



PAC2

Estructura de computadors

2018 s2

Estudis d'informàtica, multimèdia i comunicació

Presentació

La present PAC2 conté 2 preguntes amb diversos apartats cadascuna d'elles i representa el 50% de la nota de l'avaluació contínua.

Com podreu veure, els exercicis són molt semblats als quals heu fet durant aquests dies, en els quals a més heu pogut donar les solucions, comentar-les i plantejar dubtes en el fòrum. Aquesta PAC és **individual**, **avaluable** i per tant no pot comentar-se.

Competències

Les competències específiques que persegueix la PAC2 són:

- [13] Capacitat per identificar els elements de l'estructura i els principis de funcionament d'un ordinador.
- [14] Capacitat per analitzar l'arquitectura i organització dels sistemes i aplicacions informàtics en xarxa.
- [15] Conèixer les tecnologies de comunicacions actuals i emergents i saber-les aplicar convenientment per dissenyar i desenvolupar solucions basades en sistemes i tecnologies de la informació.

Objectius

Els objectius de la següent PAC són:

- Conèixer la organització del sistema de memòria d'un computador.
- Conèixer el funcionament de la memòria cau, així com dels algorismes de correspondència i reemplaçament.
- Conèixer la organització del sistema d'entrada/sortida.
- Comprendre les tècniques d'entrada/sortida (entrada/sortida programada, Interrupcions i DMA).

Enunciat

Respondre cada pregunta o apartat en el requadre corresponent.

Recursos

Podeu consultar els recursos disponibles a l'aula, però no fer ús del fòrum.

Criteris de valoració

La **puntuació** de cada pregunta i els **criteris d'avaluació** els trobareu a cada pregunta.

Format i data de lliurament

La PAC2 podeu lliurar-la a l'apartat de **lliurament d'activitats** amb el nom **cognom1_cognom2_nom_PAC2 (pdf / odt / doc / docx)**.

La data límit de lliurament és el **03/05/2019**.



Enunciat

Pregunta 1 (4 punts)

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 $8*N$, $8*N+1$, $8*N+2$, $8*N+3$, $8*N+4$, $8*N+5$, $8*N+6$, $8*N+7$.

Una fórmula per calcular el identificador numèric del bloc és la següent:

Bloc = adreça de memòria (adreça paraula) DIV 8 (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, 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).

Pregunta 1.1 (2 punts) Memòria cau d'assignació directa

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):

Línia = identificador de bloc MOD 4 (mida de la cau en línies)

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

18, 19, 20, 12, 13, 21, 1, 30, 31, 32, 33, 7, 42, 43, 24, 6, 59, 60, 28, 15

1.1.a) (1,2 punts)

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	18	19	20	12	13
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)	E 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)	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)

Línia	21	1	30	31	32	33
0	0:0 (0 - 7)	E 0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	F 4:1 (32-39)	E 4:1 (32-39)
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	E 2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)
3	3:0 (24 - 31)	3:0 (24 - 31)	E 3:0 (24 - 31)	E 3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)

Línia	7	42	43	24	6	59
0	F 0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	E 0:0 (0 - 7)	0:0 (0 - 7)
1	1:0 (8 - 15)	F 5:1 (40 - 47)	E 5:1 (40 - 47)	5:1 (40 - 47)	5:1 (40 - 47)	5:1 (40 - 47)
2	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)
3	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	E 3:0 (24 - 31)	3:1 (24 - 31)	F 7:1 (56 - 63)

Línia	60	28	15			
0	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)			
1	5:1 (40 - 47)	5:1 (40 - 47)	F 1:0 (8 - 15)			
2	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)			
3	E 7:1 (56 - 63)	F 3:0 (24 - 31)	3:0 (24 - 31)			

1.1.b) (0,4 punts)

Quina és la taxa de fallades (T_f) ?

$$T_f = 6 \text{ fallades} / 20 \text{ accessos} = 0,3$$

@Ó

1.1.c) (0,4 punts) Suposem que el temps d'accés a la memòria cau, o temps d'accés en cas d'encert (t_e), és de 3 ns i el temps total d'accés en cas de fallada (t_f) és de 30 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitjà d'accés a memòria (t_m) ?

$$t_m = T_f \times t_f + (1 - T_f) \times t_e = 0,3 \times 30 \text{ ns} + 0,7 \times 3 \text{ ns} = 9 \text{ ns} + 2,1 \text{ ns} = 11,1 \text{ ns}$$

Pregunta 1.2 (2 punts) Memòria cau d'accés completament associatiu

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 es referència.

Considerem la mateixa llista de lectures a memòria:

18, 19, 20, 12, 13, 21, 1, 30, 31, 32, 33, 7, 42, 43, 24, 6, 59, 60, 28, 15



1.2.a) (1,2 punts)

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.

Línea	Estado Inicial	18	19	20	12	13
0	0 (0 - 7)	0 (0 - 7)	0 (0 - 7)	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)	E 1 (8 - 15)
2	2 (16 - 23)	E 2 (16 - 23)	E 2 (16 - 23)	E 2 (16 - 23)	2 (16 - 23)	2 (16 - 23)
3	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)

Línea	21	1	30	31	32	33
0	0 (0 - 7)	E 0 (0 - 7)	0 (0 - 7)	0 (0 - 7)	0 (0 - 7)	0 (0 - 7)
1	1 (8 - 15)	1 (8 - 15)	1 (8 - 15)	1 (8 - 15)	F 4 (32 - 39)	E 4 (32 - 39)
2	E 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)	E 3 (24 - 31)	E 3 (24 - 31)	3 (24 - 31)	3 (24 - 31)

Línea	7	42	43	24	6	59
0	E 0 (0 - 7)	0 (0 - 7)	0 (0 - 7)	0 (0 - 7)	E 0 (0 - 7)	0 (0 - 7)
1	4 (32 - 39)	4 (32 - 39)	4 (32 - 39)	4 (32 - 39)	4 (32 - 39)	F 7 (56 - 63)
2	2 (16 - 23)	F 5 (40 - 47)	E 5 (40 - 47)	5 (40 - 47)	5 (40 - 47)	5 (40 - 47)
3	3 (24 - 31)	3 (24 - 31)	3 (24 - 31)	E 3 (24 - 31)	3 (24 - 31)	3 (24 - 31)

Línea	60	28	15			
0	0 (0 - 7)	0 (0 - 7)	0 (0 - 7)			
1	E 7 (56 - 63)	7 (56 - 63)	7 (56 - 63)			
2	5 (40 - 47)	5 (40 - 47)	F 1 (8 - 15)			
3	3 (24 - 31)	E 3 (24 - 31)	3 (24 - 31)			

1.2.b) (0,4 punts)

Quina és la taxa de fallades (T_f) ?

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

1.2.c) (0,4 punts)

Suposem que el temps d'accés a la memòria cau, o temps d'accés en cas d'encert (t_e), és de 3 ns i el temps total d'accés en cas de fallada (t_f) és de 30 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitjà d'accés a memòria (t_m) ?

$$t_m = T_f \times t_f + (1 - T_f) \times t_e = 0,2 \times 30 \text{ ns} + 0,8 \times 3 \text{ ns} = 6 \text{ ns} + 2,4 \text{ ns} = 8,4 \text{ ns}$$

Criteris de valoració. Pels apartats 1.1.a i 1.2.a cada errada en les fallades o encerts de la memòria cau o en la col·locació d'un bloc a la cau resta 0,6. Els apartats restants es puntuaran amb els 0,4 punts cadascun d'ells si la solució és correcta i coherent amb la vostra resposta als apartats a) corresponents.

Pregunta 2 (5 punts)

Es vol analitzar el rendiment de la comunicació de dades entre la memòria d'un processador i un port USB, que tenen les següents característiques:

- Velocitat de transferència del dispositiu d'E/S (v_{transf}) = 2 MB/s
- Temps de latència mitjà del dispositiu ($t_{\text{latència}}$) = 0
- Adreces dels **registres de dades i d'estat** del controlador d'E/S: 0A00h i 0A04h
- El bit del **registre d'estat** que indica que el controlador del port d'E/S està disponible és el bit 3, o el quart bit menys significatiu (quan val 1 indica que està disponible)
- Processador amb una freqüència de rellotge de 500 MHz, el processador pot executar 1 instrucció per cicle de rellotge ($t_{\text{instr}} = t_{\text{cicle}}$)
- Temps de programació i finalització de la transferència de 200 ns ($t_{\text{prog}} + t_{\text{final}}$)
- Transferència de **escriptura** des de memòria al port d'E/S
- En cada escriptura d'una dada es transfereixen 4 Bytes
- Transferència de $N_{\text{dades}}=800.000$ dades, es a dir, $800.000 * 4 \text{ Bytes} = 3.200.000$ Bytes
- Adreça inicial de memòria on resideixen les dades: 80000000h

Apartat 2.1 (2 punts) E/S programada

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

```
1.      MOV    R3, VALOR1
2.      MOV    R2, VALOR2
3. Bucle: IN     R0, [VALOR3]      ; llegir 4 bytes
4.      AND    R0, VALOR4
5.      JE     Bucle
6.      MOV    R0, [R2]           ; llegir 4 bytes
7.      ADD    R2, 4
8.      OUT    [VALOR5], R0      ; escriure 4 bytes
9.      SUB    R3, 1
10.     JNE    Bucle
```

2.1.a) (0,5 punts)

Substituir pels valors adients:

VALOR1= 800000

VALOR2= 80000000h

VALOR3= A04h

VALOR4= 00001000b= 08h = 8d

VALOR5= A00h



2.1.b) (0,5 punts)

Quant temps dura la transferència $t_{\text{transf_bloc}}$?

Quin percentatge d'aquest temps dedica la CPU a la transferència?

$$t_{\text{transf_bloc}} = t_{\text{atència}} + (N_{\text{dades}} * t_{\text{transf_dada}})$$

$$t_{\text{atència}} = 0$$

$$N_{\text{dades}} = 800000$$

$$t_{\text{transf_dada}} = m_{\text{dada}} / v_{\text{transf}} = 4 \text{ Bytes} / 2 \text{ Mbytes/s} = 0,002 \text{ ms}$$

$$t_{\text{transf_bloc}} = 0 + (800000 * 0,002 \text{ ms}) = 1600 \text{ ms} = 1,6 \text{ s}$$

La CPU dedica el 100% del temps i per tant, el temps coincideix amb el temps dedicat pel perifèric t_{bloc}

2.1.c) (1 punt)

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?

Freqüència de rellotge = 500 MHz, implica un temps de cicle de

$$t_{\text{cicle}} = 1/500 * 10^9 = 2 \text{ ns (nanosegons)}$$

$$t_{\text{instr}} = 2 \text{ ns}$$

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 $8 * t_{\text{instr}} = 8 * 2 \text{ ns} = 16 \text{ ns}$

Per tant, el temps mínim per a transferir una dada és: 16 ns

Es poden transferir 4 bytes cada 16 ns, és a dir: $4 / 16 * 10^{-9} = 250 \text{ Mbyte/s}$

Apartat 2.2 (2 punts) E/S per Interrupcions

Suposeu que el següent codi CISCÀ és una rutina de servei a les interrupcions (RSI) per a transferir a través del dispositiu d'E/S anterior, el mateix nombre de dades que abans amb E/S programada, però ara mitjançant la tècnica de E/S per interrupcions.

Suposeu:

- El temps per atendre la interrupció (t_{rec_int}), o temps addicional des de que la CPU detecta la interrupció fins que es comença a executar la primera instrucció de la RSI és de 5 cicles de rellotge.
- Es fa servir una variable global que es representa amb l'etiqueta **Dir**, i que al principi del programa conté l'adreça inicial de memòria on resideixen les dades a transferir.

```
1.  CLI
2.  PUSH R0
3.  PUSH R1
4.  IN    R0, [VALOR1] ; llegir 4 bytes
5.  AND   R0, VALOR2
6.  JE    Error ;salta a un codi de tractament de l'error no
                ;descrit, s'ha produït la petició per part del
                ;dispositiu però la dada no està disponible

7.  MOV   R1, [VALOR3]
8.  MOV   R0, [R1]
9.  OUT   [VALOR4],R0 ; escriure 4 bytes
10. ADD   R1, 4
11. MOV   [VALOR3], R1
12. POP   R1
13. POP   R0
14. STI
15. RETI
```

2.2.a) (0,5 punts)

Substituir pels valors adients:

VALOR1= A04h

VALOR2= 08h = 8d

VALOR3= Dir

VALOR4= A00h



2.2.b) (1 punt)

Quin és el temps total que dedica la CPU a la tasca d'Entrada/Sortida, t_{cpu} ? Quin percentatge del temps representa el tems de transferència del bloc t_{transf_bloc} respecte al temps de transferència del bloc per part del perifèric t_{bloc} ?

El temps d'un cicle, $t_{cicle} = 2$ ns (nanosegons)

Temps per atendre la interrupció, t_{rec_int} : $5 \text{ cicles} * 2 \text{ ns} = 10 \text{ ns}$

Temps d'execució de una instrucció, $t_{instr} = 2$ ns

Temps d'execució RSI, t_{rsi} : $N_{rsi} * t_{instr} = 15 \text{ instr.} * 2 \text{ ns} = 30 \text{ ns}$

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

$$t_{transf_dada} = t_{rec_int} + t_{rsi} = 10 + 30 = 40 \text{ ns}$$

Nombre d'interrupcions produïdes (nombre total de dades, N_{dades}): 800.000 interrupcions

Temps consumit en total en TOTES les interrupcions:

$$t_{transf_bloc} = t_{transf_dada} * N_{dades} = 40 \text{ ns} * 800.000 \text{ interrupcions} = 32 \text{ ms} \text{ (milisegons)}$$

El temps final d'ocupació de la CPU cal que inclogui el temps de programació i finalització de la transferència:

$$t_{cpu} = (t_{prog} + t_{final}) + t_{transf_bloc} = 200 \text{ ns} + 32000000 \text{ ns} = 32000200 \text{ ns} = 32,0002 \text{ ms}$$

Dels 1,6 s = 1600 ms de temps total per a realitzar la transferència (temps calculat a l'apartat 2.1 b), la CPU està dedicada a la tasca de E/S:

$$\%ocupació = t_{transf_bloc} * 100 / t_{bloc} = 32 * 100 / 1600 \Rightarrow 2 \% \text{ del temps.}$$

2.2c) (0,5 punts)

Si volguéssim reduir la freqüència de rellotge del processador per a reduir el seu consum energètic, fins a quina freqüència ho podríem fer sense reduir la velocitat de transferència amb el dispositiu d'E/S?

En la fase de transferència de dades, el controlador d'E/S genera 800.000 interrupcions durant 1,6 segons (dada calculada a l'apartat 2.1 b):

$$N_{dades} * t_{dada} = 1,6 \text{ s} = 1600 \text{ ms} = 1600000 \text{ us} \text{ (microsegons)}$$

Es a dir, tenim una interrupció cada $1600000 / 800000 = 2 \text{ us}$ (microsegons). Aquest és el temps màxim que hauria de tardar la gestió de la interrupció, incloent el temps addicional per transferir el control a la RSI, el temps que pot consumir la CPU en una interrupció és

t_{transf_dada}

El temps consumit per la CPU en cada interrupció és, com hem vist a l'apartat anterior, la suma del temps de transferir el control a la RSI + executar la RSI:

A l'enunciat es defineix que $t_{rec_int} = 5$ cicles de rellotge = $5 * t_{cicle}$ i per tant:

$$t_{transf_dada} = t_{rec_int} + t_{rsi} = t_{rec_int} + (N_{rsi} * t_{instr}) = 5 * t_{cicle} + (15 * t_{instr})$$

Tal com hem vist a l'apartat anterior, i com diu l'enunciat, el temps d'una instrucció és: $t_{instr} = t_{cicle}$

$$\text{Per tant: } t_{transf_dada} = 5 * t_{cicle} + (15 * t_{cicle}) = 20 * t_{cicle}$$

Volem trobar el temps de cicle tal que el temps de transferència d'una dada sigui 2 us:

$$2 \text{ us} = 20 * t_{cicle} \Rightarrow t_{cicle} = 2 / 20 = 0,1 \text{ us}$$

$$1 / 0,1 * 10^{-6} = 10 \text{ MHz}$$

Apartat 2.3 (1 punt) E/S per DMA

Suposarem que el controlador d'ES pot funcionar en mode DMA (Accés Directe a Memòria). La suma del **temps de cessió** del bus i del **temps de recuperació** del bus es de 10 ns ($t_{\text{cessió}} + t_{\text{recup}} = 10 \text{ ns}$). El **temps de la transferència** pel bus es de 1 ns ($t_{\text{mem}} = 1 \text{ ns}$).

2.3.a) (0,5 punts)

Considerem que en la transferència per DMA, les dades s'envien entre el controlador de DMA i la memòria, en mode ràfega, i es disposa d'un buffer de mida $m_{\text{buffer}} = 400$ bytes. Calcular el temps total d'ocupació del bus per part del controlador de DMA per a dur a terme la transferència que venim analitzant.

Mida de les ràfegues $N_{\text{ràfega}} : m_{\text{buffer}} / m_{\text{dada}} = 400 / 4 = 100$

Temps ocupació Bus, $t_{\text{transf_ràfega}} : t_{\text{cessió}} + 100 * t_{\text{mem}} + t_{\text{recup}} = 10 + 100 * 1 = 110 \text{ ns}$

Nombre de peticions del Bus, $N_{\text{dades}} / N_{\text{ràfega}} : 800.000 / 100 = 8000$

Temps total d'ocupació del Bus $t_{\text{transf_bloc}} : t_{\text{transf_dada}} * (N_{\text{dades}} / N_{\text{ràfega}}) = 110 * 8000 = 880000 \text{ ns} = 880 \text{ us} = 0,88 \text{ ms}$

2.3.b) (0,5 punts) La CPU no pot fer cap tasca durant tot el temps en que el bus està ocupat per part del controlador de DMA. Quin percentatge de temps té disponible la CPU per a executar codi efectiu d'altres programes durant la transferència?

$t_{\text{transf_bloc}} = 0,88 \text{ ms}$

$t_{\text{bloc}} = 1600 \text{ ms}$

$\% \text{ocupació} = (t_{\text{transf_bloc}} * 100) / t_{\text{bloc}}$

Percentatge de temps disponible: $100 - \% \text{ocupació} = 100 - (t_{\text{transf_bloc}} * 100) / t_{\text{bloc}} = 100 - (0,88 * 100) / 1600 = 100 - 0,055 = 99,945\%$

Criteris de valoració. En els apartats 2.1.a i 2.2.a cada valor erroni resta 0,25. La resta d'apartats estan bé o estan malament. No hi ha gradació.



Pregunta 3 (1 punt)

Qüestions teòriques

3.a) (0,25 punts)

En què consisteix la proximitat referencial? Quins tipus de proximitat referencial podem distingir?

El codi dels programes s'organitza en subrutines, té estructures iteratives i treballa amb conjunts de dades agrupades. Això, unit al fet que l'execució del codi és seqüencial, porta al fet que durant un interval de temps determinat s'utilitzi només una petita part de tota la informació emmagatzemada: aquest fenomen es denomina proximitat referencial.

Distingim dos tipus de proximitat referencial:

1) Proximitat temporal. Es produeix quan en un interval de temps determinat la probabilitat que un programa accedeixi de manera repetida a les mateixes posicions de memòria és molt gran. La proximitat temporal es deu principalment a les estructures iteratives; un bucle executa les mateixes instruccions repetidament, de la mateixa manera que les crides repetitives a subrutines.

2) Proximitat espacial. Es produeix quan la probabilitat que un programa accedeixi a posicions de memòria pròximes és molt gran. La proximitat espacial es deu principalment al fet que l'execució dels programes és seqüencial –s'executa una instrucció darrera l'altra excepte les bifurcacions- i també a la utilització d'estructures de dades que estan emmagatzemades en posicions de memòria contigües.

3.b) (0,25 punts)

A la memòria cau, quan s'ha d'utilitzar un algoritme de reemplaçament? Quins són els principals algoritmes de reemplaçament?

Quan es produeix una fallada de memòria cau i s'ha de dur a la memòria cau un bloc de memòria principal determinat, si aquest bloc de memòria es pot emmagatzemar en més d'una línia de la memòria cau, cal decidir en quina línia de totes les possibles es posa, i sobre escriure les dades que es troben en aquella línia. L'algorisme de reemplaçament s'encarrega d'aquesta tasca.

Els principals algoritmes de reemplaçament són:

1) FIFO (first in first out). Per triar la línia s'utilitza una cua, de manera que la línia que fa més temps que està emmagatzemada a la memòria cau serà la reemplaçada. Aquest algoritme pot reduir el rendiment de la memòria cau perquè la línia que es troba emmagatzemada a la memòria cau des de fa més temps no ha de ser necessàriament la que s'utilitzi menys. Es pot implementar fàcilment utilitzant tècniques de buffers circulars (o round-robin): cada vegada 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 des del principi.

2) LFU (least frequently used). En aquest algoritme es tria 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 algoritme tria la línia que fa més temps que no s'utilitza. És l'algoritme més eficient, però el més difícil d'implementar, especialment si s'ha de triar entre moltes línies. S'utilitza habitualment en memòries cau associatives per conjunts, amb conjunts petits de 2 o 4 línies. Per 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 indicar quina de les dues línies ha estat l'última que s'ha utilitzat.

4) Aleatori. Els algoritmes anteriors es basen en factors relacionats amb la utilització de les línies de la memòria cau; en canvi, aquest algoritme tria la línia que s'ha de reemplaçar a l'atzar. Aquest algoritme és molt simple i s'ha demostrat que té un rendiment només lleugerament inferior als algoritmes que tenen en compte factors d'utilització de les línies.

3.c) (0,25 punts)

¿Quines són les tres parts bàsiques d'un mòdul d'E/S? Quins tipus de registres inclou un mòdul d'E/S?

En un mòdul d'E / S distingim tres parts bàsiques:

- 1) Una interfície interna normalitzada amb la resta del computador mitjançant el bus de sistema que ens dona accés al banc de registres del mòdul d'E/S.
- 2) Una interfície externa específica per al perifèric que controla. Habitualment la connexió amb el perifèric es realitza mitjançant un sistema d'interconnexió normalitzat d'E/S.
- 3) La lògica necessària per gestionar el mòdul d'E/S. És responsable del pas d'informació entre la interfície interna i externa.

Els registres que inclou un mòdul d'E / S són:

- Registres de control.
- Registres d'estat.
- Registres de dades.

3.d) (0,25 punts)

En la E/S programada, com es realitza la transferència d'una dada? Descriure el procés.

Per fer l'operació d'E/S entre el processador i el mòdul d'E / S, el processador executa un programa que controla tota l'operació d'E/S (programació, transferència de dades i finalització).

Per transferir una dada s'han de dur a terme dues fases:

- 1) Sincronització. Durant la sincronització, el processador, com a responsable de la transferència, executa un programa que mira constantment l'estat del perifèric consultant el registre d'estat del mòdul d'E/S. Aquest programa té un bucle que s'executa contínuament fins que detecta el canvi d'estat i indica que el perifèric està preparat. Aquest mètode de sincronització s'anomena sincronització per enquesta o espera activa. Mentre es porta a terme la sincronització, el processador està dedicat al cent per cent a aquesta tasca i, per tant, no pot atendre altres processos o aplicacions. Si aquesta espera



és molt llarga, pot degradar el nivell de prestacions de tot el sistema. Per tant, és recomanable que les transferències fetes utilitzant aquesta tècnica siguin curtes i ràpides.

2) Intercanvi de la dada. Durant l'intercanvi de la dada, si és una operació de lectura (entrada), el processador llegeix el registre de dades del mòdul d'E/S per recollir la dada enviada pel perifèric, i la guarda en memòria; si és una operació d'escriptura (sortida), el processador agafa la dada que volem enviar a perifèric de la memòria i l'escriu en el registre de dades del mòdul d'E/S.