

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/06/2012	12:00

05.573R16R06R12REEE
05.573 16 06 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? Quant?
- Indicacions específiques per a la realització d'aquest examen:

Enunciats

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	16/06/2012	12:00

Pregunta 1 (20%: 1.1: 10%, 1.2: 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 de la subrutina següent:

```
; Calcular una potència: potència=base^exponent per exponents >= 0. (pow=b^e)
; Si l'exponent és 0, pow=1.
; Paràmetres d'entrada : base(rbx), exponent(rcx)
; Paràmetres de sortida: rax (pow = b^e)
p_pow:
    push rbp
    mov rbp, rsp
    push rbx                ;guardem a la pila els registres que modifiquem
    push rcx                ;per deixar tots els registres com estaven abans de
sortir.
    push rdx

    mov eax, 1              ;on guardarem pow

    cmp ecx, 0
    jle p_pow_end          ;si e=0, b^0=1.

p_pow_loop:
    mul ebx
    dec ecx
    jg    p_pow_loop

p_pow_end:
    pop rdx                ;restaurar l'estat dels registres.
    pop rcx
    pop rbx
    mov rsp, rbp
    pop rbp
    ret
```

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	16/06/2012	12:00

Pregunta 1.2 (10%)

Completa el codi de la subrutina següent:

```
; Ordenar el polinomi r(x) pel mètode de la bombolla.
; Ordenar, segons l'exponent, els termes del polinomi de més gran a més petit
; S'ha de mantenir el coeficient de cada terme associat a l'exponent que l'hi correspon,
; Si es canvia un exponent de lloc al vector v_exp,
; s'ha de fer el mateix canvi al vector dels coeficients v_coef.
; Paràmetres d'entrada : Cap
; Paràmetres de sortida: Cap
p_bubbleSort:
...
mov esi, 0                ;inicialitzem els índex
mov edi, 0
mov eax, r_coef
mov ebx, r_exp            ;vector que ordenem de més gran a més petit

p_bubble_sort_loop1:
mov ecx, [eax+esi*4]      ;agafem el terme amb el volem comparar
mov edx, [ebx+esi*4]
mov edi, esi
inc edi                  ;ordenem a partir de la posició següent.
p_bubble_sort_loop2:
cmp [ebx+edi*4], edx
    jle p_bubble_sort_no_xchg

    p_bubble_sort_xchg:
    xchg ecx, [eax+edi*4];intercanviem tot el terme,
    xchg edx, [ebx+edi*4];coeficient i exponent amb el terme
                                ;més gran que hem trobat fins el moment

p_bubble_sort_no_xchg:
inc edi
cmp edi, 20
jle p_bubble_sort_loop2

    mov [eax+esi*4], ecx ;terme més gran que hem trobat
    mov [ebx+esi*4], edx ;en cada iteració.

inc esi
cmp esi, 20
jle p_bubble_sort_loop1

p_bubble_sort_end:
...
ret
```

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/06/2012	12:00

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

Apartat 2.1 (15%)

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

- Registres: $R_i = 4 * i$ per a $i = 0, 1, \dots, 15$.
- Memòria: $M(i) = (800h + i) \bmod 2^{32}$ per a $i = 0, 4, 8, \dots, 2^{32}-4$.

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

Suposeu que l'adreça simbòlica A val 200h.

a)
<pre>MOV R1, A ADD R1, [R1+334h] MOV R3, -5</pre>
<pre>R1 = F34h R3 = FFFFFFFBh Z = 0, C = 0, S = 0, V = 0</pre>

b)
<pre>MOV R1, 1 SUB R6, 0Dh JLE F MOV R1, FFFFFFFFh F:</pre>
<pre>R1 = FFFFFFFFh R6 = 0Bh Z = 0, C = 0, S = 0, V = 0</pre>

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.

També tenim un vector A de 100 elements que volem inicialitzar a partir de la matriu anterior M.

Completeu els buits del fragment de codi CISCA per inicialitzar la posició triada del vector A amb l'element $M[i][j]$, la qual cosa en C s'especificaria amb la sentència:

$A[j] = M[i][j];$

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.

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/06/2012	12:00

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
PUSH R3
MUL R1, 100
ADD R1, R2
MUL R1, 4; o també SAL R1, 2
MOV R3, [M+R1]
SAL R2, 2
MOV [A+R2], R3
POP R3
POP R2
POP R1

```

Apartat 2.3 (10%)

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

```

SAL R2, 2
MOV R1, [V+R2]
AND 20h, [R1]

```

Tradueix-lo a llenguatge màquina i expresseu-lo en la següent taula. Supposeu que la primera instrucció del codi s'ensambla a partir de l'adreça 00FF0000h (que és el valor del PC abans de començar l'execució del fragment de codi). Supposeu 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

Instrucció	B0<7..4>	B0<3..0>	Direccionament operands
SAL	3h	5h	Veure taula de modes

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	16/06/2012	12:00

MOV	1h	0h	Veure taula de modes
AND	3h	0h	Veure taula de modes

Taula de modes de direccionament (Bk<7..4>)

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

Taula de modes de direccionament (Bk<3..0>)

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

		Bk para k=0..10											
@	Assemblador	0	1	2	3	4	5	6	7	8	9	10	
00FF0000	SAL R2,2	35	12	00	02	00	00	00					
00FF0007	MOV R1, [V+R2]	10	11	52	80	1D	00	00					
00FF000E	AND 20h, [R1]	30	00	20	00	00	00	31					

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

Pregunta 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)

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/06/2012	12:00

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, 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).

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):

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

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

1, 2, 3, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1, 2, 3, 25, 8, 9, 10

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: 16	Fallada: 1	Fallada: 25	Fallada: 8
Línia 0	0, 1, 2, 3	16, 17, 18, 19	0, 1, 2, 3		
Línia 1	4, 5, 6, 7				
Línia 2	8, 9, 10, 11			24, 25, 26, 27	8, 9, 10, 11
Línia 3	12, 13, 14, 15				

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

$$T_f = 4 \text{ fallades} / 20 \text{ accessos} = 0,2$$

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 20 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitja d'accés a memòria (t_m) ?

$$t_m = T_f \times t_f + (1 - T_f) \times t_e = 0,2 * 20 \text{ ns} + 0,8 * 4 \text{ ns} = 4 \text{ ns} + 3,2 \text{ ns} = 7,2 \text{ ns}$$

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/06/2012	12:00

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, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1, 2, 3, 25, 8, 9, 10

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 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 nou estat de la cau.

	Estat Inicial	Fallada: 16	Fallada: 25	Fallada: 8	Fallada:
Línia 0	0, 1, 2, 3				
Línia 1	4, 5, 6, 7	16, 17, 18, 19			
Línia 2	8, 9, 10, 11		24, 25, 26, 27		
Línia 3	12, 13, 14, 15			8, 9, 10, 11	

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

$$T_f = 3 \text{ fallades} / 20 \text{ accessos} = 0,15$$

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 20 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitja d'accés a memòria (t_m) ?

$$t_m = T_f \times t_f + (1-T_f) \times t_e = 0,15 * 20 \text{ ns} + 0,85 * 4 \text{ ns} = 3 \text{ ns} + 3,4 \text{ ns} = 6,4 \text{ ns}$$

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	16/06/2012	12:00

Pregunta 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 aquest valor

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 us / 32.000 us = 0,002 → 0,2 %

Pregunta 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, $t_{\text{transf_dada}}$: $t_{\text{cessió}} + t_{\text{mem}} + t_{\text{recup}} = 4 + 2 = 6$ ns
 Nombre de peticions del Bus, N_{dades} : 2.000
 Temps total d'ocupació del Bus $t_{\text{transf_bloc}}$: $t_{\text{transf_dada}} \times N_{\text{dades}} = 6 \times 2.000 = 12$ us

Examen 2011/12-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	16/06/2012	12:00