



Interrogación 3

Pregunta 1

- a) Describa la función del northbridge y southbridge en la arquitectura x86. **(0,75 ptos.)**

Solución: El North Bridge es el controlador encargado de comunicar a la CPU, la memoria RAM, la tarjeta gráfica y al resto de los dispositivos de I/O, conectados con el South Bridge. El South Bridge es el encargado de comunicar a todo el resto de los I/O con la memoria y la CPU, a través del North Bridge, por lo que se le conoce como controlador de I/O.

- b) Describa brevemente los pasos que se realizan cuando un programa realiza dinámicamente una petición de una nueva página de memoria. **(0,75 ptos.)**

Solución: Al ocurrir una petición, el programa le cede el control de la CPU al sistema operativo, el cual se encargará de mapear un marco físico disponible a una nueva página del programa, actualizando la tabla de páginas del programa. Si no hay marcos disponibles, se deberá hacer antes un proceso de swap-out para liberar un marco.

- c) ¿Que datos nunca serán almacenados en la memoria caché de un computador con un sistema operativo que implementa multiprogramación? **(0,75 ptos.)**

Solución: Las tablas de páginas de los programas nunca serán almacenadas en la cache de la memoria principal, ya que los datos más usados de éstas son almacenadas en la TLB.

- d) Describa el peor caso posible, desde la óptica de la cantidad de accesos, que puede ocurrir cuando un programa intenta obtener un dato almacenado en memoria, en un sistema con memoria virtual y caché. **(0,75 ptos.)**

Solución: El programa quiere acceder a una dirección virtual, el primero paso siempre será ir a buscar la entrada de la tabla de páginas correspondiente a la TLB. En caso de que la TLB no tenga la entrada, ocurre una excepción denominada **TLB miss exception**, la cual entrega el control al sistema operativo para que este vaya a buscar a la tabla de páginas del proceso actual la entrada y la copie a la TLB. Cuando el sistema operativo vaya a buscar la entrada, si esta está marcada que está en disco, ocurre un page fault y el SO debe hacer un swap in del marco correspondiente desde el swap file, reemplazar algún marco de memoria física, actualizar la tabla de páginas y copiar la entrada a la TLB. Una vez completado esto se repite el acceso a la TLB y ahora habrá un hit. Este es el **worst path** de acceso a memoria.

- e) ¿Qué almacena el vector de interrupciones en la arquitectura x86? **(0,75 ptos.)**

Solución: El vector de interrupciones almacena las direcciones de memoria de las ISR (Interrupt Service Routine), que se ejecutan al gatillarse la interrupción correspondiente.

- f) En un computador de 32 bits con 1 GB de RAM, se ejecuta un proceso que en cierto momento utiliza 1.5 GB de RAM. ¿Cómo es posible que ocurra esto sin que el proceso se caiga? Comente acerca del tiempo de ejecución de este proceso. **(0,75 ptos.)**

Solución: Esto se debe a que el espacio direccionable con 32 bits es de 4GB. Al utilizar memoria virtual, un proceso tiene acceso a todo este espacio, sin importar la cantidad de memoria física existente. Luego, si la memoria física se acaba, se utilizará espacio en disco. Por este motivo, el tiempo de ejecución del proceso será altísimo, ya que el acceso a disco es órdenes de magnitud más lento que el de memoria.

- g) Se tiene una memoria caché, con función de correspondencia de mapeo directo, de 64KB con bloques de 16 palabras de 1 byte y una memoria principal de 32MB. La CPU quiere leer el dato de la memoria principal cuya dirección es 0x0104. Asumiendo que los bloques parten desde la dirección 0000h ¿En cuál bloque de la caché debería estar este dato? **(0,75 ptos.)**

Solución: La dirección 0x0104 corresponde a la 260 en decimal y se encuentra en el bloque 16 ($\lfloor \frac{260}{16} \rfloor$) de la memoria física. Luego, como la caché es de 64KB, para las direcciones desde la 0 hasta la 65535, su bloque de la memoria física y de la caché tendrán el mismo número. De esta manera, el bloque en la caché de la dirección 0x0104 es el 16.

- h) Escriba 2 motivos por los cuales el acceso a los dispositivos de I/O se realiza comúnmente a través de interrupciones de software. **(0,75 ptos.)**

Solución:

- I. El acceso a los dispositivos de I/O es posible realizarlo generalmente sólo cuando la CPU está en modo supervisor, por lo que los programas no pueden acceder directamente a ellos. Así, el mecanismo de interrupciones de software permite este acceso, protegiendo la comunicación.
- II. Permite ahorrarse tener que implementar esa comunicación cada vez que se quiera acceder a un dispositivo de I/O.

Pregunta 2

Suponga que se tiene un computador con arquitectura x86, que tiene una memoria principal de 256 bytes, con el siguiente mapa de memoria:

Dirección	Función asociada
0-7	Exception handlers
8-15	Vectores de interrupciones de hardware
16 17-31	Vector de interrupción de escritura en disco Vectores de interrupciones de software
32-234	Memoria de uso libre
235 236-250	Registro de estado del disco duro (memory mapped) Buffer del disco duro (memory mapped)
251	Registro de controlador DMA: estado del controlador (memory mapped)
252	Registro de controlador DMA: dirección inicial de memoria a copiar (memory mapped)
253	Registro de controlador DMA: dirección inicial de buffer del disco (memory mapped)
254	Registro de controlador DMA: cantidad de palabras a transferir (memory mapped)
255	Registro de controlador DMA: comandos (memory mapped)

Se desea escribir un programa que permita escribir una secuencia de datos en un archivo del disco duro, ocupando interrupciones de software:

- a) Escriba la ISR asociada al vector de interrupción 16, que permita escribir una secuencia de palabras en el disco, **sin ocupar DMA**. Para esto, la subrutina asume que cuando fue llamada se cumple lo siguiente: **(2 ptos.)**
- La dirección de la primera palabra de memoria a ser copiado se encuentra almacenada en el registro BX.
 - La cantidad de palabras a copiar se encuentra almacenada en el registro CX.
 - La secuencia se escribirá a partir de la primera dirección del buffer y hasta que se complete la cantidad de palabras o este se llene.
- b) Escriba un programa que escriba las palabras desde la dirección 50 a la 100 de memoria al disco duro, ocupando interrupción de software para acceder a la ISR antes implementada. Considere que cuando el buffer se llena, el controlador del disco copiará estos datos al medio magnético y avisará el término de la operación en su registro de estado, escribiendo el valor 1: mientras no se complete esta operación, no se pueden transferir más datos. **(2 ptos.)**

Solución a) y b):

```
org 100h

ISR16:
    MOV SI, 0          ; SI se ocupara como indice para recorrer memoria y buffer
    MOV DX,BX          ; DX almacena como backup la direccion inicial de memoria

start:
    ; Loop que recorre arreglo a ser copiado
    CMP SI, CX         ; Revisar condicion de termino arreglo if(SI > CX) goto end
    JGE end

    MOV BX, DX         ; Recuperamos la direccion inicial de memoria
    MOV AL, [BX+SI]    ; AL = mem[BX+SI]
    MOV BX, 236        ; Se almacena en BX direccion de primera palabra del buffer
    MOV [BX + SI], AL  ; buffer[BX+SI] = AL

    INC SI             ; Incrementar iterador arreglo
    JMP start

end:
    IRET

program:
    MOV BX, 50         ; Indicamos que queremos transferir a partir de la palabra 50
    MOV CX, 15         ; Indicamos que queremos transferir 15 palabras (tamano buffer)

    INT 16             ; Llamamos a la ISR por primera vez

    CALL waitBuffer

    MOV BX, 65         ; Indicamos que queremos transferir a partir de la palabra 65
    MOV CX, 15         ; Indicamos que queremos transferir 15 palabras (tamano buffer)

    INT 16             ; Llamamos a la ISR por segunda vez

    CALL waitBuffer

    MOV BX, 80         ; Indicamos que queremos transferir a partir de la palabra 75
    MOV CX, 15         ; Indicamos que queremos transferir 15 palabras (tamano buffer)

    INT 16             ; Llamamos a la ISR por segunda vez

    CALL waitBuffer

    MOV BX, 95         ; Indicamos que queremos transferir a partir de la palabra 75
    MOV CX, 5          ; Indicamos que queremos transferir 15 palabras (tamano buffer)

    INT 16             ; Llamamos a la ISR por segunda vez

    RET

waitBuffer:
    MOV AL, [251]      ; Leemos registro de estado del controlador, mapeado a Mem[251]
    CMP AL,1           ; if(AL != 1) wait
    JNE waitbuffer
    RET
```

Puntaje a):

- (0,6 ptos) Recorrer arreglo de palabras de memoria
- (0,6 ptos) Obtener dato de memoria
- (0,8 ptos) Copiar dato de memoria a buffer del disco

Puntaje b):

- (0,8 ptos) Configurar registros para llamar a interrupcion de software (4 veces, 0,2 ptos cada vez)
 - (0,8 ptos) Llamar a interrupcion de software (4 veces, 0,2 ptos cada vez)
 - (0,4 ptos) Esperar que este disponible el buffer para volver a escribir. Si esta parte la hicieron dentro de la ISR también es válido.
- c) Modifique lo que sea necesario en la ISR y el programa anterior de manera que ahora la transferencia de realice mediante DMA. Para hacerlo, debe configurar el controlador DMA y luego iniciar el proceso de transferencia escribiendo el valor 255 en el registro de comando del controlador DMA. Puede asumir que en caso de llenarse el buffer, el controlador DMA se encargará de reconfigurarse para continuar con el proceso de transferencia, y no necesita intervención de la CPU para eso. **(2 ptos.)**

Solución c):

```
org 100h

ISR16:
    MOV [252], BX      ; Configuramos la direccion inicial de memoria
    MOV [253], 236     ; Configuramos la direccion inicial del buffer
    MOV [254], CX      ; Configuramos la cantidad de palabras
    MOV [255], 255     ; Enviamos el comando para iniciar la transferencia
    IRET

program:
    MOV BX, 50          ; Indicamos que queremos transferir a partir de la palabra 50
    MOV CX, 50          ; Indicamos que queremos transferir 50 palabras (todo)
    INT 16              ; Llamamos a la ISR una vez
    RET
```

Puntaje 2.3:

- (0,3 ptos) Configurar la direccion inicial de memoria
- (0,3 ptos) Configurar la direccion inicial del buffer
- (0,3 ptos) Configurar la cantidad de palabras
- (0,4 ptos) Enviar el comando para iniciar la transferencia
- (0,4 ptos) Configurar registros para llamar a interrupcion
- (0,3 ptos) Llamar a ISR

Pregunta 3

El siguiente programa se ejecutó en un computador con arquitectura x86 que tiene una caché de 4 bloques de 2 palabras cada uno:

Dirección	Label	
0		MOV [var2],1
1	loop1:	MOV AL,[var2]
2		MUL [var2]
3		CMP [var1],AL
4		JL end
5		INC [var2]
6		JMP loop1
7	end:	DEC [var2]
8		RET
9	var1	db ?
10	var2	db 0

Al ejecutar el programa completo se obtuvo la siguiente secuencia de accesos a memoria:

0-10-1-10-2-10-3-9-4-5-10-6-1-10-2-10-3-9-4-5-10-6-1-10-2-10-3-9-4-7-10-8

Esta secuencia de accesos a memoria generó los siguiente estados en la caché:

Dir	B0	B1	B2	B3
0	0-1			
10	0-1	10-11		
1	0-1	10-11		
10	0-1	10-11		
2	0-1	10-11	2-3	
10	0-1	10-11	2-3	
3	0-1	10-11	2-3	
9	0-1	10-11	2-3	8-9
4	0-1	10-11	2-3	4-5
5	0-1	10-11	2-3	4-5
10	0-1	10-11	2-3	4-5
6	6-7	10-11	2-3	4-5
1	6-7	10-11	0-1	4-5
10	6-7	10-11	0-1	4-5
2	6-7	10-11	0-1	2-3
10	6-7	10-11	0-1	2-3

Dir	B0	B1	B2	B3
3	6-7	10-11	0-1	2-3
9	8-9	10-11	0-1	2-3
4	8-9	10-11	4-5	2-3
5	8-9	10-11	4-5	2-3
10	8-9	10-11	4-5	2-3
6	6-7	10-11	4-5	2-3
1	0-1	10-11	4-5	2-3
10	0-1	10-11	4-5	2-3
2	0-1	10-11	4-5	2-3
10	0-1	10-11	4-5	2-3
3	0-1	10-11	4-5	2-3
9	8-9	10-11	4-5	2-3
4	8-9	10-11	4-5	2-3
7	6-7	10-11	4-5	2-3
10	6-7	10-11	4-5	2-3
8	8-9	10-11	4-5	2-3

En base a esta información, responda lo siguiente:

- a) ¿Qué valores puede tener la variable **var1** para que efectivamente se genere la secuencia de accesos detallada previamente? **(1,5 ptos)**

Solución: El rango de valores es entre 4 y 8 (ambos inclusive).

Puntaje:

- Si el rango es correcto 1,5 ptos
- Si el rango inferior está correcto y el superior tiene un error de ± 1 (i.e se respondió 7 o 9) 0,5 ptos.
- Si el rango superior está correcto y el inferior tiene un error de ± 1 (i.e se respondió 3 o 5) 0,5 ptos.

b) ¿Cuál es el hit rate? **(0,5 ptos)**

Solución: El hit rate es $\frac{17}{32}$. 0,5 ptos si está correcto, 0 en otro caso.

c) ¿Qué tipo de caché es: unified o split? **(0,5 ptos)**

Solución: La cachés es unified. 0,5 ptos si está correcto, 0 en otro caso.

d) ¿Qué función de correspondencia y algoritmo de reemplazo (si corresponde) utiliza esta caché? **(1,5 ptos)**

Solución: La función de correspondencia es fully associative (0,5 ptos) y el algoritmo de reemplazo es random (1,0 ptos). También es válido indicar en que partes el algoritmo de reemplazo obedece a que tipo, por ejemplo, LFU desde el acceso N hasta M , LRU desde X a Y .

e) ¿Es posible mejorar el desempeño de esta caché durante la ejecución de este programa, sin modificar la cantidad y tamaño de los bloques? Si es posible, explique una posible mejora que se podría realizar para lograr un mejor hit rate que el actual y demuestre que efectivamente su modificación logrará mejorar el hit rate. Si no es posible, justifique porqué. **(2,0 ptos)**

Solución: Hay dos formas de responder esta respuesta. La primera es modificar la función de correspondencia o algoritmo de reemplazo con alguno de los que se vio en clases. En este caso se obtiene 0,5 ptos por elegir una solución que efectivamente mejore el hit rate, y 1,5 por implementarla y demostrar que mejoró el hit rate. La segunda forma es inventar un algoritmo de reemplazo propio que sea específico al programa propuesto y mejore el hit rate. En este caso se obtiene 0,5 ptos por explicar el algoritmo propuesto y 1,5 ptos por implementarlo y demostrar la mejora

Pregunta 4

Asuma que una CPU tiene un espacio de direccionamiento virtual de 13 bits cuyas páginas son de 1KB. Esta máquina, sin embargo, cuenta tan sólo con 4KB de memoria RAM disponibles para marcos. Asuma que los marcos están inicialmente vacíos. En un momento comienza a ejecutarse un proceso (P1) el cual, durante su ejecución utiliza las direcciones de memoria desde la 0 hasta la 1500. Luego de esto el sistema operativo hace un cambio de contexto con lo que empieza a ejecutarse un segundo proceso (P2) el cual, durante su ejecución, utiliza las direcciones de memoria desde la 0 hasta la 500, y desde la 4500 a la 5000. Los datos que el proceso P2 almacena en éstas últimas direcciones (de la 4500 hasta la 5000) son compartidos por el proceso P3 (tanto para lectura como para escritura), el cual los accesa a través de las direcciones virtuales 2452 a la 2952. Se genera otro cambio de contexto y empieza a ejecutarse este tercer proceso (P3), el cual hace uso de las direcciones de memoria desde la 0 hasta la 1000, utilizando además datos desde la dirección 2500 a la 2600. Suponga que la política de reemplazo de páginas en los marcos es FIFO.

- a) Determine, para cada marco, de qué proceso o procesos es la información y/o datos que contiene. También indique qué paginas, de haber, se encuentran en disco. **(1 pto.)**

Solución:

Antes de P3

P1
P1
P2
P2

Después de P3

P3	P1 en disco
P1	
P2	
P2/P3	

- b) Escriba las tablas de página asociadas a estos tres procesos. **(1 pto.)**

Solución:

Proceso 1			
Pág.	Marco	Val.	Disco
0	0	0	1
1	1	1	0
2	x	0	0
3	x	0	0
4	x	0	0
5	x	0	0
6	x	0	0
7	x	0	0

Proceso 2			
Pág.	Marco	Val.	Disco
0	2	1	0
1	x	0	0
2	x	0	0
3	x	0	0
4	3	1	0
5	x	0	0
6	x	0	0
7	x	0	0

Proceso 3			
Pág.	Marco	Val.	Disco
0	0	1	0
1	x	0	0
2	3	1	0
3	x	0	0
4	x	0	0
5	x	0	0
6	x	0	0
7	x	0	0

Luego de un cambio de contexto el proceso P1 lee la dirección de memoria 500. Posterior a esto, el mismo proceso requiere escribir en la dirección de memoria 600.

- c) Determine en qué dirección real se escribe en la memoria principal al escribir este proceso en la dirección 600. **(1 pto.)**

Solución: La dirección virtual 600 corresponde a la página 0 del proceso P1. Luego del último

cambio de contexto, esta página está mapeada al marco 1 de la memoria física. Luego, la dirección física correspondiente a la dirección virtual 600 será $1024 + 600 = 1624$.

- d) Determine, para cada marco, de qué proceso o procesos es la información y/o datos que contiene. También indique qué páginas, de haber, se encuentran en disco. (1 pto.)

Solución:

P3	P1 en disco
P1	
P2	
P2/P3	

- e) Escriba las tablas de página asociadas a estos tres procesos. (1 pto.)

Solución:

Proceso 1				Proceso 2				Proceso 3			
Pág.	Marco	Val.	Disco	Pág.	Marco	Val.	Disco	Pág.	Marco	Val.	Disco
0	1	1	0	0	2	1	0	0	0	1	0
1	1	0	1	1	x	0	0	1	x	0	0
2	x	0	0	2	x	0	0	2	3	1	0
3	x	0	0	3	x	0	0	3	x	0	0
4	x	0	0	4	3	1	0	4	x	0	0
5	x	0	0	5	x	0	0	5	x	0	0
6	x	0	0	6	x	0	0	6	x	0	0
7	x	0	0	7	x	0	0	7	x	0	0

- f) Indique en qué direcciones físicas, de estar, se encuentran las direcciones virtuales: (1 pto.)

- 2500 del proceso P3

Solución: 3524

- 2000 del proceso P1

Solución: No hay dirección física, ya que se encuentra en disco.

- 4548 del proceso P2.

Solución: 3524