Espai grapa

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

05.573 15 06 13 EX

Enganxeu en aquest espai una etiqueta identificativa amb el vostre codi personal
Examen

Fitxa tècnica de l'examen

- **Comprova que el codi i el nom de l'assignatura corresponen a l'assignatura en la qual estàs matriculat.**
- **Només has d'enganxar una etiqueta d'estudiant a l'espai corresponent d'aquest full.**
- **No es poden adjuntar fulls addicionals.**
- **No es pot realitzar la prova en llapis ni en retolador gruixut.**
- **Temps total:** 2 **h.**
- **En cas que els estudiants puguin consultar algun material durant l'examen, quin o quins materials poden consultar?**
  No es pot utilitzar calculadora ni material auxiliar
- **Valor de cada pregunta:** Pregunta 1 (20%); Pregunta 2 (40%); Pregunta 3 (40%)
- **En cas que hi hagi preguntes tipus test: Descompten les respostes errònies?** NO **Quant?**

- **Indicacions específiques per a la realització d'aquest examen:**

Enunciats

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

Pregunta 1 (20%)

Pregunta sobre la pràctica.

Cal completar les instruccions marcades o afegir el codi que es demana.
Els punts suspensius indiquen que hi ha més codi però no l'heu de completar.

NOTA: En cas que el codi proposat en cada pregunta no es correspongui amb la forma que vosaltres plantejaríeu la resposta, podeu reescriure el codi o part del codi segons el vostre plantejament.

1.1: 10%

1.2: 10%

Pregunta 2 (40%)

2.1: 15%

2.2: 15%

2.3: 10%

Pregunta 3 (40%)

3.1: 20%

3.1.1: 10%

3.1.2: 10%

3.2: 20%

3.2.1: 10%

3.2.2: 10%

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

Pregunta 1

1.1

```
;;;;;
; Llegir la combinació secreta.
; Primer netejar l'espai on es llegeix la combinació amb espais en blanc,
; cridant la funció clearArea_C passant com a paràmetre row i col.
; Inicialitzar a zeros el vector secret.
; Llegir la combinació secreta, cridant la subrutina GetCode, passant com a paràmetre
; l'adreça del vector secret a través del registre ebx, posar showChar=0, per a
; indicar que GetCode no mostri els caràcters llegits i mostri *.
; ...
;
; Variables utilitzades:
; secret : vector que emmagatzema la combinació secreta
; row : fila de la pantalla on llegirem la combinació
; col : columna inicial de la pantalla on llegirem la combinació
; showChar: 0: mostrar un * per al caràcter llegit, 1: mostrar el caràcter llegit
;
; Paràmetres d'entrada :
; Cap
;
; Paràmetres de sortida:
; eax: codi de sortida: 6 si s'ha premut ESC, 0 en cas contrari
;;;;;
GetSecretCode:
push rbp
mov rbp, rsp

...

; escriure el codi per inicialitzar a 0 les 5
; posicions de tipus char del vector secret.
```

```
  GSC_iniVec:
```

```
...

mov rsp, rbp
pop rbp
ret
```

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

1.2

```
;;;;;
; ...
; Llegir la combinació secreta, cridant la subrutina GetCode, passant com a paràmetre
; l'adreça del vector secret a través del registre ebx, posar showChar=0, per a
; indicar que GetCode no mostri els caràcters llegits i mostri *.
; Si GetCode retorna un 6,s'ha premut ESC, imprimir el missatge corresponent
; cridant la funció printMessage_C i sortir.
; en cas contrari cridar a la subrutina CheckSecret.
; Si la combinació secreta conté caràcters invàlids o caràcters repetits,
; mostrar un missatge indicant-ho cridant la funció printMessage_C i
; repetir la lectura fins que la combinació sigui correcta.
; ...
;
; Variables utilitzades:
; secret : vector que emmagatzema la combinació secreta
; row : fila de la pantalla on llegirem la combinació
; col : columna inicial de la pantalla on llegirem la combinació
; showChar: 0: mostrar un * per al caràcter llegit, 1: mostrar el caràcter llegit
;
; Paràmetres d'entrada :
; Cap
;
; Paràmetres de sortida:
; eax: codi de sortida: 6 si s'ha premut ESC, 0 en cas contrari
;;;;;
GetSecretCode:
push rbp
mov rbp, rsp
...
mov ebx, _____
mov dword _____, 0
call GetCode
cmp eax, 6
je GSC_finish

call _____
cmp eax, 0
je GSC_finish
mov _____, eax
_____ printMessage_C


mov dword[col], 20
jmp GSC_readSecret
GSC_finish:
...
mov rsp, rbp
pop rbp
ret
```

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

Pregunta 2

2.1

Suposeu el següent estat inicial de la CISCA (abans de cada apartat):

- Registres: Ri = 8 * i per a i = 0, 1, ...,15.
- Memòria: M(i)=(i+16) per a i = 0, 4, 8,..., 2^{32}-4.

Completeu l'estat del computador després d'executar cada codi (indiqueu els valors dels registres en hexadecimal).

Suposeu que l'adreça simbòlica v val 400h i l'adreça simbòlica B val 800h

a)

```
ADD  R1, 10h
XOR  R10,[R2+E0h]
MOV  [B+R2],R10
```

```
R1 =
R10=
M(        ) =
Z =  , C =  , S =  , V =
```

b)

```
   XOR   [V], R10
   SAR   [V], 1h
   JMP   F
S: SUB   R1, R1
F:
```

```
M(        ) =
Z =  , C =  , S =  , V =
```

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

2.2

En la memòria d'un computador CISCA tenim emmagatzemades dos vectors A i B de 10 i 20 elements respectivament. Cada element és un nombre enter codificat en complement a 2 amb 32 bits.

Completeu els buits del fragment de codi CISCA realitzar l'equivalent a la següent sentència en C:

A[i]= A[i] + B[j]

Els vectors estan emmagatzemats en posicions consecutives de memòria, com és habitual quan es tradueix codi en C. Per exemple, els elements A[0], A[1], A[2] i M[7] es troben emmagatzemats en les adreces de memòria A, A+4, A+8 i A+28 respectivament. El mateix per al vector B.

Se sap que en R1 es troba emmagatzemat el valor de la variable “i”, i en R2 el de la “j” i que després d'executar-se el fragment de codi tots els registres han de mantenir els valors originals.

```
PUSH R2
PUSH _____
MUL  _____ , _____
SAL  R2, _____
____ R3, _____
____ _____ , R3
POP  _____
____ _____
```

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

2.3

Donat el següent fragment de codi d'un programa en llenguatge assemblador del CISCA:

```
MOV   R2, [V]
ADD   R1, 6
SUB   R1, R2
```

Traduïu-ho a llenguatge màquina i expresseu-ho en la següent taula. Suposeu que la primera instrucció del codi es troba a partir de l'adreça 00FF0010h (que és el valor del PC abans de començar l'execució del fragment de codi). Suposeu també que l'adreça simbòlica V val 00001D80h. En la taula de resultats useu una fila per codificar cada instrucció. Si suposem que la instrucció comença en l'adreça @, el valor Bk de cadascun dels bytes de la instrucció amb adreces @+k per a k=0, 1,... s'ha d'indicar en la taula en hexadecimal en la columna corresponent (recordeu que els camps que codifiquen un desplaçament en 2 bytes o un immediat o una adreça en 4 bytes ho fan en format little endian, això cal tenir-ho en compte escrivint els bytes de menor pes, d'adreça més petita, a l'esquerra i els de major pes, adreça major, a la dreta). Completeu també la columna @ que indica per a cada fila l'adreça de memòria del byte B0 de la instrucció que es codifica en aquesta fila de la taula.

A continuació us donem com a ajuda les taules de codis:

Taula de codis d'instrucció

B0	Instrucció
10h	MOV
20h	ADD
21h	SUB

Taula de modes d'adreçament (Bk<7..4>)

Camp mode Bk<7..4>	mode
0h	Immediat
1h	Registre
2h	Memòria
3h	Indirecte
4h	Relatiu
5h	Indexat
6h	Relatiu a PC

Taula de modes d'adreçament (Bk<3..0>)

Camp mode Bk<3..0>	Significat
Nº registre	Si el mode ha d'especificar un registre
0	No s'especifica registre.

		Bk per a k=0..10										
@	**Assemblador**	0	1	2	3	4	5	6	7	8	9	10
	`MOV R2, [V]`											
	`ADD R1, 6`											
	`SUB R1, R2`											

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

Pregunta 3

3.1

Tenim un sistema de memòria en el que tots els accessos es fan a paraula (no ens importa quina és la mida d'una paraula). Suposarem que l'espai d'adreces de memòria es descompon en blocs de 8 paraules. Cada bloc comença a una adreça múltiple de 8. Així, el bloc 0 conté les adreces 0, 1, 2, 3, 4, 5, 6 i 7; el bloc 1, les adreces 8, 9, 10, 11, 12, 13, 14 i 15, i el bloc N les adreces 8*N, 8*N+1, 8*N+2, 8*N+3, 8*N+4, 8*N+5, 8*N+6 i 8*N+7.

Suposem que el sistema també disposa d'una memòria cau de 4 línies (on cada línia té la mida d'un bloc, és a dir, 8 paraules). Aquestes línies s'identifiquen com a línies 0, 1, 2 i 3. Quan es fa referència a una adreça de memòria principal, si aquesta adreça no es troba a la memòria cau, es porta tot el bloc corresponent des de la memòria principal a una línia de la memòria cau (així si fem referència a l'adreça 2 de memòria principal portarem el bloc format per les paraules 0, 1, 2, 3, 4, 5, 6 i 7).

3.1.1 Memòria Cau d'Accés Directe

Suposem que el sistema fa servir una **política d'assignació directa**, de manera que cada bloc de la memòria principal només es pot portar a una línia determinada de la memòria cau.

L'execució d'un programa genera la següent llista de lectures a memòria:

24, 16, 8, 25, 2, 48, 22, 53, 2, 12, 21, 32, 22, 23, 13

3.1.1.a) La següent taula mostra l'estat inicial de la cau, que conté les primeres 32 paraules de la memòria (organitzades en 4 blocs). Completar la taula per a mostrar l'evolució de la cau durant l'execució del programa. Per a cada fallada en la cau cal omplir una nova columna indicant quina referència a memòria ha provocat la fallada i el canvi que es produeix en l'estat de la memòria cau (la línia que es modifica).

	Estat Inicial	Fallada:
Línia 0	0, 1, 2, 3, 4, 5, 6, 7	
Línia 1	8, 9, 10, 11, 12, 13, 14, 15	
Línia 2	16, 17, 18, 19, 20, 21, 22, 23	
Línia 3	24, 25, 26, 27, 28, 29, 30, 31	

	Fallada:	Fallada:
Línia 0		
Línia 1		
Línia 2		
Línia 3		

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

	Fallada:	Fallada:
Línia 0		
Línia 1		
Línia 2		
Línia 3		

3.1.1.b) Quina és la taxa de fallades (T_f) ?

3.1.1.c) Suposem que el temps d'accés a la memòria cau, o temps d'accés en cas d'encert (t_e), és de 2 ns i el temps total d'accés en cas de fallada (t_f) és de 20 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitja d'accés a memòria (t_m) ?

3.1.2 Memòria Cau d'Accés Completament Associatiu

Ara suposem que el mateix sistema fa servir una política d'emplaçament completament associativa, de manera que qualsevol bloc de la memòria principal es pot portar a qualsevol bloc de la memòria cau.
Si trobem que la cau ja està plena, es fa servir un algorisme de reemplaçament LRU, de manera que traurem de la memòria cau aquell bloc que fa més temps que no es referència.

Considerem la mateixa llista de lectures a memòria:

24, 16, 8, 25, 2, 48, 22, 53, 2, 12, 21, 32, 22, 23, 13

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

3.1.2.a) La següent taula mostra l'estat inicial de la cau, que conté les primeres 32 paraules de la memòria (organitzades en 4 blocs). Completar la taula.

	Estat Inicial	Fallada:
Línia 0	0, 1, 2, 3, 4, 5, 6, 7	
Línia 1	8, 9, 10, 11, 12, 13, 14, 15	
Línia 2	16, 17, 18, 19, 20, 21, 22, 23	
Línia 3	24, 25, 26, 27, 28, 29, 30, 31	

	Fallada:	Fallada:
Línia 0		
Línia 1		
Línia 2		
Línia 3		

	Fallada:	Fallada:
Línia 0		
Línia 1		
Línia 2		
Línia 3		

3.1.2.b) Quina és la taxa de fallades (T_f) ?

3.1.2.c) Suposem que el temps d'accés a la memòria cau, o temps d'accés en cas d'encert (t_e), és de 2 ns i el temps total d'accés en cas de fallada (t_f) és de 20 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitja d'accés a memòria (t_m) ?

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

3.2

Es vol realitzar la següent comunicació de dades entre la memòria d'un computador i un port USB, que tenen les següents característiques:

- La CPU funciona amb un rellotge de 1GHz de freqüència i executa 1 instrucció per cada cicle de rellotge
- Adreces dels **registres de dades** i **d'estat** del controlador d'E/S: 0B0h i 0B4h
- El bit del **registre d'estat** que indica que el controlador del port d'E/S està disponible és el bit 3, o el quart bit menys significatiu (quan val 1 indica que està disponible)
- Transferència de **escriptura** des de memòria al port d'E/S
- Transferència de N_{dades}=400.000 dades, és a dir, 400.000 × 4 Bytes = 1.600.000 Bytes
- Adreça inicial de memòria on resideixen les dades: 20000000h
- La velocitat de transferència el port és de 10.000 Bytes per segon

3.2.1 E/S programada

Completar el següent codi realitzat amb el repertori CISCA que realitza la transferència descrita abans mitjançant la tècnica d'E/S programada.

```
1.          MOV     R3, _________
2.          MOV     R2, 20000000h
3. Bucle:   IN      R0, [0B4h]          ; llegir 4 bytes
4.          _____   R0, 00001000b
5.          _____   Bucle
6.          MOV     R0,[_______]        ; llegir 4 bytes
7.          ADD     R2, _______
8.          _____   0B0h, R0            ; escriure 4 bytes
9.          SUB     R3, _______
10.         _____   Bucle
```

Quin és el percentatge de temps que dedica la CPU a la tasca d'Entrada/Sortida?

Examen 2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

3.2.2 E/S per Interrupcions

Completar el següent codi CISCA que és una rutina de servei a les interrupcions (RSI) per a transferir a través del dispositiu d'E/S anterior, el mateix nombre de dades que abans amb E/S programada, però ara mitjançant la tècnica de E/S per interrupcions. Suposeu:

- Es fa servir una variable global que es representa amb l'etiqueta **Dir,** i que al principi del programa conté l'adreça inicial de memòria on resideixen les dades a transferir

```
1.      _______
2.      PUSH    _______
3.      PUSH    R1
4.      _______ R1, [ Dir ]
5.      MOV     R0, _______
6.      OUT     _______, R0     ; escriure 4 bytes
7.      _______ R1, 4
8.      MOV     _______, R1
9.      POP     _______
10.     POP     R0
11.     STI
12.     _______
```

Quin és el percentatge de temps que dedica la CPU a la tasca d'Entrada/Sortida?