

# Sistemi Operativi

## 30 gennaio 2012

### Compito

Si risponda ai seguenti quesiti, giustificando le risposte.

- 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.
  - Si descrivano gli obiettivi generali di un sistema operativo.

**Risposta:**

- 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;
    - il 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).
  - 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.
- 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.

- I processi  $P_1$ ,  $P_2$ ,  $P_3$ ,  $P_4$  alternano 1ms di elaborazione sulla CPU a 4ms di operazioni di I/O. I tempi di arrivo e le richieste totali di CPU di ciascun processo sono specificate nella seguente tabella:

# Sistemi Operativi

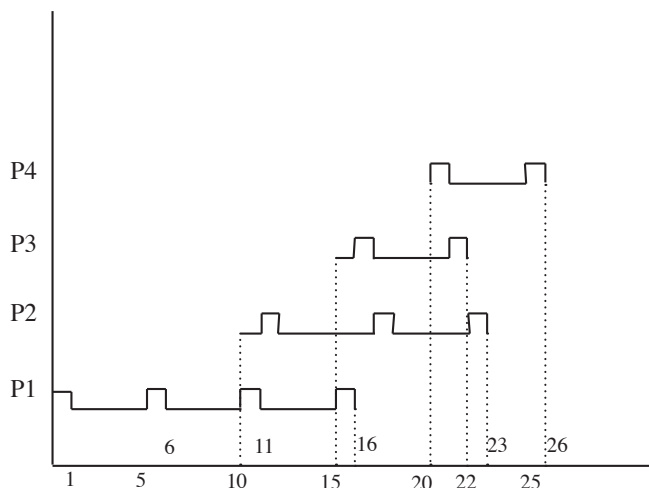
## 30 gennaio 2012

### Compito

	<i>arr.time</i>	<i>CPU<sub>tot</sub></i>
$P_1$	0	4
$P_2$	10	3
$P_3$	15	2
$P_4$	20	2

Lo scheduling della CPU è Round Robin con quanto pari a 1ms. 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  $\frac{11}{26} = 42,3\%$ . Nel caso in cui si consideri anche un ulteriore ciclo di I/O finale, la percentuale di utilizzo diventa  $\frac{11}{30} = 36,7\%$ .

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* e *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

# Sistemi Operativi

## 30 gennaio 2012

### Compito

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

	<u>C</u>				<u>Max</u>		
	A	B	C	A	B	C	
$P_0$	0	3	0	2	4	2	
$P_1$	1	2	0	2	5	1	
$P_2$	0	0	0	2	4	3	

$\underline{Available (A)}$		
$A$	$B$	$C$
1	$x$	3

- (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  $Max - C$ :

	<u>R</u>	
A	B	C
2	1	2
1	3	1
2	4	3

- (b) Con una risorsa disponibile di tipo  $A$  è possibile soddisfare soltanto le richieste di  $P_1$  se  $x$  vale almeno 3. Infatti, assumendo  $x = 3$ , posso eseguire  $P_1$  dato che  $R_1 \leq A$ . Terminato  $P_1$ , il nuovo valore di  $A$  diventa  $(2, 5, 3)$ . Quindi è possibile eseguire  $P_0$ , liberando le risorse ad esso allocate. Il nuovo valore di  $A$  diventa  $(2, 8, 3)$  e non resta che eseguire  $P_2$  per completare la sequenza di esecuzione sicura.
6. Si illustrino brevemente i passi eseguiti dai driver dei dispositivi. Cosa vuol dire che *un driver deve essere rientrante*?

**Risposta:** I passi eseguiti dai driver dei dispositivi sono i seguenti:

1. Controllare i parametri passati.
2. Accodare le richieste in una coda di operazioni (soggette a scheduling).
3. Eseguire le operazioni, accedendo al controller.
4. Passare il processo in modo wait (I/O interrupt-driven), o attendere la fine dell'operazione in busy-wait.
5. Controllare lo stato dell'operazione nel controller.
6. Restituire il risultato.

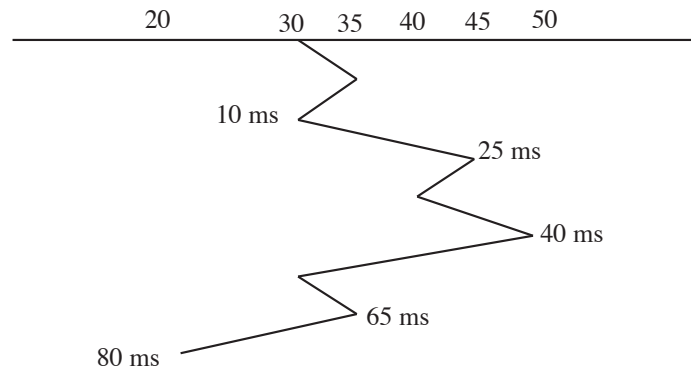
Un driver si dice rientrante se, mentre viene eseguito, può essere lanciata una nuova esecuzione del suo codice.

7. Si consideri un disco gestito con politica SSTF. Inizialmente la testina è posizionata sul cilindro 30; lo spostamento ad una traccia adiacente richiede 1 ms. Al driver di tale disco arrivano richieste per i cilindri 45, 30, 20, 50, 35, rispettivamente agli istanti 0 ms, 5 ms, 20 ms, 30 ms, 60 ms. Si trascuri il tempo di latenza.
1. In quale ordine vengono servite le richieste?
  2. Il tempo di attesa di una richiesta è il tempo che intercorre dal momento in cui è sottoposta al driver a quando viene effettivamente servita. Qual è il tempo di attesa medio per le cinque richieste in oggetto?

**Risposta:**

1. Le richieste vengono servite nell'ordine 30, 45, 50, 35, 20:

**Sistemi Operativi**  
**30 gennaio 2012**  
Compito



2. Il tempo di attesa medio per le cinque richieste in oggetto è
- $$\frac{(10-5)+(25-0)+(40-30)+(65-60)+(80-20)}{5} = \frac{5+25+10+5+60}{5} = \frac{105}{5} = 21 \text{ ms.}$$