EX SOL - CAT - Estructura de Computadors

Estructura de computadors (Universitat Oberta de Catalunya)

Examen 2020/21-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/1/2021	18:30

Fitxa tècnica de l'examen

- Comprova que el codi i el nom de l'assignatura corresponen a l'assignatura de què t'has matriculat.
- Temps total: **2 hores** Valor de cada pregunta:
- Es pot consultar cap material durant l'examen? **NO** Quins materials estan permesos?
- Es pot fer servir calculadora? **NO** 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:

Enunciat: L'enunciat de l'examen estarà en format PDF.
A l'ordinador des d'on fareu l'examen cal tindre instal·lat algun programari per a poder llegir documents en format PDF. Per exemple, es pot utilitzar el programari gratuït Adobe Acrobat Reader DC, però podeu utilitzar qualsevol altre programari.

Identificació de l'estudiant: No és necessari identificar-se, d'aquesta forma es garanteix que l'examen serà tractat de forma anònima.

Respostes:
S'ha d'identificar cada resposta dins l'examen. És ***obligatori indicar el número de pregunta i l'apartat***, opcionalment també es pot afegir tot o part de l'enunciat si això us ajuda en la resolució de la pregunta.
Si no s'identifica correctament a quina pregunta fa referència la resposta no s'avaluarà.

En cas de ser necessari aplicar un procediment per resoldre alguna pregunta, mostreu clarament i argumenteu el procediment aplicat, no només el resultat. En cas de dubte, si no es poden resoldre pels mecanismes establerts o per manca de temps, feu els supòsits que considereu oportuns i argumenteu-los.

Elaboració document a lliurar:
Utilitzar qualsevol editor de text per crear el document amb les respostes, sempre que després us permeti exportar el document a format PDF per fer el lliurament.

Lliurament: És obligatori lliurar les respostes de l'examen en un únic document ***en format PDF***.
No s'acceptaran altres formats.
És responsabilitat de l'estudiant que la informació que contingui el document PDF que es lliuri reflecteixi correctament les respostes donades a l'examen. Recomanem que obriu el fitxer PDF generat i reviseu atentament les respostes per evitar que s'hagi pogut perdre, canviar o modificar alguna informació al generar el document en format PDF.
El lliurament es pot fer tantes vegades com es vulgui, es corregirà el darrer lliurament que es faci dins l'horari especificat per realitzar l'examen.

COMPROMÍS D'AUTORESPONSABILITAT: aquest examen s'ha de resoldre de forma individual sota la vostra responsabilitat i seguint les indicacions de la fitxa tècnica (sense utilitzar cap material, ni calculadora).
En cas que no sigui així, l'examen s'avaluarà amb un zero. Per altra banda, i sempre a criteri dels Estudis, l'incompliment d'aquest compromís pot suposar l'obertura d'un expedient disciplinari amb possibles sancions.

Examen 2020/21-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/1/2021	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 : 15%
1.2 : 5%

Pregunta 2 (40%)

2.1 : 10%
2.2 : 15%
2.3 : 15%

Pregunta 3 (40%)

3.1: 20%
3.1.1 : 10%
3.1.2 : 10%

3.2: 20%
3.2.1 : 10%
3.2.2 : 5%
3.2.3 : 5%

Examen 2020/21-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/1/2021	18:30

Pregunta 1

1.1 Pràctica – 1a Part

Modifiqueu la subrutina copyMatrixP1 perquè compti els espais en blanc que té la matriu Tiles. Si no hi ha 1 únic espai posar la variable state a '5'. (No s'ha d'escriure el codi de tota la subrutina, només cal modificar el codi per fer el què es demana).

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Copiar la matriu (tilesIni), a la matriu (tiles).
; Recorrer tota la matriu per files d'esquerra a dreta i de dalt a baix.
; Per recorrer la matriu en assemblador l'índex va de 0 (posició [0][0])
; a 8 (posició [2][2]) amb increments de 1 perquè les dades son de
; tipus char(BYTE) 1 byte.
;
; Variables globals utilitzades:
; (tilesIni): Matriu amb els nombres inicials del joc
; (tiles)   : Matriu on guardem els nombres del joc.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
copyMatrixP1:
   push rbp
   mov  rbp, rsp
   push rax
   push rbx                        ;i
   push rcx                        ;j
   push rdx                        ;índex matriu

   mov r10, 0
   mov rdx, 0
   mov ebx, 0                      ;i=0
   copyMatrixP1_Bucle_Row:
   cmp ebx, DimMatrix              ;i<DimMatrix
   jge copyMatrixP1_Row_End
     mov ecx, 0                    ;j=0
     copyMatrixP1_Bucle_Col:
     cmp ecx, DimMatrix            ;j<DimMatrix
     jge copyMatrixP1_Col_End
       mov al, BYTE[tilesIni+rdx] ;al = tilesIni[i][j]
       mov BYTE[tiles+rdx], al    ;t[i][j] = al
       inc rdx
       inc ecx                    ;i++
       cmp al, ' '
       jne copyMatrixP1_Bucle_Col
       inc r10
     jmp copyMatrixP1_Bucle_Col
     copyMatrixP1_Col_End:         ;}
     inc  ebx                      ;j++
   jmp copyMatrixP1_Bucle_Row
   copyMatrixP1_Row_End:
   cmp r10, 1
   jne copyMatrixP1_Row_EndIf
     mov BYTE[State], '5'
   copyMatrixP1_Row_EndIf:
   pop rdx
   pop rcx
   pop rbx
   pop rax

   mov rsp, rbp
   pop rbp
   ret
```

Examen 2020/21-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/1/2021	18:30

1.2 Pràctica – 2a part

Feu els canvis necessaris al codi assemblador d'aquesta subrutina considerant que les variables spacePos i newSpacePos s'ha declarat de tipus long(8 bytes), no es poden afegir instruccions, només modificar les instruccions que sigui necessari.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Moure la fitxa en la direcció indicada pel caracter (charac),
; ('i':dalt, 'k':baix, 'j':esquerra o 'l':dreta) rebut com a
; paràmetre a la posició on hi ha l'espai dins la matriu indicada per
; la variable (spacePos) rebuda com a paràmetre, controlant el casos
; en els que no es pot fer el moviment.
; fila    (i = spacePos / DimMatrix) i columna (j = spacePos % DimMatrix)
; Si la casella on hi ha l'espai està als extrems de la matriu no es
; podrà fer el moviment des d'aquell costat.
; Si es pot fer el moviment:
; Moure la fitxa a la posició on està l'espai de la matriu (tiles)
; i posar l'espai a la posició on hi havia la fitxa moguda.
; Per recorrer la matriu en assemblador l'índex va de 0 (posició [0][0])
; a 9 (posició [2][2]) amb increments de 1 perquè les dades son de
; tipus char(BYTE) 1 byte.
; No s'ha de mostrar la matriu amb els canvis, es fa a UpdateBoardP2().
;
; Variables globals utilitzades: (tiles): Matriu on guardem els nombres del joc.
; Paràmetres d'entrada : rdi (dil): (charac)  : Caracter llegit de teclar.
;                        rsi (esi): (spacePos): Posició de l'espai a la matriu (tiles).
; Paràmetres de sortida: rax (eax): (newspacePos): Nova posició de l'espai a la matriu (tiles).
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
moveTileP2:
   push rbp
   mov  rbp, rsp
   ...
   mov r8, rsi            ;spacePos
   mov rax, rsi
   mov edx, 0
   mov ebx, DimMatrix
   div ebx
   mov r10d, eax          ;int i = spacePos / DimMatrix;
   mov r11d, edx          ;int j = spacePos % DimMatrix;
   mov r9, r8             ;int newPosSpace = spacePos;
                                    ;switch(charac){
   moveTileP2_Switchi:
   cmp dil, 'i'                     ;case 'i':
   jne moveTileP2_Switchk
      cmp r10d, DimMatrix-1         ;if (i < (DimMatrix-1)) {
      jge moveTileP2_SwitchEnd
         mov r9d, r8d
         add r9d, DimMatrix         ;newspacePos = spacePos+DimMatrix;
         mov dl, _BYTE[tiles+r9]_   ;tiles[i+1][j];
         mov _BYTE[tiles+r8]_, dl   ;tiles[i][j]= tiles[i+1][j];
         mov _BYTE[tiles+r9]_, ' '  ;tiles[i+1][j] = ' ';
         jmp moveTileP2_SwitchEnd   ;break
   moveTileP2_Switchk:
   ...
   moveTileP2_Switchj:
   ...
   moveTileP2_Switchl:
   ...
   moveTileP2_SwitchEnd:
   mov rax, r9                      ;return newspacePos;
   moveTileP2_End:
   ...
   mov rsp, rbp
   pop rbp
   ret
```

Examen 2020/21-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/1/2021	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:

R2 = 00000050h R4 = 00000010h R8 = 00000200h	M(00000200h) = 0810077Eh M(00000050h) = F0000810h M(00000250h) = 00000810h	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 200h.

a)

```
XOR R4,R4
MOV R4,[A]
ADD R4,[A]
```

R4 = 0
R4 = 0810077Eh
R4 = 0810077Eh + 0810077Eh
= 10200EFCh

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

b)

```
SUB R8, [A]
NOT R8
```

R8 = 200h – 0810077Eh = F7EFFA82h
R8 = 0810057Dh

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

Examen 2020/21-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/1/2021	18:30

2.2

Suposem que tenim el vector V de 10 elements de 32 bits. Completar la traducció del programa en assemblador CISCA perquè executi l'algorisme d'alt nivell mostrat. (Hem deixat 7 espais per omplir)

```
i= 9;
while i!=0 do {
     if (V [i] <= 10) V[i] = V[i]-1;
     else V[i] = 0;
     i= i-1;
};
```

```
      MOV  R1, 36
COND: CMP  R1 , 0
      JE   END
      CMP  [V+R1], 10
      JG   ELSE
      SUB  [V+R1], 1
      JMP  NEXT
ELSE: MOV  [V+R1], 0
NEXT: SUB  R1, 4
      JMP  COND
END:
```

Examen 2020/21-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/1/2021	18:30

2.3

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

```
            CMP R11, [B]
            JNE   Label1
            INC   R11
Label1:     SUB   4, [B+R11]
```

Traduïu-ho a llenguatge màquina i expresseu-ho en la següent taula. Suposeu que la primera instrucció del codi s'assembla a partir de l'adreça 00023C00h (que és el valor del PC abans de començar l'execució del fragment de codi). Suposeu que l'adreça simbòlica B val 00000080h. 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
26h	CMP
42h	JNE
24h	INC
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
Num. 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
00023C00h	CMP R11, [B]	26	1B	20	80	00	00	00				
00023C07h	JNE Label1	42	60	02	00							
00023C0Bh	INC R11	24	1B									
00023C0Dh	SUB 4, [B+R11]	21	00	04	00	00	00	5B	80	00	00	00
00023C18h												

Examen 2020/21-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/1/2021	18:30

Pregunta 3

3.1. Memòria cau

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'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.**

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

12, 4, 14, 28, 5, 20, 6, 44, 28, 7, 8, 30, 45, 55, 10, 43, 0, 56, 46, 1

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 s'indicarà el nou bloc que es porta a la memòria cau en la línia que li correspongui, expressat de la forma b (a_0 - a_7) on b:número de bloc, 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	12	4	14	28	5
0	0 (0 - 7)	0 (0 - 7)	E 0 (0 - 7)	0 (0 - 7)	0 (0 - 7)	E 0 (0 - 7)
1	1 (8 - 15)	E 1 (8 - 15)	1 (8 - 15)	E 1 (8 - 15)	1 (8 - 15)	1 (8 - 15)
2	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)
3	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	E 3 (24 - 31)	3 (24 - 31)

Línia	20	6	44	28	7	8
0	0 (0 - 7)	E 0 (0 - 7)	0 (0 - 7)	0 (0 - 7)	E 0 (0 - 7)	0 (0 - 7)
1	1 (8 - 15)	1 (8 - 15)	F 5 (40 – 47)	5 (40 – 47)	5 (40 – 47)	5 (40 – 47)
2	E 2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	F 1 (8 - 15)
3	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	E 3 (24 - 31)	3 (24 - 31)	3 (24 - 31)

Examen 2020/21-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/1/2021	18:30

Línia	30	45	55	10	43	0
0	0 (0 - 7)	0 (0 - 7)	F 6 (48 - 55)	6 (48 - 55)	6 (48 - 55)	6 (48 - 55)
1	F 5 (40 – 47)	E 5 (40 – 47)	5 (40 – 47)	5 (40 – 47)	E 5 (40 – 47)	5 (40 – 47)
2	1 (8 - 15)	1 (8 - 15)	1 (8 - 15)	E 1 (8 - 15)	1 (8 - 15)	1 (8 - 15)
3	E 3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	F 0 (0 - 7)

Línia	56	46	1			
0	F 7 (56 - 63)	7 (56 - 63)	7 (56 - 63)			
1	5 (40 – 47)	E 5 (40 – 47)	5 (40 – 47)			
2	1 (8 - 15)	1 (8 - 15)	1 (8 - 15)			
3	0 (0 - 7)	0 (0 - 7)	E 0 (0 - 7)			

3.1.2 a)

Quina és la taxa d'encerts (T_e) ?

T_e = 15 encerts / 20 accessos = 0,75

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 2 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 mitja d'accés a memòria (t_m) ?

$t_m = T_e \times t_e + (1-T_e) \times t_f$ = 0,75 * 2ns + 0,25 * 25ns = 1,5ns + 6,25ns = 7,75ns

Examen 2020/21-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/1/2021	18:30

3.2 Sistema d'E/S

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

- Velocitat de transferència del dispositiu d'E/S v_{transf} = 4 MBytes/s = 4000 Kbytes/s
- Temps de latència mitjà del dispositiu $t_{latència}$ = 0
- Adreces dels **registres d'estat** i **dades** del controlador d'E/S: 1F00h i 1F04h
- El bit del **registre d'estat** que indica que el controlador del port d'E/S està disponible és el bit 5, o el sisè bit menys significatiu (quan val 1 indica que està disponible)
- Processador amb una freqüència de rellotge de 4 GHz, el temps de cicle t_{cicle} = 0,25 ns. El processador pot executar una instrucció en 8 cicles de rellotge
- Transferència de **lectura** des del port d'E/S cap a memòria
- Transferència de N_{dades}= 800.000 dades
- La mida d'una dada és m_{dada} = 4 bytes
- El temps per atendre la interrupció (t_{rec_int}) és de 20 cicles de rellotge

3.2.1 Completeu 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, mitjançant la tècnica de E/S per interrupcions.

Es fa servir una variable global que es representa amb l'etiqueta **Addr,** i que al principi del programa conté l'adreça inicial de memòria on emmagatzemar les dades rebudes.

```
1.   _CLI_
2.   PUSH R0
3.   PUSH R1
4.   IN   R0,_[1F04h]_
5.   MOV  R1, [Addr]
6.   MOV  _[R1]_,R0
7.   ADD  R1,_4_
8.   MOV  _[Addr],R1
9.   POP  R1
10.  POP  R0
11.  STI
12.  _IRET_
```

3.2.2 Quant temps dura la transferència del bloc de dades t_{transf_bloc}?

El temps d'un cicle, t_{cicle} = 0,25 ns (nanosegons)

Temps per atendre la interrupció, t_{rec_int}: 20 cicles * 0,25 ns = 5 ns

Temps d'execució de una instrucció, t_{instr}: t_{cicle} * 8 = 2 ns

Temps d'execució RSI, t_{rsi}: N_{rsi} * t_{instr} = 12 instr. * 2 ns = 24 ns

Temps consumit per CPU en cada interrupció, t_{transf_dada}:

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/1/2021	18:30

$t_{transf\_dada} = t_{rec\_int} + t_{rsi} = 5 + 24 = 29$ ns

Nombre d'interrupcions produïdes (nombre total de dades, N_{dades}): 800000 interrupcions

Temps consumit en total en TOTES les interrupcions:

$t_{transf\_bloc} = t_{transf\_dada} * N_{dades} = 29$ ns * 800000 interrupcions = 23.200.000 ns = 23,2 ms (milisegons)

3.2.3 Quin és el percentatge d'ocupació del processador? Percentatge que representa el temps de transferència del bloc t_{transf_bloc} respecte al temps de transferència del bloc per part del perifèric t_{bloc}

$t_{bloc} = t_{latència} + (N_{dades} * t_{dada})$
$t_{latència} = 0$
$N_{dades} = 800000$
$t_{dada} = m_{dada} / v_{transf} = 4 / 4000$ Kbytes/s = 0,001 ms
$t_{bloc} = 0 + (800000 * 0{,}001)$ ms = 800 ms
% ocupació = $(t_{transf\_bloc} * 100 / t_{bloc}) = (23{,}2 * 100) / 800 = 2{,}90$ %

Examen 2020/21-1

Assignatura	**Codi**	**Data**	**Hora inici**
Estructura de computadors	05.573	16/1/2021	18:30