



Es consideraran vàlides les respostes que siguin justificades.

1. Es té un processador, anomenat Core2007, amb les següents especificacions:

Registres:			Tres formats d'instrucció:		
Nom	No. de bits	Funció	Format 1:		
RI	8	Registre d'instrucció	7	4	3
PC	16	Comptador de programa	referència operand		
SP	16	Apuntador de pila	Format 2:		
A	8	Acumulador	7	0	
B	8	Dades diverses	codi operació		
			referència operand bits 7-0		
			Format 3:		
			7	0	
			codi operació		
			referència operand bits 7-0		
			referència operand bits 15-8		

Instruccions (format  $\rightarrow$ codi operació  $\rightarrow$ efecte):

Format	Codi	Efecte
3	1	$\rightarrow$ Carrega a l'acumulador l'operand adreçat pels bits 15-0 ( $A := Memoria[operand]$ )
3	2	$\rightarrow$ Escriu contingut de l'acumulador a l'adreça donada pels bits 15-0 ( $Memoria[operand] := A$ )
2	3	$\rightarrow$ Inicialitza l'acumulador en el valor del operand ( $A := operand$ )
3	5	$\rightarrow$ Carrega al registre B l'operand adreçat pels bits 15-0 ( $B := Memoria[operand]$ )
3	6	$\rightarrow$ Escriu contingut del registre B a l'adreça donada pels bits 15-0 ( $Memoria[operand] := B$ )
1	7	$\rightarrow$ Multiplica per 2 el contingut de l'acumulador. ( $A = A * 2$ ).
1	9	$\rightarrow$ Decrementa en 1 el registre referit en l'operand (Si l'operand és 0, $A := A - 1$ , si és 1, $B := B - 1$ ).
1	A	$\rightarrow$ Empila (push) el contingut adreçat per l'operand de la instrucció (0 $\rightarrow$ A, 1 $\rightarrow$ PC, 2 $\rightarrow$ B). ( $A0 \equiv SP := SP - 1$ ; $Memoria[SP] := A$ ) ( $A1 \equiv SP := SP - 1$ ; $Memoria[SP] := H(PC)$ ; $SP := SP - 1$ ; $Memoria[SP] := L(PC)$ ; ) On $H(X)$ vol dir la part de bits 15 - 8 de $X$ , i $L(X)$ la de 7 - 0 de $X$
1	B	$\rightarrow$ Desempila (pop) el contingut adreçat per el SP i el posa en el registre indicat per l'operand de la instrucció (0 $\rightarrow$ A, 1 $\rightarrow$ PC, 2 $\rightarrow$ B). ( $B2 \equiv B := Memoria[SP]$ ; $SP := SP + 1$ ; ) ( $B1 \equiv L(PC) := Memoria[SP]$ ; $SP := SP + 1$ ; $H(PC) := Memoria[SP]$ ; $SP := SP + 1$ ; )
3	C	$\rightarrow$ Salta a l'adreça indicada pels bits 15-0. ( $PC := operand$ )
3	D	$\rightarrow$ Si es troba que el contingut del registre B és zero, salta a l'adreça indicada pels bits 15-0. (si $B = 0 \rightarrow PC := operand$ fsi). En cas contrari, no fa res

L'estat inicial del processador, el registre PC està a 2D i SP a FFFF.

- Indiqueu els paràmetres (capacitat màxima, longitud de paraula) de memòria principal adjunts al processador Core2007.
- A partir de la posició de memòria 2D hi ha el següent programa en hexadecimal: 1, 0, 10, 5, 1, 10, D, 3B, 0, 7, 91, C, 33, 0, 2, 2, 10, i en les adreces 1000, 1001 i 1002, contenen respectivament 8, 2 i 0. Feu i descriviu el seguiment del programa i determineu l'estat de registres del Core2007 i de les posicions 1000, 1001 i 1002 quan s'ha executat la instrucció que hi ha a la posició de memòria 3B. Raoneu com s'ha arribat a aquest estat.
- Quan s'està executant per primer cop la instrucció que hi ha a la 33 de l'anterior programa, arriba una interrupció. El servei d'interrupció comença a l'adreça 10, a partir de la qual, es troba el següent contingut: B1, 3, 0, B0, B1. Com afecta la interrupció a l'execució del programa de l'apartat b. Raoneu la resposta.

Entre una memòria de 32K i Core2007, tenim una caché associativa per conjunts de 2 conjunts de 2 línies de 4 paraules cadascuna. Es demana que

- Dissenyeu el format de l'adreça per gestionar la caché.
- Sense tenir en compte la interrupció de l'apartat 1.c, calculeu el nombre de fallides que tindrà l'execució d'un tros del programa de l'apartat 1.b, sabent que la caché aplica la substitució LRU. El tros del programa a examinar és el que va des de l'inici a 2D fins que torna a la instrucció 33 per primer cop (Aquesta darrera no cal incloure-la).

2. Es tenen els següents programes corresponents a quatre processos residents en la cua de preparats en ordre d'arribada A, B, C i D.

<p><b>A:</b>  <b>var</b> <math>T</math>: taula[1..5, 1..5] <b>de</b> enter  <math>i, j</math>: enter;</p> <p><math>i := 1</math>  <b>mentre</b> <math>i \leq 5</math> <b>fer</b>  <math>j := 1</math>  <b>mentre</b> <math>j \leq 5</math> <b>fer</b>  <math>T[i, j] := 1</math>  <math>j := j + 1</math>  <b>fmentre</b>  <math>i := i + 1</math>  <b>fmentre</b></p>	<p><b>B:</b>  <b>var</b> <math>T</math>: taula[1..5, 1..5] <b>de</b> enter  <math>i, j</math>: enter;</p> <p><math>j := 1</math>  <b>mentre</b> <math>j \leq 5</math> <b>fer</b>  <math>i := 1</math>  <b>mentre</b> <math>i \leq 5</math> <b>fer</b>  <math>T[i, j] := 1 - T[i, j]</math>  <math>i := i + 1</math>  <b>fmentre</b>  <math>j := j + 1</math>  <b>fmentre</b></p>
<p><b>C:</b>  <b>var</b> <math>T</math>: taula[1..5, 1..5] <b>de</b> enter  <math>i, j</math>: enter; <math>f</math>: FST</p> <p><math>j := 1</math>  <b>mentre</b> <math>j \leq 5</math> <b>fer</b>  <math>i := 1</math>  <b>mentre</b> <math>i \leq 5</math> <b>fer</b>  <math>T[i, j] := (T[i, j]/2) + 6 * T[i, j]</math>          escriureEnterFST(<math>f, T[i, j]</math>)  <math>i := i + 1</math>  <b>fmentre</b>  <math>j := j + 1</math>  <b>fmentre</b></p>	<p><b>D:</b>  <b>var</b> <math>T</math>: taula[1..5, 1..5] <b>de</b> enter  <math>i, j</math>: enter; <math>f</math>: FST</p> <p><math>i := 1</math>  <b>mentre</b> <math>i \leq 5</math> <b>fer</b>  <math>j := 1</math>  <b>mentre</b> <math>j \leq 5</math> <b>fer</b>          llegirEnterFST(<math>f, T[i, j]</math>)  <math>T[i, j] := 4 * T[i, j] + 30</math>  <math>j := j + 1</math>  <b>fmentre</b>  <math>i := i + 1</math>  <b>fmentre</b></p>

Es té que el cost del cos central de les iteracions pel procés A és 0.8 unitat de temps, 0.4 unitats pel procés B, 5 pel procés C, i 7 pel procés D. Pel cas dels processos C i D el cost donat no té en compte el temps d'accés al disc. El cost de l'increment i inicialització de les variables d'iteració és zero (menyspreable). D'acord als costos donats,

- Calculeu el temps de tongada de cada procés i illustreu l'execució dels processos en els casos en que el planificador usat és Primer en Entrar Primer en ser Servit (FCFS), torn rotatori (Round Robin, quantum=6) i Primer el Treball Més Curt (SJF).
- Calculeu els temps de retorn i temps d'espera de cada procés per cada algorisme de planificació esmentat abans.

Els processos s'executaran en un sistema de memòria paginada de marcs de 5 paraules. La mida d'un enter és d'una paraula. El codi de cada procés ocupa una pàgina. El sistema operatiu assigna 3 pàgines per cada procés i aplica l'algorisme de substitució de pàgines LRU. La taula es guarda a memòria seguint l'ordre  $T[1, 1], T[1, 2], \dots, T[1, N], T[2, 1], T[2, 2], \dots$ . El compilador ha decidit a més que les variables  $i$  i  $j$  siguin registres del processador i per tant no fan cap accés a memòria.

- Determineu el nombre de fallides de pàgina pel programa A i pel programa B.
- Augmentant el nombre de pàgines per programa, s'obtidria alguna millora en els programes A i B?
- Tenint en compte ara el sistema de memòria paginada esmentat i considerant que les pàgines no actives dels programes estan en la memòria d'intercanvi, indiqueu per quins estats de procés passa el programa A des de que està en la cua de preparats. Comenteu breument com s'ha arribat a cadascun dels estats que enumereu.