Espai grapa

Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

⊂05.573ℜ12ℜ01ℜ13ℜEΞ[∈
05.573 12 01 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:** S'indica a l'enunciat de la prova.
- **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 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

Pregunta 1 (20%)

Només s'han de completar les instruccions marcades.
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 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

Pregunta 1

1.1

Completa el codi de la subrutina següent:

```
; Inicialitza un vector de N posicions, posant cada posició a 0.
;
; Variables modificades:
; vector[]
;
; Paràmetres d'entrada:
; ebx: adreça del vector
; esi: posicions del vector
;
ini_vec:
push rbp
mov rbp, rsp
push rdi     ;guardem a la pila els registres que modifiquem
             ;per deixar tots els registres com estaven abans de sortir.

______ edi, ______ ;índex per accedir al vector

ini_vec_for: ;Inicialitzem el vector

mov _________ [ _____________ ], 0
;ESPECIFICAR: Modificador [ accés al vector ]
;Modificadors vàlids: BYTE, WORD, DWORD, QWORD

inc _______ ;incrementa index del vector
_____ edi, _______
_____ ini_vec_for

ini_vec_end:
pop rdi         ;restaurar l'estat dels registres.
mov rsp, rbp
pop rbp
ret
```

Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

1.2

Completa el codi de la subrutina següent:

```
; Llegir 1, 2, 4 o 8 bytes del codi màquina emmagatzemat al vector
; machine_code_Mem[] a partir de la posició indicada per la variable PC.
; El codi llegit es guardar en el vector IR. (IR[i]=machine_code_Mem[PC+i])
; Actualitzar l'índex per accedir al vector machine_code_Mem[] (PC)
; amb l'adreça del darrer byte llegit + 1.
;
; Variables llegides:
; machine_code_Mem[] //Vector de caràcters on tenim el codi.
;
; Variables modificades:
; IR[] //bytes de codi llegit a machine_code[] a partir de l'adreça indicada.
;
; Paràmetres d'entrada:
; edx (dl) //PC:Índex per accedir al codi.
; ecx (cl) //Bytes que s'han de llegir.(1,2,4,8)
; Paràmetres de sortida:
; edx (dl) //PC:Índex per accedir al codi actualitzat.
;
read_MemCode:
...

mov ebx, IR ;adreça del vector
mov esi, 9 ;mida del vector
_____ ini_vec ;Inicialitza el vector IR de 9 posicions amb zeros.
;declarat en el codi C com: char IR[9];

mov eax, _____ ;carregar l'adreça del vector
;amb el codi màquina
mov ebx, IR
mov esi, edx ;índex per accedir a machine_code_Mem
mov edi, 0 ;índex per accedir a IR
read_MemCode_loop: ;llegim de la memòria de codi
cmp edi, ecx
jge read_MemCode_Ok
mov _____ , _____ ;guardar 1, 2, 4 o 8 bytes de codi
mov _____ , _____ ;al vector IR.
inc esi
inc edi
_____ read_MemCode_loop

read_MemCode_Ok:

...

ret
```

Examen 1

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 A val 400h i l'adreça simbòlica B val 800h

a)

```
MOV   R1, [A+100h]
XOR   R1, 10h
SUB   R1, [B]
MOV   [A], 10h
```

R1 =

M( ) =

Z = , C = , S = , V =

b)

```
   MOV   R1, 1
   MOV   R2, 3
   SAR   R1, R2
   JE    F
   MOV   R1, FFFFFFFFh
F:
```

R1 =

R2 =

Z = , C = , S = , V =

Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

2.2

En la memòria d'un computador CISCA tenim emmagatzemada una matriu de 100 x 100 elements (100 files per 100 columnes) a partir de l'adreça simbòlica M. Cada element és un nombre enter codificat en complement a 2 amb 32 bits. També tenim un vector A de 100 elements també d'enters de 32 bits.

Completeu els buits del fragment de codi CISCA per inicialitzar la posició M[i][i] amb el valor de A[i], la qual cosa en C s'especificaria amb la sentència:

M[i][i]= A[i];

La matriu està emmagatzemada per files en posicions consecutives de memòria, com és habitual quan es tradueix codi en C. Per exemple, els elements M[0][0], M[0][1], M[1][0] i M[7][40] es troben emmagatzemats en les adreces de memòria M, M+4, M+400 i M+2960 respectivament.

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

```
PUSH  R1
PUSH  ____
PUSH  R3

MOV   R3, R1
MUL   ____ , ____
ADD   ____ , R1
_____ R3, 2

_____ R1 , 2
MOV   R2 , ______
MOV   ______ , R2

____  ____
POP   ____
POP   ____
```

Examen 1

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 R1,[V+R2]
        JMP stop
fi:     INC R1
stop:
```

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
40h	JMP
10h	MOV
24h	INC

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

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.

Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

		Bk per a k=0..10										
@	Assemblador	0	1	2	3	4	5	6	7	8	9	10
	MOV R1,[V+R2]											
	JMP STOP											
	INC R1											

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. Una fórmula per calcular l'identificador numèric del bloc és la següent:

Bloc = adreça de memòria (adreça a paraula) DIV 8 (mida del bloc en paraules)

Suposem que el sistema també disposa d'una memòria cau de 4 línies (on cada línia té la mida d'un bloc, es 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. En aquest cas, l'identificador del bloc determina la línia específica on es pot guardar fent servir la següent fórmula (similar a la fórmula per determinar el bloc):

Línia = identificador de bloc MOD 4 (mida de la cau en línies)

Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

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

0, 16, 8, 2, 24, 48, 22, 53, 23, 50

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 (afegint les columnes que siguin necessàries) 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		

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

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

Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

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 5 ns i el temps total d'accés en cas de fallada (t_f) és de 25 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.

Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

Considerem la mateixa llista de lectures a memòria:

0, 16, 8, 2, 24, 48, 22, 53, 23, 50

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		

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

Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

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 5 ns i el temps total d'accés en cas de fallada (t_f) és de 25 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitja d'accés a memòria (t_m) ?

Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

3.2

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

- Adreces dels **registres de dades** i **d'estat** del controlador d'E/S: 0A0h i 0A4h
- El bit del **registre d'estat** que indica que el controlador del port d'E/S està disponible és el bit 4, o el cinquè bit menys significatiu (quan val 1 indica que està disponible)
- Transferència de **escriptura** des de memòria al port d'E/S
- Transferència de $\boldsymbol{N}_{\text{dades}}$=400.000 dades, es a dir, 400.000 × 4 Bytes = 1.600.000 Bytes
- Adreça inicial de memòria on resideixen les dades: 20000000h

Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

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, 400000
2.          MOV     _______,  20000000h
3. Bucle:   IN      R0, [0A4h]              ; llegir 4 bytes
4.          _____R0, 00010000b
5.          _____Bucle
6.          MOV     R0, [_____]             ; llegir 4 bytes
7.          ADD     R2, ______
8.          _____0A0h , R0      ; escriure 4 bytes
9.          SUB     R3, ______
10.         _____Bucle
```

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