



Escola Tècnica Superior d'Enginyeria  
de Telecomunicació de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA  
DEPARTAMENT D'ENGINYERIA TELEMÀTICA

**AST – Control d'avaluació continuada**  
24 de maig de 2011

Professors: Jordi Casademont, Marcel Fernández i Juan Luis Gorricho

Informacions addicionals:

- Duració 60 minuts
- Les preguntes de teoria s'han de respondre únicament en l'espai disponible

Nom Alumne:

1. Explica perquè el TCP té els ports anomenats "ben coneguts" i els aleatoris. Explica què són, quan s'utilitzen i dóna un exemple.
2. Dibuixa un diagrama que representi el què passa quan un ordinador realitza una petició DNS. Aquest diagrama ha de contenir tots els elements involucrats i els paquets intercanviats.
3. Què són les connexions persistents del HTTP?

4. Two computers are connected using a radio link that introduces a propagation delay of 0,5 ms. The processing time of each computer is 0. We also know that:
- The radio link has a data rate of 100 Mbps.
  - Data packets are 1530 bytes in length (30 bytes of MAC header, 20 bytes of IP header and 20 bytes of TCP header).
  - The TCP Window Size is 65535 bytes.
  - The channel is full duplex.
  - TCP uses the Slow Start mechanism.

In these conditions a user wants to transmit an image file of 10220 bytes:

- a) Plot a diagram with all involved packets.
- b) Compute the total time of the whole transmission.
- c) What is the user data rate in this case?

5. Un compte d'estalvis és compartit entre diferents processos. Cada procés pot treure o dipositar diners. El balanç del compte mai pot ser negatiu. Es demana programar una classe `Compte_Bancari` que resolgui aquest problema amb les operacions `dipositar(quantitat)` i `extreure(quantitat)` garantint que les operacions son resoltes per ordre d'invocació, és a dir, si hi ha un saldo de 1000€ i un procés vol treure 2000€ i posteriorment un altre procés vol treure 500€, aquest últim procés s'haurà d'esperar a que el que en vol 2000€, els pugui treure. Per la mateixa raó, una crida a dipositar amb un increment de saldo suficient pot provocar la consecució de més d'una operació d'extracció pendent.

Implementar **NOMÉS** el mètode `extreure(int quantitat)` de la classe `Compte_Bancari` amb la següent especificació:

```
import java.util.*;

public class Compte_Bancari{

    private int saldo;
    private int seguent_servit, ultim_arribat;
    private List<Integer> quantitats_pendents;

    public Compte_Bancari(){
        quantitats_pendents = new ArrayList<Integer>();
    }

    public synchronized void extreure(int quantitat){...}

    public synchronized void dipositar(int quantitat){

        saldo+= quantitat;
        if(!quantitats_pendents.isEmpty() && saldo>=quantitats_pendents.get(0)){
            saldo-=quantitats_pendents.remove(0);
            seguent_servit++;
            notifyAll();
        }
    }
}
```

Recordatori: la classe **ArrayList** té definits els mètodes:

- boolean **isEmpty()** per consultar si la llista està buida.
- E **get(int pos)** consulta l'element de la llista a la posició pos sense treure'l.
- E **remove(int pos)** consulta i treu l'element de la llista a la posició pos.