Sistemi Operativi
30 gennaio 2012
Compitino 1 B

Si risponda ai seguenti quesiti, giustificando le risposte.

1. (a) Si descrivano brevemente le componenti di un sistema di calcolo, fornendo uno schema che mostri la stratificazione del sistema, evidenziando la collocazione del sistema operativo.

   (b) Si descrivano gli obiettivi generali di un sistema operativo.

   Risposta:

   (a) Le componenti di un sistema di calcolo sono le seguenti:

   - al livello più basso c'è l'hardware, che fornisce le risorse computazionali di base: (CPU, memoria, dispositivi di I/O);
   - il sistema operativo controlla e coordina l'uso dell'hardware tra i vari programmi applicativi per i diversi utenti;
   - sl sistema operativo vero e proprio fanno affidamento altri programmi di sistema (cioè indipendenti dall'applicazione, come compilatori, editor, ecc., forniti con il sistema operativo stesso);
   - i programmi applicativi definiscono il modo in cui le risorse del sistema sono usate per risolvere i problemi computazionali dell'utente (database, programmi di produttività personale, ecc.);
   - al livello più alto vi sono gli utenti (persone, altri calcolatori).

   (b) In generale, l'obiettivo principale di un sistema operativo è quello di realizzare una macchina astratta, ovvero, implementare funzionalità di alto livello, nascondendo i dettagli di basso livello. Ciò permette di:

   - eseguire i programmi utente e rendere più facile la soluzione dei problemi dell'utente;
   - rendere il sistema di calcolo più facile da utilizzare e programmare;
   - gestire le risorse del sistema, utilizzando l'hardware del calcolatore in modo sicuro ed efficiente.

2. Si descriva la system call Unix per la creazione di un nuovo processo; in particolare, si parli delle modalità di allocazione dello spazio indirizzi del nuovo processo.

   Risposta: In Unix la chiamata di sistema per la creazione di un nuovo processo è la fork. Essa crea il processo figlio duplicando in memoria il processo padre e restituendo a quest'ultimo il PID del figlio. Il processo figlio invece si vede restituire come PID 0: ciò permette, a livello di codice C di distinguere fra i due processi e, eventualmente, di diversificarne il comportamento. In particolare, per quanto riguarda le modalità di allocazione dello spazio di indirizzi del nuovo processo, la fork alloca una nuova process structure per il processo figlio:

   - nuove tabelle per la gestione della memoria virtuale;
   - nuova memoria viene allocata per i segmenti dati e stack;
   - i segmenti dati, stack e la user structure vengono copiati: quindi vengono preservati i file aperti, UID e GID, la gestione dei segnali, ecc.
   - il text segment viene condiviso, puntando alla stessa text structure.

   La variante vfork invece non copia i segmenti data e stack, che vengono condivisi fra padre e figlio:

   - il system data segment e la process structure vengono creati;
   - il processo padre rimane sospeso finché il figlio non termina o esegue una execve;
   - il processo padre usa vfork per produrre il figlio, che usa execve per cambiare immediatamente lo spazio di indirizzamento virtuale: non è necessario copiare data e stack segment del padre.

   Questo schema è quello comunemente usato da una shell per eseguire un comando e attendere il suo completamento. È efficiente per processi grandi (con risparmio di tempo di CPU), ma è potenzialmente pericolosa, in quanto le modifiche fatte dal processo figlio prima della execve si riflettono sullo spazio indirizzi del padre.

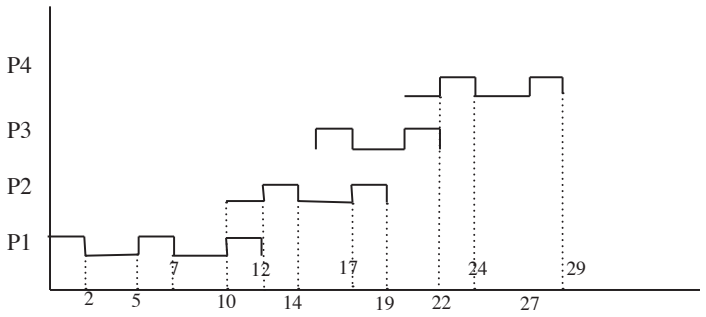
Sistemi Operativi
30 gennaio 2012
Compitino 1 B

3. I processi P_1, P_2, P_3, P_4 alternano 2ms di elaborazione sulla CPU a 3ms di operazioni di I/O. I tempi di arrivo e le richieste totali di CPU di ciascun processo sono specificate nella seguente tabella:

	$arr.time$	CPU_{tot}
P_1	0	6
P_2	10	4
P_3	15	4
P_4	20	4

Lo scheduling della CPU è Round Robin con quanto pari a 2ms. Si trascurino i tempi di context switch. Se due processi arrivano in coda ready nello stesso istante, quello con indice minore precede l'altro. Si consideri l'intervallo di tempo necessario per completare i 4 processi e si calcoli la percentuale di utilizzo della CPU in quell'arco di tempo.

Risposta: Il diagramma relativo all'esecuzione dei quattro processi è il seguente:



Quindi la percentuale di utilizzo della CPU è data da $\frac{18}{29} = 62,07\%$. Nel caso in cui si consideri anche un ulteriore ciclo di I/O finale, la percentuale di utilizzo diventa $\frac{18}{32} = 56,25\%$.

4. (a) Si descriva il funzionamento dell'istruzione assembler Test-and-Set-Lock $TSL\ RX, LOCK$.
   (b) Utilizzando l'istruzione *TSL*, si fornisca un'implementazione in pseudo-codice assembler delle procedure *enter-region* and *leave-region*, che precedono e seguono la sezione critica.

   Risposta:

   (a) L'istruzione assembler Test-and-Set-Lock controlla e modifica il contenuto di una parola atomicamente. Le due operazioni devono essere implementate in modo atomico (bloccando il bus della memoria) in assembler per evitare che si verifichino delle race condition. Quindi *TSL RX,LOCK* copia il contenuto della cella *LOCK* nel registro *RX* e poi imposta la cella *LOCK* ad un valore uguale a 0.

   (b) L'istruzione assembler *TSL RX,LOCK* può essere utilizzata come segue per implementare le procedure di entrata/uscita da una regione critica:

```
enter_region:
    TSL REGISTER,LOCK
    CMP REGISTER,#0
    JNE enter_region
    RET

leave_region:
    MOVE LOCK,#0
    RET
```

5. Si consideri la seguente situazione, dove P_0, P_1, P_2 sono tre processi in esecuzione, C è la matrice delle risorse correntemente allocate, Max è la matrice del numero massimo di risorse da assegnare

ad ogni processo e A è il vettore delle risorse disponibili:

	C			Max		
	A	B	C	A	B	C
P_0	0	0	0	3	3	2
P_1	0	2	1	1	4	2
P_2	0	3	0	2	3	2

Available (A)		
A	B	C
3	x	1

(a) Calcolare la matrice R delle richieste.

(b) Determinare il minimo valore di x tale che il sistema si trovi in uno stato sicuro.

Risposta:

(a) La matrice R delle richieste è data dalla differenza $Max - C$:

R		
A	B	C
3	3	2
1	2	1
2	0	2

(b) Con una risorsa disponibile di tipo C è possibile soddisfare soltanto le richieste di P_1 se x vale almeno 2. Infatti, assumendo $x = 2$, posso eseguire P_1 dato che $R_1 \leq A$. Terminato P_1, il nuovo valore di A diventa $(3, 4, 2)$. Quindi è possibile eseguire P_0, liberando le risorse ad esso allocate. Il nuovo valore di A rimane $(3, 4, 2)$ e non resta che eseguire P_2 per completare la sequenza di esecuzione sicura.

6. Un sistema che usa l'algoritmo del banchiere per l'allocazione delle risorse, contiene n_1 istanze della risorsa R_1 e n_2 istanze della risorsa R_2, e 3 processi, P_1, P_2, P_3. Le risorse non allocate sono $(1, 1)$. Si fanno le seguenti osservazioni relativamente al sistema:

   1. Se il processo P_1 fa una richiesta $(1, 0)$, seguita da $(0, 1)$, la prima viene soddisfatta, ma non la seconda.
   2. Se invece il processo P_1 fa solo la richiesta $(0, 1)$, allora questa viene soddisfatta.

   Si trovi un possibile insieme di valori per le allocazioni correnti e per le richieste massime da parte dei processi, in modo che il comportamento dell'algoritmo del banchiere sia coerente con le osservazioni sopra.

   Risposta: Una situazione compatibile con i vincoli descritti è la seguente:

	C		Max	
	R_1	R_2	R_1	R_2
P_1	0	0	2	2
P_2	1	0	1	1
P_3	0	1	1	1

   dove $E = (4, 4)$. Quindi la matrice R è la seguente:

	R	
	R_1	R_2
P_1	1	2
P_2	0	1
P_3	1	0

   Se la richiesta (1,0) da parte di P_1 viene accettata dal sistema, allora $C[1] = (1, 0)$ e $A = (0, 1)$. Il sistema si trova in uno stato sicuro dato che può schedulare l'esecuzione di P_2, P_3 e P_1 nell'ordine elencato. Se accetta anche la richiesta (0,1) invece $C[1] = (1, 1)$, $A = (0, 0)$ e $R_1 = (1, 1)$: il sistema si trova in stallo (non esiste nessuna riga della matrice delle richieste minore od uguale a A).

Invece la sola richiesta (0,1) da parte di P_1 può essere accettata in quanto $C[1] = (0, 1)$, $A = (1, 0)$ e la matrice R è la seguente:

	R	
	R_1	R_2
P_1	2	1
P_2	0	1
P_3	1	0

Quindi il sistema è in uno stato sicuro dato che può schedulare l'esecuzione di P_3, P_2 e P_1 nell'ordine indicato.