Ingegneria Informatica – Sistemi operativi – 16 giugno 2009 – DOMANDE

COGNOME____SOLUZIONI_____NOME_____MATRICOLA_____

1.
Spiegare i principi di località temporale e località spaziale e come i dispositivi di memoria di un calcolatore possono essere organizzati in modo da sfruttare tali principi.

2.
Dato il seguente codice:

```
1. int main ( ) {
2. int a, b=3, pid;
3. for (a = 5; a > 3; a--) {
4.     pid = fork ( );
5.     if (pid != 0) {
6.       if (a == 4) {
7.         pid = fork ( );
8.         if (pid == 0) {
9.             b=a;
10.            exit (b-1);
11.        } /* end if */
12.        wait (&b);
13.      } /* end if */
14.      b = b - 1;
15.   } /* end if */
16.   if (pid <= 200) exit();
17. } /* end for */
18. wait (&b);
19.} /*end main */
```

Si completino le tabelle nel foglio delle risposte, 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 è 201, poi il S.O. assegna pid consecutivi in ordine di creazione
- "1° esec. istr. X" significa: prima esecuzione dell'istruzione X

3.
Si consideri il seguente algoritmo di scheduling con preemption, basato su priorità variabili dinamicamente. I numeri di priorità maggiori corrispondono a priorità più alte. Quando un processo attende la CPU (nella coda dei processi pronti), la sua priorità varia a un tasso α, mentre quando è in esecuzione, la sua priorità varia a un tasso β. Quando un processo entra nella coda dei processi pronti, il sistema gli attribuisce priorità 0. I tassi α e β possono essere impostati in modo da creare algoritmi di scheduling diversi.

a. Descrivere l'algoritmo che risulta da $\beta > \alpha > 0$.
b. Descrivere l'algoritmo che risulta da $\alpha < \beta < 0$.

UNIVERSITÀ DEGLI STUDI DI BERGAMO

Ingegneria Informatica – Sistemi operativi – 16 giugno 2009 – RISPOSTE

COGNOME__ SOLUZIONI ____NOME_____MATRICOLA_____

RISPOSTA 1:
(in breve)
Principio di localita' temporale: e' probabile che un dato utilizzato (o un'istruzione eseguita) a un certo istante venga riutilizzato (o rieseguita) in un istante nell'immediato futuro.
Principio di localita' spaziale: e' probabile che se un dato viene utilizzato (o un'istruzione viene eseguita) anche i dati (o le istruzioni) salvati nelle zone di memoria circostanti vengano utilizzati (o eseguite).
Le gerarchie di memoria ottimizzano la loro performance sfruttando entrambi questi principi: quello temporale implementando ad esempio una strategia LRU (Least Recently Used) per decidere quale dato eliminare da un dispositivo di memoria pieno, quello spaziale spostando da un livello della gerarchia all'altro non singoli dati, ma blocchi di dati.

Ingegneria Informatica – Sistemi operativi – 16 giugno 2009 – RISPOSTE

COGNOME____ **SOLUZIONI** **NOME**_____ **MATRICOLA**_____

RISPOSTA 2:

Processo 201	variabile a	variabile b	variabile pid
subito dopo 1° esec. istr. 4	5	3	202
subito dopo 2° esec. istr. 4	4	2	203
subito dopo l'istruzione 12	4	3	203

Processo 202	variabile a	variabile b	variabile pid
subito dopo 1° esec. istr. 4	5	3	0
subito dopo 2° esec. istr. 4	NE	NE	NE
subito dopo l'istruzione 12	NE	NE	NE

Processo 203	variabile a	variabile b	variabile pid
subito dopo 1° esec. istr. 4	NE	NE	NE
subito dopo 2° esec. istr. 4	4	2	0
subito dopo l'istruzione 12	NE	NE	NE

Processo 204	variabile a	variabile b	variabile pid
subito dopo 1° esec. istr. 4	NE	NE	NE
subito dopo 2° esec. istr. 4	NE	NE	NE
subito dopo l'istruzione 12	NE	NE	NE

Ingegneria Informatica – Sistemi operativi – 16 giugno 2009 – RISPOSTE

COGNOME___ SOLUZIONI ___NOME_____MATRICOLA_____

RISPOSTA 3:
α: tasso di crescita (se positivo) o diminuzione (se negativo) della priorita' dei processi pronti.
β: tasso di crescita (se positivo) o diminuzione (se negativo) della priorita' del processo in esecuzione.
Priorita' di un processo entrato nella coda dei processi pronti: 0 (zero).
A indici di priorita' piu' grandi corrispondono priorita' maggiori (quindi un processo con indice 5 ha priorita' superiore a un processo con priorita' 3).

1) $\beta > \alpha > 0$, significa che le priorita' dei processi pronti crescono col tempo, ma piu' lentamente della priorita' del processo in esecuzione. Questo vuol dire che non c'e' mai preemption e che una volta il processo in esecuzione ha terminato, entra in esecuzione il processo che sta nella coda dei pronti da piu' tempo. La strategia e' FCFS (First Come First Served)

2) $\alpha < \beta < 0$, significa che le priorita' dei processi pronti decrescono col tempo, e piu' velocemente della priorita' del processo in esecuzione. Questo vuol dire che un processo appena arrivato nella coda dei processi pronti con priorita' 0 ha una priorita' maggiore di quella del processo in esecuzione. Subita la preemption, il processo che era prima in esecuzione finisce nella coda dei processi pronti, dove comunque mantiene la priorita' piu' elevata, visto che gli altri hanno visto la propria priorita' diminuire piu' velocemente. A meno di nuovi arrivi, quindi, sara' il prossimo a entrare (o meglio, tornare) in esecuzione. La strategia e' LCFS (Last Come First Served)