



Corso di laurea in ingegneria informatica
Esame di sistemi operativi – 21 gennaio 2009 **SOLUZIONI**

1.

Si consideri il programma C seguente (gli "#include" necessari sono omessi):

```
/* programma principale */
1. int main ( ) {
2. int a, b, stato;
3. pid_t pid;
4. for (a = 1; a < 4; a ++ ) {
5. pid = fork ( );
6. if (pid == 0) {
7. if (a == 2) {
8. pid = fork ( );
9. if (pid == 0) {
10. b=3;
11. exit (a + b);
12. } /* end if */
13. wait (&stato);
14. } /* end if */
15. exit (a);
16. } /* end if */
17. if (a == 1) wait (&stato);
18. } /* end for */
19. pid = wait (&stato);
20. } /* end main */
```

a) Si illustri la gerarchia di creazione di tutti i processi (origine in P), figli saranno indicati successivamente con Fi e i nipoti con Fii

Soluzione:

LA creazione dei processi (in totale 5 processi, compreso P):

P

Il processo padre P crea F1, F2 e F3 (nell'ordine); il processo F2 crea un figlio F21.

Il processo P si sincronizza con la terminazione di F1 (riga 17) prima di creare F2 e F3.

Il processo F2 crea F21 e aspetta la sua terminazione (riga 13). L'ordine di creazione di F21 rispetto a F3 non è determinabile.

Il processo P dopo aver creato TUTTI i suoi figli aspetta la terminazione di F2 o F3 prima di terminare.

b) Nelle tabelle seguenti, i simboli F1, F2 e F3 indicano i processi figli creati dal processo padre P (in ordine di creazione).

Si completino le tre tabelle riportate sotto, indicando i valori delle variabili negli istanti di tempo specificati.

Attenzione:

- quando la variabile non esiste (in quanto non esiste il processo), si scriva NE
- quando non si può dire con certezza se la variabile esista o quale ne sia il valore, si scriva U
- si suppone che tutte le chiamate ai servizi di sistema abbiano sempre successo
- il pid del processo padre P è 100, poi il S.O. assegna pid consecutivi in ordine di creazione

Tabella 1. Valore delle variabili nel processo padre P	pid	a	stato
Subito dopo l'esecuzione dell'istruzione 8	U	U	U
Subito dopo l'esecuzione dell'istruzione 19	U	4	U

Tabella 2. Valore delle variabili nel processo figlio F2	pid	a	stato
Subito dopo l'esecuzione dell'istruzione 8	U	2	1
Subito dopo l'esecuzione dell'istruzione 19	U	U	U

Tabella 3. Valore delle variabili nel processo figlio F3	pid	a	stato
Subito dopo l'esecuzione dell'istruzione 15 "exit" eseguita da F1	NE	NE	NE
Subito dopo l'esecuzione dell'istruzione 15 "exit" eseguita da F2	U	U	U

2.

Prima parte

Si consideri un sistema di memoria (memoria + cache) caratterizzato dalle seguenti dimensioni:

- memoria di 4 GigaByte (indirizzata a livello di byte);
- cache di 256 KByte;
- ogni blocco della cache contiene 256 Byte.

Indicare la struttura degli indirizzi per la memoria cache nelle seguenti situazioni:

1. cache a indirizzamento diretto (direct mapped)
2. cache completamente associativa
3. cache set-associativa a 4 vie

Prima parte: soluzione

Memoria di lavoro: indirizzo di 32 bit

Memoria cache: indirizzo di 18 bit

Blocchi: indirizzo di 8 bit

1) Cache a indirizzamento diretto

8 bit per il byte nel blocco

10 bit per l'indice del blocco nella cache

14 bit di etichetta

2) Cache completamente associativa

8 bit per il byte nel blocco

24 bit di etichetta

3) Cache set-associativa a 4 vie

8 bit per il byte nel blocco

8 bit per l'indice del gruppo nella cache

16 bit di etichetta

Seconda parte

Si consideri un sistema di memoria (memoria + cache) caratterizzato dalle seguenti dimensioni:

- memoria di lavoro di 1 KByte, indirizzata a livello di singolo byte;
- cache di 256 Byte;
- ogni blocco della cache contiene 64 Byte.

Considerando la sequenza di richieste alla memoria riportata qui sotto, completare la tabella che illustra il comportamento di una cache **set-associativa a 2 vie** nel rispetto delle seguenti indicazioni:

- Nella colonna "**esito**" riportare **H** (hit) se il blocco richiesto si trova nella cache, **M** (miss) se invece il blocco deve essere caricato dalla memoria.
- Nelle colonne "**dati**" deve essere riportato il **numero del blocco della memoria** (la memoria di 1024 Byte comprende 16 blocchi di 64 Byte) che si trova nel corrispondente blocco della cache. Si noti che questi valori sono riportati come numeri decimali (base dieci), mentre le etichette sono scritte in binario. Per questo motivo l'indirizzo 00 0001 0010 individua un byte compreso nel blocco $0000_{\text{due}} = 0_{\text{dieci}}$ (che quindi ha come etichetta il valore binario 000).
- Nella colonna "**azione**" deve essere indicato il **blocco cui si accede** (in caso di successo, H) o il blocco in cui vengono caricati i dati della memoria (in caso di fallimento, M).
- Nella cache ci sono quattro blocchi denotati dalle lettere **A**, **B**, **C** e **D**, che sono organizzati in due insiemi: si ipotizzi che i blocchi **A** e **B** siano compresi nell'**insieme 0** e che i blocchi **C** e **D** facciano invece parte dell'**insieme 1**.
- La politica di sostituzione adottata nella cache è quella LRU (Least Recently Used).

Note: dei 10 bit di indirizzo, 6 servono per individuare il byte nel blocco. Nella cache ci sono quattro blocchi organizzati in due gruppi, quindi il 7° bit (da destra) indica il gruppo, mentre i restanti 3 bit formano l'etichetta.

Passo	Indirizzo richiesto	Esito	Blocco A			Blocco B			Blocco C			Blocco D			Azione
			Valido	Etichetta	Dati	Valido	Etichetta	Dati	Valido	Etichetta	Dati	Valido	Etichetta	Dati	
0			1	100	8	1	000	0	1	000	1	1	001	3	Situazione iniziale
1.	00 0001 0010	H	1	100	8	1	000	0	1	000	1	1	001	3	Accesso a B
2.	10 0001 0100	H	1	100	8	1	000	0	1	000	1	1	001	3	Accesso ad A
3.	00 1111 1111	H	1	100	8	1	000	0	1	000	1	1	001	3	Accesso a D
4.	11 1000 1010	M	1	100	8	1	111	14	1	000	1	1	001	3	Carica blocco 14 in B
5.	11 1101 0010	M	1	100	8	1	111	14	1	111	15	1	001	3	Carica blocco 15 in C
6.	01 0100 0110	M	1	100	8	1	111	14	1	111	15	1	010	5	Carica blocco 5 in D

3.

Illustrare le 4 condizioni necessarie perché si verifichi un deadlock e proporre delle soluzioni per prevenire ciascuna di esse.

Cfr. capitolo 8 del libro di testo