



Corso di Laurea in Ingegneria Informatica Sistemi Operativi App. 5 Settembre 2012

1. Che cosa si intende per funzionamento *dual-mode* del sistema operativo? [max 3 pt]
2. Si descriva almeno un' istruzione hardware speciale per la sincronizzazione e si mostri l'uso per implementare il protocollo di accesso alla sezione critica. [max 7 pt]
3. Nell'ambito degli algoritmi per la *sostituzione della pagina* per l'implementazione della memoria virtuale, si consideri la seguente stringa di riferimento di numeri di pagina:
2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2 [max 10 pt]

Rispondere ai seguenti quesiti:

- a. Quanti *page fault* vengono generati con gli algoritmi **LRU**, **FIFO**, e **Ottimo**, con uno, due, tre, o quattro frame? Si assuma che i frame siano inizialmente vuoti (conteggiare, quindi, anche i *page fault* per il caricamento iniziale delle pagine).
 - b. In cosa consiste l'algoritmo **FIFO second chance** (*seconda opportunità*)?
 - c. Cos'è il fenomeno chiamato *anomalia di Belady*?
 - d. Quali algoritmi tra **LRU**, **FIFO**, **FIFO second chance**, e **Ottimo** soffrono dell'*anomalia di Belady*?
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 dei lock* e le *variabili condizione*. [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]
- Nell'ambito dei metodi di sincronizzazione dei processi, si consideri il *problema dei lettori/scrittori con priorità agli scrittori*: nessun scrittore deve essere tenuto in attesa, a meno che i lettori hanno già ottenuto l'accesso alla base di dati condivisa. Si definisca, con un pseudolinguaggio di programmazione, la classe Database contenente i metodi per acquisire i lock di lettura e di scrittura facendo uso dei *semafori*.

SOLUZIONE:

- 1. Vedi libro di testo/lucidi della lezione relativa.
- 2. Vedi libro di testo/lucidi della lezione relativa.
- 3.a. Si consideri la seguente stringa di riferimento di numeri di pagina:
2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2

Tabella finale:

<u>N. FRAME</u>	<u>LRU</u>	<u>FIFO</u>	<u>Ottimo</u>
1	12	12	12
2	9	9	8
3	8	9	6
4	5	6	5

Passaggi intermedi (page fault in "rosso"):

FIFO:

4 frame: 2 1 1
 0 0 2
 3 3 3
 4 4 4

3 frame: 2 4 4 4 3 3 3
 0 0 2 2 2 1 1
 3 3 3 0 0 0 2

2 frame: 2 3 3 2 2 0 0 1
 0 0 4 4 3 3 2 2

1 frame: page fault= num. totale di pagine richieste = 12

LRU:

4 frame: 2 2
 0 0
 3 3
 4 1

3 frame: 2 4 4 4 0 1
 0 0 0 3 3 3
 3 3 2 2 2 2

2 frame: 2 3 4 4 3 3 3 1
 0 0 0 2 2 0 2 2

1 frame: page fault= num. totale di pagine richieste = 12

OTTIMO:

4 frame: 2 2
0 1
3 3
4 4

3 frame: 2 2 2 2
0 4 0 1
3 3 3 3

2 frame: 2 3 3 3 3 2 2
0 0 4 2 0 0 1

1 frame: page fault= num. totale di pagine richieste = 12

3.b. Vedi Sez. 10.4.5.2 del libro adottato.

3.c. Per anomalia di Belady si intende il fenomeno per cui, nonostante si incrementi la memoria fisica disponibile e quindi il numero di frame totali, non è detto che i page fault diminuiscano, anzi in alcuni casi possono addirittura aumentare (“anomalia”).

3.d. In generale, gli algoritmi FIFO e FIFO second-chance soffrono dell’anomalia di Belady.

4. Esiste una scala di priorità:

1. Versamento vip
2. Versamento standard
3. Prelievo vip
4. Prelievo standard

Per la sospensione dei thread basta definire 4 condition (una per ogni livello di priorità) e 4 contatori dei thread sospesi per ogni coda.

Possibile traccia:

```
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
private Lock lock= new ReentrantLock();
private Condition VV= lock.newCondition(); //coda Vip che vogliono versare
private Condition VS= lock.newCondition(); //coda Standard che vogliono versare
private Condition PV= lock.newCondition(); //coda Vip che vogliono prelevare
private Condition PS= lock.newCondition(); //coda Standard ch vogliono prelevare
private int sospVV, sospVS;
private int sospPV, sospPS;
//Costruttore:
public monitor( int nc) {
    int i; N=nc; CC=new int[N]; for(i=0; i<N; i++) CC[i]=0;
    sospPV=0;
    sospPS=0;
    sospVV=0;
    sospVS=0;
}
}
```

```

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 {
    lock.lock();
    try { while ( CC[mioconto]<somma || sospVV>0 || sospVS>0 ||
        ((sospPV > 0) && (tipo(mioconto)==STA))
        )
        if (tipo(mioconto) == VIP){
            sospPV++;
            PV.await();
            sospPV--;
        }
        else{
            sospPS++;
            PS.await();
            sospPS--;
        }
        CC[mioconto]-=somma;
        //signal in ordine di priorità:
        if (sospVV>0) VV.signalAll();
        else if (sospVS>0) VS.signalAll();
        else if (sospPV>0) PV.signalAll();
        else if (sospPS>0) PS.signalAll();
    } finally{ lock.unlock();}
    return;
}

public void versamento (int somma, int mioconto, int conto) throws InterruptedException {
    lock.lock();
    try{ while ( CC[mioconto]<somma ||
        ((sospVV > 0) && (tipo(mioconto)==STA)))
        if (tipo(mioconto) == STA){
            sospVS++;
            VS.await();
            sospVS--;
        }
        else { //VIP con meno di "somma" sul conto
            sospVV++;
            VV.await();
            sospVV--;
        }
        CC[mioconto]-=somma;
        CC[conto]+=somma;
        //signal in ordine di priorità:
        if (sospVV>0) VV.signalAll();
        else if (sospVS>0) VS.signalAll();
        else if (sospPV>0) PV.signalAll();
        else if (sospPS>0) PS.signalAll();
    } finally{ lock.unlock();}
    return;
}
}

```

Risultati prova scritta

CARISSONI MARCO	10
CHILLURA STEFANO 2011 / 2012	20
DELIGIOS SILVIO 2011 / 2012	20
FINAZZI SOLIDEA 2011 / 2012	11
GHISLENI STEFANO 2011 / 2012	28
GIUDICI LORENZO	30
HU ALESSANDRO	18
MARZIALI MICHEL 2011 / 2012	20
MILESI MATTEO 2011 / 2012	24
QUATTRI ANDREA	22
PEZZOTTA YURI RICCARDO 2008 / 2009	
PIZZOLANTE ADRIANO	19
RATTI DANIELE 2011 / 2012	24
ROVETTA NICOLA 2009 / 2010	11
SURDU VASILE 2008 / 2009	26