



Corso di Laurea in Ingegneria Informatica Sistemi Operativi - Appello 29 Gennaio 2013

1. Spiegare la differenza tra *starvation* e *deadlock* di processi. [max 4 pt]
2. Descrivere il meccanismo di *paginazione gerarchica* della memoria ed illustrare la traduzione degli indirizzi logici in indirizzi fisici con 2 livelli di paginazione. [max 8 pt]
3. Si consideri l'algoritmo di schedulazione *Round Robin (RR)* della CPU ed il seguente insieme di processi, con la durata del "picco" della CPU espressa in millisecondi: [max 8 pt]

Processi	Tempo di arrivo	Durata del picco in
P1	0	24
P2	0	3
P3	15	3
P4	18	15
P5	0	1

Si assuma che i processi arrivino nell'ordine P1, P2,...,P5, tutti all'istante 0 fatta eccezione per P3, che arriva all'istante 15, e per P4, che arriva all'istante 18. Si assuma inoltre che i processi vengano eseguiti con una schedulazione RR con *quanto di tempo q = 10*.

- Disegnare il diagramma di Gantt che illustri l'esecuzione di questi processi.
 - Calcolare il *tempo di attesa medio*.
 - Calcolare il *tempo di completamento (turnaround time)* di ogni processo.
4. *Quesito riservato agli studenti di Informatica II (21013+23014)*: Uno sportello di una piccola banca offre servizi di *prelievo/versamento* ai clienti intestatari di (uno ed un solo!) conto corrente (CC) individuato univocamente da un codice intero. Il prelievo può essere eseguito soltanto dall'intestatario del CC e il richiedente aspetta nel caso nel conto non vi sia un importo sufficiente. Il versamento può essere eseguito da un cliente della banca anche diverso dall'intestatario del CC. In particolare, i clienti vengono classificati in due categorie:

- Clienti Standard: sono i clienti il cui saldo attuale è minore di 50.000 €;
- Clienti Vip: sono i clienti il cui saldo attuale è maggiore o uguale di 50.000 €.

Ogni cliente può cambiare categoria dinamicamente (in base, cioè, al saldo attuale del CC).

Si progetti una politica di gestione dello sportello che soddisfi i seguenti vincoli di priorità:

- a. Le operazioni di versamento hanno la priorità su quelle di prelievo;
- b. In una operazione (prelievo o versamento), i Vip hanno priorità sui clienti Standard.

Si assuma che ogni cliente effettua periodicamente un'operazione e si completi la definizione della classe Sportello mediante il *meccanismo di sincronizzazione a livello del linguaggio Java (modificatore `synchronized` ed i metodi `wait()/notify()`)*. [max 10 pt]

```
public class Sportello{
    private final int N; //costante che esprime il numero massimo di CC
    private final int VIP=0; //TIPO VIP
    private final int STA=1; //TIPO CLIENTE STANDARD
    private final int soglia=50000;
    private int[] CC; //CONTI CORRENTI: CC[i] è il saldo del conto corrente di codice intero i
    ... <completare>
    public Sportello(nc) { //Costruttore:
        N=nc; CC=new int[N]; for(i=0; i<N; i++) CC[i]=0; //All'inizio in ciascun CC non ci sono soldi
        ... <completare> }
    private int tipo(int conto){ // determina il tipo di conto
        if (CC[conto] < soglia) return STA; else return VIP; }
    public void prelievo (int somma, int mioconto) throws InterruptedException { ... <completare> }
    public void versamento (int somma, int mioconto, int conto) throws InterruptedException { ... <completare> }
}
```

5. *Quesito per gli studenti del C.I. di Reti di calcolatori e Sistemi operativi*: [max 10 pt]
Spiegare (con un esempio) la semantica delle chiamate di sistema *wait()* ed *exit()* per il controllo dei processi in un sistema operativo Unix-like, specie nel loro uso combinato.

SOLUZIONE:

1. Vedi libro di testo/lucidi della lezione relativa.
2. Vedi libro di testo/lucidi della lezione relativa.
3. **Diagramma di Gantt**

P1	P2	P5	P1	P3	P4	P1	P4	
0	10	13	14	24	27	37	41	46

Tempo di attesa medio

Sia T_i il tempo di attesa del processo P_i , per $i=1,\dots,5$. Si ha:

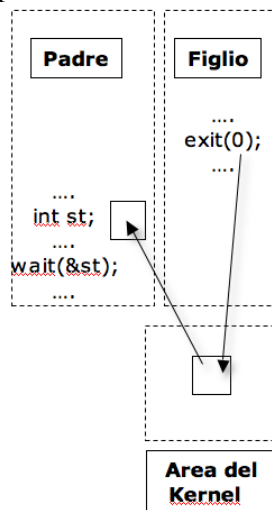
- $T_1 = 4+13=17$
- $T_2 = 10$
- $T_3 = 9$
- $T_4 = 9+4=13$
- $T_5 = 13$

Tempo attesa medio = $(T_1+T_2+T_3+T_4+T_5)/5=62/5= 12,4$ ms

Tempo di completamento C_i di ogni processo

$C_1 = 41, C_2=13, C_3=27-15= 12, C_4=46-18= 28, C_5= 14$

4. Omessa. Vedi appello Set. 2012
5. L'uso combinato delle chiamate di sistema `wait` e `exit` consente di sincronizzare il padre sulla terminazione dei processi figli.
wait: consente al padre di raccogliere lo stato di terminazione dei figli e restituisce l'ID del processo figlio che è terminato. Se i figli non sono ancora terminati, il kernel sospende il processo padre finché uno dei figli non è terminato.
exit: termina il processo che la invoca e passa lo stato di uscita (un intero) al kernel. Tale valore è disponibile al processo padre attraverso la chiamata di sistema `wait`.



Vedi Cap. 4, Sez. 4.3 del libro adottato per maggiori dettagli.

Risultati

N.B.: La prova orale è fissata per giorno 7 Febbraio alle 10.00.

Matricola	Cognome	Nome	Voto prova scritta
1015591	ABAZI	OLTION	22
1020570	BONOMI	ENRICO	24
1021077	CARISSONI	MARCO	28
1021202	CIGLIANO	ENRICO	26
1009930	DE ANGELIS	DOMENICO	20
1020653	FINAZZI	SOLIDEA	14
1021292	MALEGORI	MATTEO	30
1020627	MANTEGAZZA	ANDREA	15
1014416	PETERLIN	GIORGIO	18
1014604	RAVASIO	FRANCESCO	14
1021063	RAVASIO	MARCO	15
1020890	SONZOGNI	MAURO	25
1020642	TAHIRAJ	ERDIT	19