

11 de Novembre 2019

Programació Ratolí a l'emulador (a través de seqüències d'escape)? wtf

Programació Ratolí al terminal real, tty (amb events).

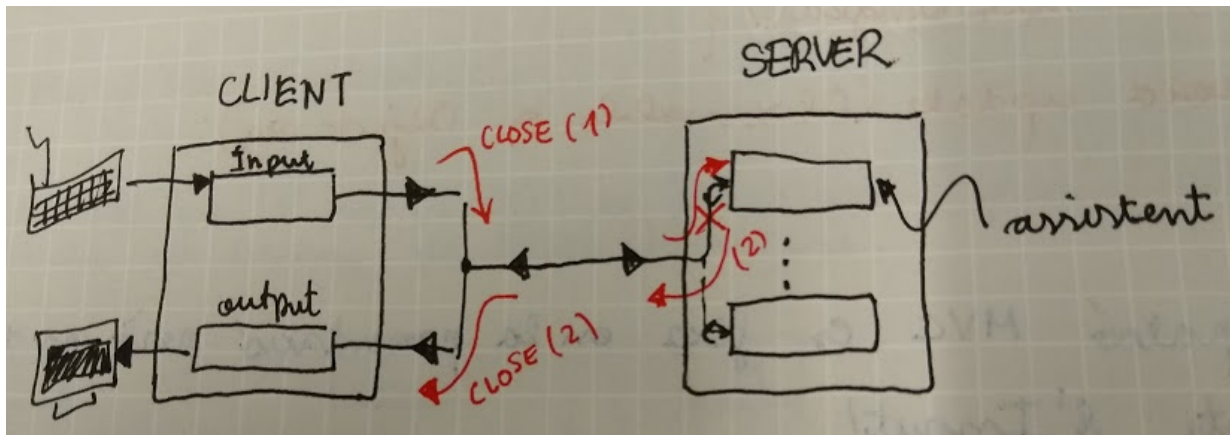
Biblioteca per unificar programació d'interfícies de consola (teclat i ratolí) independentment del terminal (xterm, tty, etc...) -> Biblioteca ncurses.

- Exemple Reproductor MP3 🎵 (amb ncurses):

Obligatori: P1, P2 (xat Java), P3 (client xat, Swing library, Java); tirar programa presentació final.

## P2: Xat de consola

- S'engega el servidor, passant-li un port efímer (i.e. port alt >32768).
  - S'engega el client, passant-li el host, el port del servidor i el nic (nom).
  - S'engega un segon client, passant-li el host, el port del servidor i el nic (nom2).
  - Es tanquen amb CNTRL+D.
  - (Al servidor es visualitza tot).
1. Programar MySocket & MyServerSocket que simplifiquin (encapsulin excepcions i els streams de text) de Socket i ServerSocket.
  2. Client: 2 threads concurrents (llegir+enviar i llegeix línies al servidor d'un altre client).  
El servidor envia les línies (mentre n'hi hagi) a tots els clients connectats (broadcast).  
El Thread d'Input tanca la connexió, després el servidor tanca l'assistent i notifica al thread d'Output del client perquè es tanqui.
  3. Servidor (multithreading), cada assistent és un thread, el reenviament a tots els assistents es fa a través d'un contenidor (un diccionari de parells clau (nick) - valor (socket), classe Map), és una zona crítica.



"No hi ha solucions, hi ha alternatives", Fncsc Öllr ☺

(...examples next pag.)

### Exemples: (útils per P2)

```
//multithreading, suposant els wrappers implementats (info per crear MySocket):
public class EchoServer {
    public static void main(String[] args){
        MyServerSocket ss = new MyServerSocket(Integer.parseInt(args[0]));
        while (true){
            MySocket s = ss.accept();
            new Thread(){ // classe anònima (sense nom) que extén Thread.
                public void run(){
                    String line;
                    while((line = s.readLine()) != null)
                        s.println(line);
                    s.close();
                }
            }.start(); // start() crea un fil i el fil executa el run()
        }
    }
}
```

```
// Multithreading, suposant els wrappers implementats:
public class EchoClient {
    public static void main(String[] args){
        MySocket sc = new MySocket(args[0], Integer.parseInt(args[1]));

        new Thread(){ // Input thread
            public void run(){
                String line;
                BufferedReader kbd = new BufferedReader(new InputStreamReader(System.in));
                try {
                    while ((line = kbd.readLine()) != null)
                        sc.println(line);
                } catch (IOException e){
                    e.printStackTrace();
                }
            }
        }.start();

        new Thread(){ // Output thread
            public void run(){
                String line;
                while ((line = sc.readLine()) != null)
                    System.out.println(line);
            }
        }.start();
    }
}
```

### **P3: Xat gràfic**

- 4 Widgets de la biblioteca Swing (Missatges, Usuaris connectats, Inputoavorriment i Send)
- Servidor multithread (no fer Selector NIO amb patró reactor)

