cualquier acceso a una variable condición definida en un monitor DEBE hacerse ejecutando código del mismo, ya que solamente un único proceso puede figurar como "ejecutando" código del monitor (por la exclusión mutua que estas construcciones proveen de forma automática).

4. Desde el punto de vista de la evitación del deadlock, si un sistema está en estado seguro, cualquier solicitud de recursos que efectúe un proceso puede satisfacerse y dejar nuevamente al sistema en estado seguro.

F: debe evaluarse, antes de otorgar los recursos disponibles solicitados, que luego de realizada la supuesta asignación, el sistema quede nuevamente en estado seguro.

5. Relacionado con los distintos niveles de RAID, RAID1 no puede implementarse para almacenar datos críticos ya que no ofrece mecanismos de redundancia.

F: ofrece mecanismo de mirroring que brinda un esquema de muy buena recuperación ante el fallo de algún miembro del array.

Ejercicio 1:

Se desea dimensionar un Sistema de Archivos UNIX para un Servidor de audio Streaming. Este Servidor enviará por la red, al cliente que se lo pide, audio comprimido (1 minuto de música por Megabyte de datos), para que el cliente lo reproduzca en tiempo real, esto es, según la descarga y sin perder el primer en un Archivo local.

Este Sistema de Archivos irá sobre un disco duro cuyas características simplificadas son las siguientes:

- 60×2^7 revoluciones por minuto.
- 2^{13} cilindros.
- 2^3 pistas (superficies) por cilindro.
- 2^{11} sectores por pista.
- 2^{-11} segundos de tiempo medio de posicionamiento del brazo, y 2^{-8} segundos en el peor caso.

a) En función de los siguientes parámetros:

TP: Logaritmo en base 2 del tiempo de posicionamiento. (Ej. Para el caso medio del disco valdría -11)

TB: Logaritmo en base 2 del tamaño de bloque. (Ej. Para un bloque de 1 Kilobyte valdría 10) Formule el tiempo de acceso a un bloque [segundos].

Formule el ancho de banda del Sistema de Archivos [Bloques / segundo]. Formule el ratio de compresión [segundos de audio / Bloque].

Formule el grado de concurrencia [segundos de audio / segundo].

El grado de concurrencia indica el número de Archivos de audio comprimido que es posible servir concurrentemente.

- b) Calcule qué tamaño de bloque aseguraría un grado de concurrencia de 60 si que se viole la restricción de tiempo real.

SOLUCIÓN

$$a) \text{Acceso} = \text{Posicionamiento} + \text{Latencia} + \text{Transferencia}$$

$$\begin{aligned}\text{Posicionamiento} &= 2^{TP} \text{ (segundos)} \\ &= 2^{-11} \text{ (segundos)} \quad (\text{casomedio}) \\ &= 2^{-8} \text{ (segundos)} \quad (\text{casopeor})\end{aligned}$$

$$\begin{aligned}\text{Latencia} &= \frac{1}{2} \text{Rotación} = 2^{-k} \text{ (segundos)} \quad (\text{casomedio}) \\ &= 1 \text{Rotación} = 2^{-7} \text{ (segundos)} \quad (\text{casopeor})\end{aligned}$$

Transferencia: Un disco transfiere datos según los sobrevuelos.

Si se transfiere una pista completa en una rotación, un bloque en:

$$\frac{2^{11+2^9}}{2^{TB} \text{ (byte)}} = 2^7 \text{ (segundos)} \quad \text{Transferencia 1 bloque} = 2^{TB-27} \text{ (segundos)}$$

s)

TA: Tiempo de Acceso a 1 bloque (segundos)

$$\begin{aligned}&= 2^{-11} + 2^{-8} + 2^{TB-27} \text{ (segundos)} \quad (\text{casomedio}) \\ &= 2^{-8} + 2^{-7} + 2^{TB-27} \text{ (segundos)} \quad (\text{casopeor})\end{aligned}$$

AB: Ancho de Banda (bloques/ segundo) = 1 / TA

RC: Ratio de Compresión (seg.audio/ bloque)

Si 1 Megabyte son 60 segundos de audio, 1 bloque serán:

$$\frac{2^{20}}{2^{TB} \text{ (bytes)}} = \frac{60 \text{ (seg.audio)}}{RC = 60 * 2^{TB-20} \text{ (seg.audio/ bloque)}}$$

GC: Grado de Concurrencia (seg.audio/segundo)

$$\begin{aligned}&= RC * AB \\ &= (60 * 2^{TB-20}) / (2^{-11} + 2^{-8} + 2^{TB-27}) \text{ (seg.audio/ bloque)} \quad (\text{casomedio}) \\ &= (60 * 2^{TB-20}) / (2^{-8} + 2^{-7} + 2^{TB-27}) \text{ (seg.audio/ bloque)} \quad (\text{casopeor})\end{aligned}$$

b) Para que no se viole la restricción de tiempo real debemos aplicar las fórmulas correspondientes al caso peor.

$$GC = 60 = (60 * 2^{TB-20}) / (2^{-8} + 2^{-7} + 2^{TB-27})$$

Despejamos 2^{TB}

$$\begin{aligned}2^{TB} &= 2^{32} * (2^8 + 2^7) / (2^{27} - 2^{20}) = \sim 12384.7 \\ 2^{13} &= 8192 \\ 2^{14} &= 16384\end{aligned}$$

Luego el tamaño de bloque adecuado será: $2^{TB} = 2^{14} = 16$ Kilobytes

c) En número de bloques del disco son:

$$2^{13} \text{ (cilindros)} * 2^3 \text{ (pistas/ cilindro)} * 2^{11} \text{ (sectores/ pista)} * 2^9 \text{ (bytes/ sector)} / 2^{TB} \text{ (bytes/ bloque)} = 2^{13+3+11+9} * 2^3 = 2^{22} \text{ (bloques)}$$

Luego para direccionar los bloques necesitaremos direcciones de 32 bits.

Y el número de direcciones que caben en un bloque de indirección serán:

$$2^{TB} \text{ (bytes/ bloque)} / 2^4 \text{ (bytes/dirección)} = 2^{10} = 1024 \text{ (direcciones/ bloque)}$$

