



Corso di Laurea in Ingegneria Informatica Sistemi Operativi - Appello del 9 Luglio 2013

Cognome: _____ Nome: _____ Matricola: _____

1. Descrivere in modo sintetico i possibili meccanismi utilizzati per implementare la comunicazione tra processi. [4 pt.]
2. Nell'ambito della gestione della memoria virtuale:
 - Descrivere la paginazione su richiesta [3 pt.]
 - Descrivere l'algoritmo LRU per la sostituzione delle pagine [3 pt.]
 - Descrivere gli algoritmi di allocazione dei frame ai processi [2 pt.]
3. Si consideri l'algoritmo di schedulazione Round Robin con quanto di tempo pari a 20ms:

	Durata del picco	Tempo di arrivo
P ₁	40 ms	0 ms
P ₂	90 ms	25 ms
P ₃	50 ms	0 ms
P ₄	25 ms	55 ms
P ₅	10 ms	0 ms

- Descrivere l'algoritmo; [3 pt]
 - Definire il diagramma di Gant della schedulazione evidenziando tutti i punti in cui lo schedulatore deve prendere una decisione. Calcolare sia il tempo di attesa medio che il tempo di completamento di ciascun processo (il token è assegnato ai processi secondo un ordine crescente dell'indice, i.e. P₁, P₂, P₃, P₄, P₅, P₁, P₂, P₃, P₄, P₅,...). [4 pt]
 - Descrivere la variante implementata con code a più livelli e feedback. [1 pt]
4. Una fabbrica inscatola palline da tennis. Le palline sono immagazzinate in un grande contenitore, dal quale sono prelevate per essere inscatolate in confezioni da quattro. Quattro bracci robotici raccolgono una pallina alla volta dal contenitore per stampare il logo della fabbrica, dopo di che mettono la pallina all'interno della confezione. Valgono le seguenti regole:
- Un solo braccio per volta può prelevare le palline dal contenitore e disporre le palline nella confezione (accesso in mutua esclusione a contenitore e confezione).
 - I bracci robotici, dopo aver sistemato la pallina, attendono che la confezione sia piena prima di ricominciare il lavoro.
 - Quando il contenitore è vuoto, è necessario notificare un rifornitore, che si occuperà di caricare nuove palline.
- Definire tutte le classi necessarie per la risoluzione del problema (*Fabbrica*, *Braccio*, *Fornitore*, *Confezione* e *Confezionatore*). La sincronizzazione deve essere implementata utilizzando solo i meccanismi dei semafori e delle barriere (si consiglia l'uso di un Runnable associato alla barriera, *Confezionatore*, che si occupi di cambiare la confezione quando essa è piena). [10 pt]

```
public class Fabbrica {
    private final int NUM_PALLINE = 4;
    public static final int CAPACITA CONTENITORE = 50;
    public Fabbrica(){
        ...
        Confezione confezione = new Confezione(); // salva palline nella conf.
        Confezionatore conf = new Confezionatore(confezione); // cambia conf
        barrier = new CyclicBarrier(NUM_PALLINE, conf);
        ...
    }
    public void prelevaPallina(){}
    public void posizionaPallina(){}
    public void attendiCompletamentoConfezione(){}
    public void rifornimento(){}
    public static void main(String[] args) {
        Fabbrica fabbrica = new Fabbrica();
        Braccio[] bracci = new Braccio[4];
        for(int i = 0; i < 4; i++){
            bracci[i] = new Braccio(fabbrica);
            bracci[i].start();
        }
        Fornitore fornitore = new Fornitore(fabbrica);
        fornitore.start();
    }
}
```