

## Examen 2017/18-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	16/06/2018	12:00

05.573R16R06R18REEXE  
05.573 16 06 18 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 2017/18-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/06/2018	12:00

### 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 2017/18-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	16/06/2018	12:00

## Pregunta 1

### 1.1 Pràctica – 1a Part

**Escriure el fragment de codi ensamblador que falta a la subrutina copyMatrixP1 per a copiar la matriu (tilesIni) sobre la matriu (tiles), que són matrius de tipus BYTE. (No s'ha d'escriure el codi de tota la subrutina).**

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Copiar la matriu (tilesIni) a la matriu (tiles), de tipus char
; (1 byte cada posició).
; Per a accedir a les matrius (tilesIni)(tiles) s'ha d'accedir amb un
; index que inicialment valdrà 0 i s'ha d'incrementar fins a 15. 0,1,2,3
; correspon a la fila 1, 4,5,6,7 correspon a la fila 2, ..., fins al 15.
; Després, actualitzar la posició de l'espai a la pantalla
; (rowcolSpaceScreen) (fila (DWORD[rowcolSpaceScreen+0])=15 i la columna
; (DWORD[rowcolSpaceScreen+4])=19).
;
; Variables globals utilitzades:
; tilesIni      : Matriu amb els nombres inicials del joc
; tiles         : Matriu on guardem els nombres del joc.
; rowcolSpaceScreen: Vector on tenim la posició de l'espai a la pantalla.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
copyMatrixP1:
    push rbp
    mov  rbp, rsp
    ...
    mov  edx, 0                ;index matriu
    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+edx] ;al = tilesIni[i][j]
    mov  BYTE[tiles+edx], al    ;tiles[i][j] = al
    inc  edx
    inc  ecx                    ;i++
    jmp  copyMatrixP1_bucle_Col

    copyMatrixP1_Col_end:
    inc  ebx                    ;j++
    jmp  copyMatrixP1_bucle_Row

copyMatrixP1_Row_end:
;Indiquem la posició de l'espai.
mov  DWORD[rowcolSpaceScreen+0], 15 ;rowcolSpaceScreen[0]=15;
mov  DWORD[rowcolSpaceScreen+4], 19 ;rowcolSpaceScreen[1]=19;
copyMatrixP1_end:
    ...
    mov  rsp, rbp
    pop  rbp
    ret

```

## Examen 2017/18-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/06/2018	12:00

### 1.2 Pràctica – 2a part

**Completar el codi de la subrutina updateBoardP2. (Només completar els espais marcats, no es poden afegir, ni 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 columna (esi) rebuts com a paràmetres
; cridant a la funció gotoxyP2_C.
;
; Variables globals utilitzades: Cap.
; Paràmetres d'entrada : rdi(edi): Fila
;                      rsi(esi): Columna.
; Paràmetres de sortida: Cap.
; Aquesta subrutina es dona feta. NO LA PODEU MODIFICAR.
; Mostrar un caràcter (dil) a la pantalla, rebut com a paràmetre,
; en la posició on està el cursor cridant la funció printchP2_C.
;
; Variables globals utilitzades: Cap
; Paràmetres d'entrada : rdi(dil): Caràcter que volem mostrar
; Paràmetres de sortida: Cap
; Aquesta subrutina es dona feta. NO LA PODEU MODIFICAR.
; Mostrar els valors de la matriu (t), rebuda com a paràmetre, a la
; pantalla, dins el taulell a les posicions corresponents.
; S'ha de recórrer tota la matriu (t), de tipus char (1 byte cada
; posició), i per a cada element de la matriu:
; Posicionar el cursor en el taulell cridant la funció gotoxyP2_C.
; La posició inicial del cursor és la fila 11, columna 11
; que és posició a pantalla de la casella [0][0].
; Mostrar els caràcters de cada posició de la matriu (t)
; cridant la funció printchP2_C.
; Després, posicionar el cursor a la pantalla en funció del vector
; (rcCurScreen) rebut com a paràmetre (fila (DWORD[rcCurScreen+0]) i la
; columna (DWORD[rcCurScreen+4]) cridant la funció gotoxyP2_C().
;
; Variables globals utilitzades: Cap
; Paràmetres d'entrada : rdi : Matriu on guardem els nombres del joc.
;                      rsi : Vector on tenim la posició del cursor a la pantalla.
; Paràmetres de sortida: Cap
; Aquesta subrutina es dona feta. NO LA PODEU MODIFICAR.
updateBoardP2:
    push rbp
    mov rbp, rsp
    ...
    mov rbx, rdi                ;rbx: t
    mov rdx, rsi                ;rdx: rcCurScreen
    mov rcx, 0                  ;rcx: indexMat
    mov edi, 11                 ;rowScreen=11;
    mov r10d, 0                 ;i=0
updateBoardP2_bucle_Row:
    cmp r10d, DimMatrix         ;i<DimMatrix
    jge updateBoardP2_Cur
    mov esi, 11                 ;colScreen=11;
    mov r11d, 0                 ;j=0;
updateBoardP2_bucle_Col:
    cmp r11d, DimMatrix         ;j<DimMatrix
    jge updateBoardP2_Col_end
    call gotoxyP2                ;gotoxyP2_C(rowScreen, colScreen);
    push rdi

```

## Examen 2017/18-2

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573	16/06/2018	12:00

```

    mov  dil, __BYTE[rbx+rcx]__ ;charac = t[i][j];
    _call_ printchP2           ;printchP2_C(t[i][j]);
    _pop__ rdi
    inc  rcx
    inc  r11d                  ;j++;
    add  __esi__, 4            ;colScreen = colScreen + 4;
    jmp  updateBoardP2_bucle_Col
updateBoardP2_Col_end:
    inc  r10d                  ;i++;
    add  edi, 2                ;rowScreen = rowScreen + 2;
    jmp  updateBoardP2_bucle_Row
updateBoardP2_Cur:
    mov  edi, __DWORD[rdx+0]__ ;edi = rcCurScreen[0];
    mov  esi, __DWORD[rdx+4]__ ;esi = rcCurScreen[1];
    call gotoxyP2              ;gotoxyP2_C(rcCurScreen[0],rcCurScreen[1]);
updateBoardP2_End:
    ...
    mov  rsp, rbp
    pop  rbp
    ret

```

## Examen 2017/18-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/06/2018	12:00

### Pregunta 2

#### 2.1

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

R0 = 00000A10h R1 = 00000B20h R2 = 00000C30h	M(00000A10h) = 0000F00Fh M(00000B20h) = 0000F000h M(00000C30h) = 00000FF0h M(000000F0h) = 00000001h M(000FF0A0h) = 0000000Ah	Z = 0, C = 0, S = 0, V = 0
--	--	----------------------------

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

a) XOR R0, R0 JE END ADD R0, [R2] END: MOV R1, [R2]	b) DEC [000000F0h] SUB [000000F0h], R2
R0 = 0 R1 = 00000FF0h  Z = 1, S = 0, C = 0, V = 0	M(000000F0h) = 00000000h M(000000F0h) = FFFF3D0h  Z = 0, S = 1, C = 1, V = 0

## Examen 2017/18-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/06/2018	12:00

### 2.2

Donat el següent llenguatge d'alt nivell:

```

Des_de i=0 fins 9 fer
    Si (V [i] > 10) llavors V[i] = V[i]*2
    sino V[i] = 10;
fi_des_de;
```

On V es tracta d'un vector de 10 enters de 32 bits. Es proposa el següent codi CISCA en el que heu d'omplir els 5 forats per què sigui operatiu:

```

        MOV R1, 0
cont:    CMP R1, 40
        JGE final
        CMP [V+R1], 10
        JLE m
        SAL [V+R1], 1
        JMP p
m:       MOV [V+R1], 10
p:       ADD R1, 4
        JMP cont
final:
```

## Examen 2017/18-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/06/2018	12:00

### 2.3

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

```
FOREVER:  MUL R10, [Q]
          MOV R1,[M+R2]
          JMP FOREVER
```

Tradueix-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 **003FC000h** (que és el valor del PC abans de començar l'execució del fragment de codi). Supposeu que les adreces simbòliques Q i M valen **00003A00h** i **00003C00h** respectivament. En la següent taula utilitzeu 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 'Adreça' 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 ajuda les taules de codis:

Taula de codis d'instrucció

B0	Instrucció
22h	MUL
10h	MOV
40h	JMP

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											
Adreça	Assemblador	0	1	2	3	4	5	6	7	8	9	10	
003FC000	MUL R10, [Q]	22	1A	20	00	3A	00	00					
003FC007	MOV R1,[M+R2]	10	11	52	00	3C	00	00					
003FC00E	JMP FOREVER	40	00	00	C0	3F	00						
003FC014													



## Examen 2017/18-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/06/2018	12:00

### 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:

0, 1, 2, 12, 62, 63, 25, 64, 17, 18, 19, 2, 4, 6, 65, 66, 20, 56, 42, 50

**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	0	1	2	12	62
0	0 (0 - 7)	E 0 (0 - 7)	E 0 (0 - 7)	E 0 (0 - 7)	0 (0 - 7)	0 (0 - 7)
1	1 (8 - 15)	1 (8 - 15)	1 (8 - 15)	1 (8 - 15)	E 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	7 (56 - 63)	7 (56 - 63)	7 (56 - 63)	7 (56 - 63)	7 (56 - 63)	E 7 (56 - 63)

Línia	63	25	64	17	18	19
0	0 (0 - 7)	0 (0 - 7)	F 8 (64 - 71)	8 (64 - 71)	8 (64 - 71)	8 (64 - 71)
1	1 (8 - 15)	1 (8 - 15)	1 (8 - 15)	F 2 (16 - 23)	E 2 (16 - 23)	E 2 (16 - 23)
2	2 (16 - 23)	F 3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)
3	E 7 (56 - 63)	7 (56 - 63)	7 (56 - 63)	7 (56 - 63)	7 (56 - 63)	7 (56 - 63)

## Examen 2017/18-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/06/2018	12:00

Línia	2	4	6	65	66	20
0	8 (64 - 71)	8 (64 - 71)	8 (64 - 71)	E 8 (64 - 71)	E 8 (64 - 71)	8 (64 - 71)
1	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)	E 2 (16 - 23)
2	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)
3	F 0 (0 - 7)	E 0 (0 - 7)	E 0 (0 - 7)	0 (0 - 7)	0 (0 - 7)	0 (0 - 7)

Línia	56	42	50			
0	8 (64 - 71)	8 (64 - 71)	F 6 (48 - 55)			
1	2 (16 - 23)	2 (16 - 23)	2 (16 - 23)			
2	F 7 (56 - 63)	7 (56 - 63)	7 (56 - 63)			
3	0 (0 - 7)	F 5 (40 - 47)	5 (40 - 47)			

**3.1.2 a)** Quina és la taxa d'encerts ( $T_e$ ) ?

$$T_e = 13 \text{ encerts} / 20 \text{ accessos} = 0,65$$

**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 40 ns. Considerant la taxa d'encerts obtinguda a la pregunta anterior, quin és el temps mitja d'accés a memòria ( $t_m$ ) ?

$$t_m = T_e \times t_e + (1 - T_e) \times t_f = 0,65 \times 5 \text{ ns} + 0,35 \times 40 \text{ ns} = 3,25 \text{ ns} + 14 \text{ ns} = 17,25 \text{ ns}$$

## Examen 2017/18-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/06/2018	12:00

### 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  $v_{\text{transf}} = 1 \text{ MBytes/s} = 1000 \text{ Kbytes/s}$
- 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
- El bit del **registre d'estat** que indica que el controlador del port d'E/S està disponible és el bit 2, o el tercer bit menys significatiu (quan val 1 indica que està disponible)
- Processador amb una freqüència de rellotge de 1 GHz, el temps de cicle  $t_{\text{cicle}} = 1 \text{ ns}$ . El processador pot executar 4 instruccions per cicle de rellotge
- Transferència de **lectura** des del port d'E/S cap a memòria
- Transferència de  **$N_{\text{dades}} = 100000$  dades**
- La mida d'una dada és  **$m_{\text{dada}} = 4 \text{ bytes}$**
- El temps per atendre la interrupció ( $t_{\text{rec\_int}}$ ) és de 2 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, [0B04h]
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

```

## Examen 2017/18-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/06/2018	12:00

**3.2.2** Quant temps dura la transferència del bloc de dades  $t_{\text{transf\_bloc}}$ ?

El temps d'un cicle,  $t_{\text{cicle}} = 1 \text{ ns}$  (nanosegons)

Temps per atendre la interrupció,  $t_{\text{rec\_int}}: 2 \text{ cicles} * 1 \text{ ns} = 2 \text{ ns}$

Temps d'execució de una instrucció,  $t_{\text{instr}}: t_{\text{cicle}} / 4 = 0,250 \text{ ns}$

Temps d'execució RSI,  $t_{\text{rsi}}: N_{\text{rsi}} * t_{\text{instr}} = 12 \text{ instr.} * 0,250 \text{ ns} = 3 \text{ ns}$

Temps consumit per CPU en cada interrupció,  $t_{\text{transf\_dada}}$ :

$$t_{\text{transf\_dada}} = t_{\text{rec\_int}} + t_{\text{rsi}} = 2 + 3 = 5 \text{ ns}$$

Nombre d'interrupcions produïdes (nombre total de dades,  $N_{\text{dades}}$ ): 100000 interrupcions

Temps consumit en total en TOTES les interrupcions:

$$t_{\text{transf\_bloc}} = t_{\text{transf\_dada}} * N_{\text{dades}} = 5 \text{ ns} * 100000 \text{ interrupcions} = 500000 \text{ ns} = 0,5 \text{ ms (milisegons)}$$

**3.2.3** Quin és el percentatge d'ocupació del processador? Percentatge que representa el temps de transferència del bloc  $t_{\text{transf\_bloc}}$  respecte al temps de transferència del bloc per part del perifèric  $t_{\text{bloc}}$

$$t_{\text{bloc}} = t_{\text{latència}} + (N_{\text{dades}} * t_{\text{dada}})$$

$$t_{\text{latència}} = 0$$

$$N_{\text{dades}} = 100000$$

$$t_{\text{dada}} = m_{\text{dada}} / v_{\text{transf}} = 4 / 1000 \text{ Kbytes/s} = 0,004 \text{ ms}$$

$$t_{\text{bloc}} = 0 + (100000 * 0,004) \text{ ms} = 400 \text{ ms}$$

$$\% \text{ ocupació} = (t_{\text{transf\_bloc}} * 100 / t_{\text{bloc}}) = (0,5 * 100) / 400 = 0,125\%$$

## Pregunta 4

### 4.1

Què és una interrupció i com s'activa?

Les interrupcions són el mecanisme mitjançant el qual un dispositiu extern al processador pot interrompre el programa que està executant el processador amb la finalitat d'executar un altre programa (una rutina de servei a la interrupció o RSI) per donar servei al dispositiu que ha produït la interrupció.

La petició d'interrupció s'efectua activant alguna de les línies de petició de les quals disposa el processador.

### 4.2

**4.2.1** En la memòria cau, quines polítiques d'assignació es defineixen? Descriure-les breument.

- Política d'assignació directa:** un bloc de la memòria principal només pot ser en una única línia de la memòria cau. La memòria cau d'assignació directa és la que té la taxa de fallades més alta, però s'utilitza molt perquè és la més barata i fàcil de gestionar.

## Examen 2017/18-2

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	16/06/2018	12:00

- 2) **Política d'assignació completament associativa:** un bloc de la memòria principal pot ser en qualsevol línia de la memòria cau. La memòria cau completament associativa és la que té la taxa de fallades més baixa. No obstant això, no se sol utilitzar perquè és la més cara i complexa de gestionar.
- 3) **Política d'assignació associativa per conjunts:** un bloc de la memòria principal pot ser en un subconjunt de les línies de la memòria cau, però dins del subconjunt pot trobar-se en qualsevol posició.

**4.2.2** Quins són els passos bàsics per a la gestió d'una interrupció en un sistema amb una única línia d'interrupció i un únic mòdul d'E/S?

- 1.- Petició del mòdul d'entrada/Sortida
- 2.- Cicle de reconeixement de la interrupció
  - 2.a.- Reconeixement de la interrupció
  - 2.b.- Salvaguarda de l'estat del processador
  - 2.c.- Crida a la RSI
- 3.- Execució de la rutina de servei d'interrupció
  - 3.a.- Inici de l'execució de la RSI
  - 3.b.- Intercanvi de la dada
  - 3.c Finalització de l'execució de la RSI
  - 3.d Retorn d'interrupció: Restaurar l'estat del processador.