



Estructura de Computadores - Examen Final - 11/01/2020 - CAT

Estructura de computadores (Universitat Oberta de Catalunya)

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	11/01/2020	18:30

05.573R11R01R20REE3E
05.573 11 01 20 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 matriculada.
- Només has d'enganxar una etiqueta d'estudiant a l'espai corresponent d'aquest full.
- No es poden adjuntar fulls addicionals, ni realitzar l'examen en llapis o retolador gruixut.
- Temps total: **2 hores** Valor de cada pregunta: **S'indica a l'enunciat**
- En cas que els estudiants puguin consultar algun material durant l'examen, quins són?
CAP En cas de poder fer servir calculadora, de quin tipus? **CAP**
- Si hi ha preguntes tipus test: Descompten les respostes errònies? **NO** Quant?
- Indicacions específiques per a la realització d'aquest examen:

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	11/01/2020	18:30

Enunciats

No es pot utilitzar calculadora. Cal saber interpretar un valor en binari, decimal o hexadecimal per a realitzar l'operació que es demani. I el resultat s'ha d'expressar en el format corresponent.

Valoració de les preguntes de l'examen

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 (35%)

2.1 : 10%

2.2 : 15%

2.3 : 10%

Pregunta 3 (35%)

3.1: 15%

3.1.1 : 10%

3.1.2 : 5%

3.2: 20%

3.2.1 : 10%

3.2.2 : 5%

3.2.3 : 5%

Pregunta 4 (10%)

4.1 : 5%

4.2 : 5%

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	11/01/2020	18:30

Pregunta 1

1.1 Pràctica – 1a Part

Escriure un fragment de codi assemblador de la subrutina posCurScreenP1 que calcula el valor rowScreen i colScreen. (No s'ha d'escriure el codi de tota la subrutina)

```

;;;;
; Posiciona el cursor a la pantalla dins del tauler, en funció
; de l'index de la matriu (indexMat), posició del cursor dins del tauler.
; Per a calcular la posició del cursor a la pantalla utilitzar
; aquestes fórmules:
; rowScreen=((indexMat/10)*2)+7
; colScreen=((indexMat%10)*4)+7
; Per a posicionar el cursor cridar a la subrutina gotoxyP1.
; Variables globals utilitzades:
; indexMat : índex per a accedir a les matrius mines i marks des d'assemblador.
; rowScreen: Fila de la pantalla on posicionem el cursor.
; colScreen: Columna de la pantalla on posicionem el cursor.
;;;;
posCurScreenP1:
    push rbp
    mov rbp, rsp
    push rax
    push rdx
    push rdi
    push rsi
    mov rdx, 0
    mov rax, QWORD[indexMat]
    mov rsi, 10
    div rsi          ;RAX = RDX:RAX / EDI, RDX = RDX:RAX mod EDI
                    ;RAX=(indexMat/10) RDX=(indexMat/10)
    mov rdi, rax
    shl rdi, 1       ; (indexMat/10)*2
    add rdi, 7        ; (indexMat/10)*2)+7

    mov rsi, rdx
    shl rsi, 2        ; ((indexMat%10)*4)
    add rsi, 7        ; ((indexMat%10)+7

    mov QWORD[rowScreen], rdi
    mov QWORD[colScreen], rsi
    call gotoxyP1
posCurScreenP1_End:
    pop rsi
    pop rdi
    pop rdx
    pop rax
    mov rsp, rbp
    pop rbp
    ret

```

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	11/01/2020	18:30

1.2 Pràctica – 2a part

Completar el codi de la subrutina `showMinesP2` per a que mostri `numMines` a la part inferior del tauler de joc. (Només completar els espais marcats, no es poden afegir o modificar altres instruccions).

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Aquesta subrutina es dona feta. NO LA PODEU MODIFICAR.
; Situar el cursor en una fila i una columna de la pantalla
; en funció de la fila (edi) i de la columna (esi) rebuts com
; a paràmetre cridant a la funció gotoxyP2_C.
; Variables globals utilitzades: Cap
; Paràmetres d'entrada : rdi(edi): (rowScreen) Fila
;                          rsi(es): (colScreen) Columna;
; Paràmetres de sortida: Cap
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
gotoxyP2:

;
; Converteix el valor del nombre de mines que queden per marcar
; (numMines) que es rep com a paràmetre (entre 0 i 99) en dos caràcters
; ASCII. S'ha de dividir el valor (nMines) entre 10, el quocient
; representarà les desenes i el residu les unitats, i després s'han de
; convertir a ASCII sumant 48, caràcter '0'.
; Mostra els dígit (caràcter ASCII) de les desenes a la fila 27,
; columna 24 de la pantalla i les unitats a la fila 27, columna 26.
; Per a posicionar el cursor s'ha de cridar a la subrutina gotoxyP2 i
; per a mostrar els caràcters a la subrutina printchP2 implementant
; correctament el pas de paràmetres
;
; Variables globals utilitzades: Cap.
; Paràmetres d'entrada: rdi(edi):(nMines)Mines que queden per marcar.
; Paràmetres de sortida: Cap
;
;
showMinesP2:
    push rbp
    mov rbp, rsp
    ...
    ; hem dividit numMines entre 10, el quocient representarà les desenes
    ; i el tenim en el registre al i el residu representarà les unitats
    ; i el tenim en el registre dl.
    add al,'0'          ;convertim els valors a caràcters ASCII
    add dl,'0'          ;
    ;Posicionar el cursor i mostrar dígit
    mov __rdi__, 27      ;rowScreen = 27;
    mov __rsi__, 25      ;colScreen = 24
    call __gotoxyP2__
    __push__ rdi
    mov dil, al
    call printchP2
    pop __rdi__
    mov rsi, 26          ;colScreen = 26
    call gotoxyP2
    mov __dil__, dl
    call printchP2

```

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	11/01/2020	18:30

```
showMinesP2_End:
...
mov rsp, rbp
pop rbp
ret
```

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	11/01/2020	18:30

Pregunta 2

2.1

L'estat inicial del computador CISCA just abans de començar l'execució de cada fragment de codi (a cada apartat) és el següent:

R0= 100h R1= 200h R2= 300h R3= 400h	M(00000200h)=FFFFFF600h M(00000300h)=00000600h M(00000400h)=0000FFFFh M(00000500h)=60000000h	Z = 0, C = 0, S = 0, V = 0
--	---	----------------------------

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

Suposeu que l'adreça simbòlica A val 100h.

a)

```
MOV R2, FFFFFFF0h
ADD R2, [A+R1]
```

R2:= FFFFFFF0h
R2:= FFFFFFF0h+[00000300h]=
FFFFFFF0h+00000600h = 000005F0h

Z= 0 , S= 0 , C= 1 , V= 0

b)

```
MOV R6, [A+100h]
ADD R6, R3
NOT R6
```

R6 = [200]= FFFFF600h
R6 = FFFFFA00h
R6 = 000005FFh

Z=0, S=1, C=0, V=0,

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	11/01/2020	18:30

2.2

Suposem que tenim el vector V de 7 elements de 32 bits. Completar la traducció del programa en assembler CISCA perquè executi l'algorisme d'alt nivell mostrat. (Hem deixat 8 espais per omplir)

```

A = 12;
for (i = 6; i >= 0; i--) {
    V[i] = A * i;
    A = A - 2;
}

```

El registre R1 representa la variable **A** i R2 representa la variable **i**

```

MOV [A], 12
MOV R1, [A]
MOV R2, 6
MOV R3, 24
COMP: CMP R2, 0
      JL END
      MOV [V+R3], R1
      MUL [V+R3], R2
      SUB R1, 2
      SUB R2, 1
      SUB R3, 4
      JMP COMP
END:  MOV [A], R1
      MOV [I], R2

```


Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	11/01/2020	18:30

2.3

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

```
PLUS: SUB R0,[A+R4]
      MOV R3,[100h]
      JE PLUS
```

Traduïu-ho a llenguatge màquina i expresseu-ho en la següent taula. Supposeu que la primera instrucció del codi s'assembla a partir de l'adreça 00044A00h (que és el valor del PC abans de començar l'execució del fragment de codi). Supposeu que l'adreça simbòlica A val 00004000h. En la següent taula 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
26h	SUB
41h	JE

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	
00044A00h	SUB R0,[A+R4]	26	10	54	00	40	00	00					
00044A07h	MOV R3,[100h]	10	13	20	00	01	00	00					
00044A0Eh	JE PLUS	41	60	EE	FF								
00044A12h													

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	11/01/2020	18:30

Pregunta 3

3.1. Memòria cau

Memòria cau d'assignació directa

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 en una adreça múltiple de 8. Així, el bloc 0 conté les adreces 0, 1, 2, 3, 4, 5, 6, 7, el bloc 1, les adreces 8, 9, 10, 11, 12, 13, 14, 15, i el bloc N les adreces $8*N$, $8*N+1$, $8*N+2$, $8*N+3$, $8*N+4$, $8*N+5$, $8*N+6$, $8*N+7$.

Suposem que el sistema també disposa d'una memòria cau de 4 línies (on cada línia té la mida d'un bloc, 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, 7).

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:

17, 18, 19, 12, 23, 28, 8, 9, 50, 51, 21, 63, 64, 27, 52, 22, 28, 58, 20, 21

3.1.1

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 accés cal omplir una columna indicant si es tracta d'un encert o una fallada.

Si és un encert escriurem E en la línia corresponent davant de les adreces del bloc, si és una fallada escriurem F i en aquest cas s'indicarà el nou bloc que es porta a la memòria cau en la línia que li correspongui, expressat de la forma b:e ($a_0 - a_7$) on b: número de bloc, e: etiqueta i ($a_0 - a_7$) són les adreces del bloc, on a_0 és la primera adreça del bloc i a_7 és la vuitena (darrera) adreça del bloc.

Línia	Estat Inicial	17	18	19	12	23
0	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)
1	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	E 1:0 (8 - 15)	1:0 (8 - 15)
2	2:0 (16 - 23)	E 2:0 (16 - 23)	E 2:0 (16 - 23)	E 2:0 (16 - 23)	2:0 (16 - 23)	E 2:0 (16 - 23)
3	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	11/01/2020	18:30

Línia	28	8	9	50	51	21
0	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)
1	1:0 (8 - 15)	E 1:0 (8 - 15)	E 1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)
2	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	F 6:1 (48 - 55)	E 6:1 (48 - 55)	F 2:0 (16 - 23)
3	E 3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)

Línia	63	64	27	52	22	28
0	0:0 (0 - 7)	F 8:2 (64 - 71)	8:2 (64 - 71)	8:2 (64 - 71)	8:2 (64 - 71)	8:2 (64 - 71)
1	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)
2	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	F 6:1 (48 - 55)	F 2:0 (16 - 23)	2:0 (16 - 23)
3	F 7:1 (56 - 63)	7:1 (56 - 63)	F 3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	E 3:0 (24 - 31)

Línia	58	20	21			
0	8:2 (64 - 71)	8:2 (64 - 71)	8:2 (64 - 71)			
1	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)			
2	2:0 (16 - 23)	E 2:0 (16 - 23)	E 2:0 (16 - 23)			
3	F 7:1 (56 - 63)	7:1 (56 - 63)	7:1 (56 - 63)			

3.1.2 a)

Quina és la taxa d'encerts (T_e) ?

$$T_e = 12 \text{ encerts} / 20 \text{ accessos} = 0,6$$

3.1.2 b)

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 d'encerts obtinguda a la pregunta anterior, quin és el temps mitjà d'accés a memòria (t_m) ?

$$t_m = T_e \times t_e + (1 - T_e) \times t_f = 0,6 \times 5 \text{ ns} + 0,4 \times 25 \text{ ns} = 3 \text{ ns} + 10 \text{ ns} = 13 \text{ ns}$$

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	11/01/2020	18:30

3.2 Sistema d'E/S

E/S programada

Es vol analitzar el rendiment de la comunicació de dades entre la memòria d'un processador i un port USB, utilitzant E/S programada, amb les següents característiques:

- Velocitat de transferència del dispositiu d'E/S $v_{\text{transf}} = 40 \text{ MBytes/s} = 40000 \text{ Kbytes/s}$
- Temps de latència mitjà del dispositiu $t_{\text{latència}} = 0$
- Adreces dels **registres d'estat** i **dades** del controlador d'E/S: 0B00h i 0B04h, respectivament
- El bit del **registre d'estat** que indica que el controlador del port d'E/S està disponible és el bit 4, o sigui el cinqué bit menys significatiu (quan val 1 indica que està disponible)
- Processador amb una freqüència de rellotge de 2 GHz, el temps de cicle $t_{\text{cicle}} = 0,5 \text{ ns}$.
- El processador pot executar 1 instrucció per cicle de rellotge
- Transferència de **escriptura** des de memòria al port d'E/S
- Transferència de $N_{\text{dades}} = 3200000$ dades
- La mida d'una dada és $m_{\text{dada}} = 4 \text{ bytes}$
- Adreça inicial de memòria on resideixen les dades: 80000000h

3.2.1

El següent codi realitzat amb el joc d'instruccions CISCA realitza la transferència descrita abans mitjançant la tècnica d'E/S programada. Completeu el codi.

```

1.      MOV R3, 3200000
2.      MOV R2, 80000000h
3. Bucle: IN      R0, [0B00h] ; llegir 4 bytes
4.      AND R0, 00010000b
5.      JE Bucle
6.      MOV R0, [R2]          ; llegir 4 bytes
7.      ADD R2, 4
8.      OUT [0B04h], R0      ; escriure 4 bytes
9.      SUB R3, 1
10.     JNE Bucle

```

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	11/01/2020	18:30

3.2.2

Quant temps dura la transferència del bloc de dades $t_{\text{transf_bloc}}$?

$$\begin{aligned}
 t_{\text{transf_bloc}} &= t_{\text{latència}} + (N_{\text{dades}} * t_{\text{transf_dada}}) \\
 t_{\text{latència}} &= 0 \\
 N_{\text{dades}} &= 3200000 \\
 t_{\text{transf_dada}} &= m_{\text{dada}} / v_{\text{transf}} = 4 \text{ Bytes} / 40000 \text{ Kbytes/s} = 0,0001 \text{ ms} = 0,1 \text{ us} \\
 t_{\text{transf_bloc}} &= 0 + (3200000 * 0,0001 \text{ ms}) = 320 \text{ ms}
 \end{aligned}$$

3.2.3

Si volguéssim fer servir el mateix processador i el mateix programa però amb un dispositiu d'E/S més ràpid, quina és la màxima taxa o velocitat de transferència del nou dispositiu que es podria suportar sense que el dispositiu s'hagués d'esperar?

$$\begin{aligned}
 t_{\text{cicle}} &= 0,5 \text{ ns (nanosegon)} \\
 t_{\text{instr}} &= 0,5 \text{ ns}
 \end{aligned}$$

El mínim nombre d'instruccions que ha d'executar el programa per a cada dada transferida són les 8 instruccions: 3, 4, 5, 6, 7, 8, 9 i 10. Executar les 8 instruccions requereix $8 * t_{\text{instr}} = 8 * 0,5 \text{ ns} = 4 \text{ ns}$

Per tant, el temps mínim per a transferir una dada és: 4 ns

Es poden transferir 4 bytes cada 4 ns, es a dir: $4 / 4 * 10^{-9} = 1000 \text{ Mbyte/s} = 1 \text{ Gbytes/s}$

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	11/01/2020	18:30

Pregunta 4

4.1

Describeu clarament el bit de arrossegament C i el bit de desbordament V.

El bit de carry C és el bit de resultat que s'activa si en una operació de suma o resta de naturals ens portem una. En operacions amb nombres amb signe, el bit equivalent és el overflow V, que indica que el resultat no pot ser representat en el format que estem utilitzant (complement a 2).

4.2

4.2.1

Describeu breument els 4 algorismes de reemplaçament més comunament utilitzats en la memòria cau.

- 1) FIFO (first in first out). Per a escollir la línia s'utilitza una cua, de manera que la línia que fa més temps que està emmagatzemada en la memòria cau serà la reemplaçada. Aquest algorisme pot reduir el rendiment de la memòria cau perquè la línia que fa més temps que es troba emmagatzemada a la memòria cau no ha de ser necessàriament la que s'utilitzi menys.
Es pot implementar fàcilment utilitzant tècniques de buffers circulars (o roundrobin): cada cop que s'ha de substituir una línia s'utilitza la línia del buffer següent, i quan s'arriba a l'última es torna a començar pel principi.
- 2) LFU (least frequently used). En aquest algorisme s'escull la línia que hem utilitzat menys vegades.
Es pot implementar afegint un comptador del nombre d'accessos a cada línia de la memòria cau.
- 3) LRU (least recently used). Aquest algorisme escull la línia que fa més temps que no s'utilitza. És l'algorisme més eficient, però el més difícil d'implementar, especialment si s'ha d'escollir entre moltes línies. S'utilitza habitualment en memòries cau associatives per conjunts, amb conjunts petits de 2 o 4 línies.
Per a memòries cau 2-associatives, es pot implementar afegint un bit en cada línia; quan es fa referència a una de les dues línies, aquest bit es posa a 1 i l'altre es posa a 0, per a indicar quina de les dues línies ha estat la darrera que s'ha utilitzat.
- 4) Aleatori. Els algorismes anteriors es basen en factors sobre la utilització de les línies de la cau, en canvi aquest algorisme escull la línia que s'ha de reemplaçar a l'atzar. Aquest algorisme és molt simple i s'ha demostrat que té un rendiment només lleugerament inferior als algorismes que tenen en compte factors d'utilització de les línies.

4.2.2

En què consisteix la tècnica d'interrupcions Vectoritzades?

S'anomena vectorització la tècnica en què el processador identifica el mòdul d'E/S mitjançant la informació que envia el mateix mòdul d'E/S. Quan per a tractar les interrupcions utilitzem aquesta tècnica per a identificar qui fa la petició diem que tenim les interrupcions vectoritzades.

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	11/01/2020	18:30