

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	18/01/2012	15:30

05.573 18 01 12 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 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	18/01/2012	15:30

Pregunta 1 (20%: 2.1: 10%, 2.2: 10%)

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.

Pregunta 1.1 (10%)

Completa el codi corresponent a la subrutina `m_mul_escalar_mr` (producte escalar, `escalar*mr`).

```
m_mul_escalar_mr:
    push rbp
    mov rbp, rsp

    mov esi, 0
m_mul_escalar_bucle:
    mov  eax, [mr+esi*4]
    imul eax, [escalar]
    mov  [mr+esi*4], eax
    inc esi
    cmp  esi, 8
    jle  m_mul_escalar_bucle

m_mul_escalar_end:
    mov rsp, rbp
    pop rbp
    ret
```

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	18/01/2012	15:30

Pregunta 1.2 (10%)

Completa el codi corresponent a la subrutina de càlcul del determinant de la matriu `mr` per la regla de Sarrus. Es demana completar només la part de codi corresponent a estendre la matriu 3x3 `mr` sobre la matriu auxiliar `mAux` 5x3.

```
m_det_mr:
    push rbp
    mov rbp, rsp
    mov esi, 0          ;Inicialitzem l'índex de la matriu mAux 5x3
    mov edi, 0          ;Inicialitzem l'índex de la matriu mr 3x3

m_det_ini:
    mov eax, [mr+edi*4]
    mov [mAux+esi*4], eax
    inc esi
    cmp esi, 14         ;Hem acabat la còpia sobre mAux
    jg m_det_next
    inc edi              ;Incrementem l'índex de mr
    cmp edi, 8          ;Finalitza el recorregut de mr
    jle m_det_ini
    mov edi, 0
    jmp m_det_ini
m_det_next:
...
```

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	18/01/2012	15:30

Pregunta 2 (40%: 2.1: 15%, 2.2: 15%, 2.3: 10%)

Apartat 2.1 (15%)

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

R0 = 00000000h R1 = 00001000h R2 = 00002000h	M(00001000h) = 00001000h M(00002000h) = 00001000h M(00003000h) = 00000000h	Z = 0 C = 0 S = 0 V = 0
--	--	----------------------------------

Quin serà l'estat del computador després d'executar cada fragment de codi? (només modificacions, excloent el PC)

a)	<pre> XOR R1, R2 MOV R2, 00001000h SUB R1, [00001000h+R2] </pre>
	R1 = 00002000h R2 = 00001000h Z = 0, S = 0 C = 0, V = 0

b)	<pre> MOV R2, 00001000h SUB R1, 60h XOR R2, R0 MOV R0, FFFFFFFFh </pre>
	R2 = 00001000h R1 = 0000FA0h R0 = FFFFFFFFh Z = 0, S = 0 C = 0, V = 0

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	18/01/2012	15:30

Apartat 2.2 (15%)

En la memòria d'un computador CISCA tenim emmagatzemada una matriu de 10 x 100 elements (10 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.

Completeu els buits del fragment de codi CISCA per posar a 0 l'element $M[i][j]$, la qual cosa en C s'especificaria amb la sentència: $M[i][j] = 0$;

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, 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. El codi és correcte però no és tot l'eficient que podria ser.

```
PUSH R1
PUSH R2
MUL R1, 400
MUL R2, 4
ADD R1, R2
MOV [M+R1], 0
POP R2
POP R1
```

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	18/01/2012	15:30

Apartat 2.3 (10%)

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

```

                JMP   Label_2
Label_1:        SUB   [B+R1], 4
Label_2:        MOV   [R3+4], R4

```

Traduïu-lo a llenguatge màquina i expresseu-lo en la següent taula. Supposeu que la primera instrucció del codi s'assembla a partir de l'adreça 00FF0014h (que és el valor del PC abans de començar l'execució del fragment de codi). Supposeu també que l'adreça simbòlica B val 0000F000h. 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

Instrucció	B0
JMP	40h
MOV	10h
SUB	21h

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

Campo modo Bk<7..4>	Modo
0h	Inmediato
1h	Registro
2h	Memoria
3h	Indirecto
4h	Relativo
5h	Indexado
6h	A PC

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	18/01/2012	15:30

Taula de codis 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.

Taula de resposta:

		Bk para k=0..10											
@	Ensamblador	0	1	2	3	4	5	6	7	8	9	10	
00FF0014	JMP Label_2	40	00	25	00	FF	00						
00FF001A	SUB [B+R1], 4	21	51	00	F0	00	00	00	04	00	00	00	
00FF0025	MOV [R3+4], R4	10	43	04	00	14							
00FF002A													

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	18/01/2012	15:30

Pregunta 3 (40%: 3.1: 15%, 3.2: 15%, 3.3: 10%)

Apartat 3.1 (15%)

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 4 paraules. Cada bloc comença a una adreça múltiple de 4. Així, el bloc 0 conté les adreces 0, 1, 2 i 3, el bloc 1, les adreces 4, 5, 6 i 7, i el bloc N les adreces $4*N$, $4*N+1$, $4*N+2$ i $4*N+3$. Una fórmula per calcular l'identificador numèric del bloc és la següent:

Bloc = adreça de memòria (adreça a paraula) DIV 4 (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, 4 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 i 3).

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	18/01/2012	15:30

Apartat 3.1.1

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):

$$\text{Línia} = \text{identificador de bloc} \bmod 4 \text{ (mida de la cau en línies)}$$

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

1, 2, 3, 4, 5, 10, 11, 12, 30, 31, 32, 1, 2, 3, 4, 5, 10, 11, 12, 30

3.1.1.a)

La següent taula mostra l'estat inicial de la cau, que conté les primeres 16 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:30	Fallada:32	Fallada:1	Fallada:12
Línia 0	0, 1, 2, 3		32, 33, 34, 35	0, 1, 2, 3	
Línia 1	4, 5, 6, 7				
Línia 2	8, 9, 10, 11				
Línia 3	12, 13, 14, 15	28, 29, 30, 31			12, 13, 14, 15

	Fallada: 30	Fallada:	Fallada:	Fallada:	Fallada:
Línia 0					
Línia 1					
Línia 2					
Línia 3	28, 29, 30, 31				

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

$$T_f = 5 \text{ fallades} / 20 \text{ accessos} = 0,25$$

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 4 ns i el temps total d'accés en cas de fallada (t_f) és de 24 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitjà d'accés a memòria (t_m) ?

$$t_m = T_f \times t_f + (1 - T_f) \times t_e = 0,25 \times 24 \text{ ns} + 0,75 \times 4 \text{ ns} = 6 \text{ ns} + 3 \text{ ns} = 9 \text{ ns}$$

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	18/01/2012	15:30

Apartat 3.1.2

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 s'ha referenciat. Considerem la mateixa llista de lectures a memòria:

1, 2, 3, 4, 5, 10, 11, 12, 30, 31, 32, 1, 2, 3, 4, 5, 10, 11, 12, 30

3.1.2.a)

La següent taula mostra l'estat inicial de la cau, que conté les primeres 16 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 omplir una nova columna indicant quina referència a memòria ha provocat la fallada i el nou estat de la cau.

	Estat Inicial	Fallada: 30	Fallada: 32	Fallada: 1	Fallada: 4
Línia 0	0, 1, 2, 3	28, 29, 30, 31			
Línia 1	4, 5, 6, 7		32, 33, 34, 35		
Línia 2	8, 9, 10, 11			0, 1, 2, 3	
Línia 3	12, 13, 14, 15				4, 5, 6, 7

	Fallada: 10	Fallada: 12	Fallada: 30	Fallada:	Fallada:
Línia 0	8, 9, 10, 11				
Línia 1		12, 13, 14, 15			
Línia 2			28, 29, 30, 31		
Línia 3					

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

$$T_f = 7 \text{ fallades} / 20 \text{ accessos} = 0,35$$

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 4 ns i el temps total d'accés en cas de fallada (t_f) és de 24 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitja d'accés a memòria (t_m) ?

$$t_m = T_f \times t_f + (1 - T_f) \times t_e = 0,35 \times 24 \text{ ns} + 0,65 \times 4 \text{ ns} = 8,4 \text{ ns} + 2,6 \text{ ns} = 11 \text{ ns}$$

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	18/01/2012	15:30

Apartat 3.2 (15%)

3.2.1 (5%)

El següent codi CISCA transfereix dades entre memòria i disc fent servir E/S per interrupcions.

```

1.      CLI
2.      PUSH  R0
3.      PUSH  R1
4.      IN    R0, [10]
5.      IN    R0, [18]
6.      MOV   [R1], R0
7.      POP   R1
8.      POP   R0
9.      STI
  
```

Indicar si la transferència és de sortida al disc o d'entrada des del disc, i els dos errors que es troben al codi

Transferència d'**entrada**.

- Falta **IRET** al final del codi
- Falta obtenir a R1 un valor **vàlid com a adreça destí** de les dades llegides de disc, i actualitzar-la

No és necessàriament un error el fet d'executar **dues instruccions IN seguides** sobre el mateix registre.

3.2.2 (10%)

Una rutina de servei a les interrupcions (RSI) feta amb el repertori CISCA serveix per a transferir una dada de 4 bytes des de disc a memòria. La transferència amb el disc d'un bloc de 8000 dades (cada dada de 4 bytes) triga 32 mil·lisegons. El temps consumit per la CPU des de que es detecta la interrupció fins que s'acaba d'executar la RSI és sempre de 8 ns.

Quin percentatge del temps total de transferència dedica la CPU?

Temps consumit per la CPU per interrupció: 8 ns
 Nombre total d'interrupcions produïdes: 8.000 dades / 1 dada per interrupció = 8.000
 Temps consumit en total: 8 ns per interrupció x 8.000 interrupcions = **64 us**

Percentatge de temps consumit per la CPU durant el temps de la transferència:

$64 \text{ us} / 32.000 \text{ us} = 0,002 \rightarrow \text{0,2 \%}$

Examen 2011/12-1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	18/01/2012	15:30

Apartat 3.3 (10%)

El sistema d'E/S per DMA d'un computador té les següents característiques:

- El **temps de cessió** i el **temps de recuperació** del bus ($t_{\text{cessió}} + t_{\text{recup}}$) són 4 ns.
- El **temps de la transferència** pel bus (t_{mem}) és de 2 ns.
- S'envien pel bus 2.000 dades de 32 bits cadascuna

Calcular el temps total d'ocupació del bus per part del controlador de disc/DMA per a dur a terme la transferència (no es fan servir ràfegues).

Temps ocupació Bus, $t_{\text{transf_dada}}$:

$$t_{\text{cessió}} + t_{\text{mem}} + t_{\text{recup}} = 4 + 2 = 6 \text{ ns}$$

Nombre de peticions del Bus, N_{dades} :

2.000

Temps total d'ocupació del Bus $t_{\text{transf_bloc}}$:

$$t_{\text{transf_dada}} \times N_{\text{dades}} = 6 \times 2.000 = 12 \text{ us}$$