

# Pràctica 3: Erlang: Botoneres

Programació Concurrent i en Temps Real — iTIC

Antoni Escobet Canal

5 d'octubre de 2021

## Índex

<b>1 Organització</b>	<b>1</b>
1.1 Lliurament . . . . .	1
<b>2 Exercicis</b>	<b>1</b>

## 1 Organització

Aquesta sessió introdueix un mòdul prefabricat que ofereix una representació gràfica d'una botonera. Aquestes botoneres tindran un paper important més endavant pel disseny de l'ascensor que començera a la propera pràctica. En aquesta pràctica es tracta d'usar-les per familiaritzar-se amb el disseny de programes concurrents.

Per desenvolupar aquesta pràctica us cal el mòdul `bots.erl` que podeu obtenir de l'OCW iTIC, `ocw.itic.cat` o a l'Atenea.

Amb l'objectiu de reforçar l'hàbit d'usar sistemes de control de versions, cal desenvolupar la pràctica amb el suport del sistema que ofereix `http://escriny3.epsem.upc.edu`.

### 1.1 Lliurament

Cal lliurar els exercicis en un tarfile a través d'Atenea en la data fixada. Cal que el desenvolupament es faci usant `Subversion` a través de les facilitats que ofereix `http://escriny3.epsem.upc.edu`. Al mateix temps caldrà presentar oralment la pràctica durant la classe de laboratori que ja s'anunciarà.

## 2 Exercicis

**EXERCICI 2.1** Una botonera és un artefacte gràfic que simula una botonera d'ascensor. En la implementació que us subministrem, aquesta simulació es fa usant la llibreria de widgets `wxWidgets` i el seu vincle a `Erlang`. Des del punt de vista d'`Erlang` una botonera s'abstreu com un procés: cada missatge enviat al procés provoca una acció a la botonera. Tota aquesta infraestructura està continguda en el mòdul `bots` (de `BOTonera Simple`). El mòdul `bots` exporta una funció:

1. `bots:new(N,Pid)`

Inicialitza el sistema de widgets gràfics. Crea una nova botonera amb N botons, que corresponen a N plantes. El procés associat a aquesta nova botonera informará dels esdeveniments capturats per la botonera al procés Pid. Retorna el pid del procés associat a la botonera.

La forma d'usar una botonera és molt senzilla: es crea i després s'envien missatges al procés associat per comandar la botonera. En el següent exemple, es crea una botonera de 8 botons i després es sobreillumina el botó corresponent al pis 4:

```
B=bots:new(8,self()),
B!{light_on,4}
```

El protocol amb que es pot parlar al procés de la botonera és el que defineixen els següents missatges:

- {light\_on,N}
  - Il·lumina el botó corresponent al pis N.
- {light\_off,N}
  - Apaga la il·luminació del botó corresponent al pis N.
- close
  - Tanca la botonera.

De forma similar, la botonera pot enviar al procés que s'ha indicat en la seva creació els següents missatges:

- {clicked, Pis}
  - Indica que s'ha polsat el botó corresponent al pis Pis.
- abort
  - Indica que s'ha destruït la botonera per una acció de l'usuari (per exemple tancant la finestra).

Podeu comprovar la recepció dels missatges a la consola amb la comanda flush().

Vist el funcionament de les botoneres, en aquest exercici es demana que dissenyeu un programa que crea un botonera de 12 pisos, i procedeix indefinidament fent el següent procés:

1. Si algú polsa un botó, s'il·lumina tots els botons que van des del pis 0 fins al polsat.
2. Aleshores la botonera resta en aquest estat fins que algú polsa algun dels botons il·luminats, la qual cosa fa que s'apaguin tots els botons fins el que s'ha premut o un dels no il·luminats que els il·lumina fins els que s'ha premut.

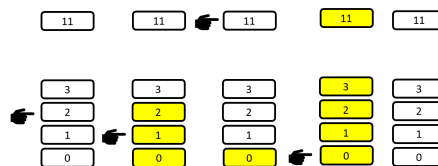


Figura 1: Seqüència dels botons.

EXERCICI 2.2 Usant la botonera, dissenyeu i implementeu un programa que fa el següent: crea una botonera de  $N$  pisos. Posteriorment espera a que algú polsi un botó. El boto polsat s'il·luminarà durant 3s, pampalluguejarà durant 3s a intervals de 0,5s, i finalment s'apagarà. Pot succeir que més d'un botó estigui encès simultàniament.  $N$  és un paràmetre que s'ha de passar quan es crida la funció.

Per implementar aquest comportament del sistema, definiu un procés encarregat d'apagar un botó que realitzi tot el cicle d'apagat: espera de 3s i les pampallugues. Cada vegada que es polsi un botó creeu un nou procés d'apagat associat al botó polsat.

En el cas en que es prem dues vegades o més el mateix botó, no es pot solapar l'acció d'encendre i apagar. S'ha d'esperar que n'acabi una per iniciar la següent.

EXERCICI 2.3 En aquest exercici es demana un programa que faci el següent: crea dues botoneres, una de dos pisos i una altra de 8 pisos. Cada vegada que es polsa un botó de la botonera petita, s'il·luminen els pisos parells o senars (segons si s'ha polsat el 0 o el 1) de la botonera gran. Si es pitja un botó de la botonera gran dels pisos il·luminats aleshores s'apaguen tots.

EXERCICI 2.4 Modifiqueu l'exercici anterior per executar les botoneres en diferents nodes. Feu que en un dels nodes hi hagi la botonera de 2 i a l'altre node la de 8. Tot el procés s'ha d'iniciar des d'un dels dos nodes. Recordeu que per poder comunicar dos processos ubicats en nodes diferents, s'ha de saber el nom del node i el seu identificador. Us serà útil registrar els processos. Per evitar problemes d'accés a la xarxa, podeu obrir dos nodes diferents al mateix ordinador.

EXERCICI 2.5 Aquest darrer exercici té com a objectiu dissenyar una "central" de botoneres: és a dir un procés que controla un conjunt de botoneres.

Implementeu un programa principal que crea una botonera de 2 pisos, que en direm botonera de control, i un procés central. El sistema complet ha de funcionar de la següent forma:

1. Cada vegada que es pitja el 1 de la botonera de control es crea una nova botonera sota el comandament del procés central. La botonera acabada de crear té els mateixos botons il·luminats que la resta. Noteu que això implica que el procés central té constància de l'estat de les botoneres que coneix.
2. Cada vegada que es pitja el 2 de la botonera de control es destrueixen totes les botoneres sota el comandament del procés central.
3. Cada vegada que algú pitja un botó de les botoneres ordinàries:
  - a) Si estava apagat, s'il·lumina el mateix botó a totes les botoneres.
  - b) Si estava encès, s'apaga el mateix botó de totes les botoneres.

El programa, estarà format per dos processos. Un procés de control, que gestionarà les accions realitzades des de la botonera de control ( 2 pisos) i un procés central per crear, destruir i activar o desactivar els botons de les botoneres de 8 pisos.

El procés central entén els següents missatges:

- new

Enviat per la botonera de control. Crea una nova botonera sempre de 8 pisos amb els botons que pertoqui il·luminats.

- `close`  
Enviat per la botonera de control. Tanca totes les botoneres que coneix.
- `abort`  
Indica que s'ha destruït alguna botonera que coneix per una acció de l'usuari (per exemple tancant la finestra).
- `{clicked, Pis}`  
Indica que en alguna botonera que coneix s'ha polsat el botó corresponent al pis `Pis`. Il·lumina o apaga totes les botoneres que coneix el botó corresponent

El procés de control entén els següents missatges que provenen de la botonera de control:

- `{clicked, 1}`  
Indica que s'ha de crear una nova botonera de 8.
- `{clicked, 0}`  
Indica que s'han de tancar totes les botoneres.
- `abort`  
Indica que s'ha de tancar tot el sistema. Les botoneres de 8 i la de control.

Aquest procés també pot enviar missatges. En particular pot enviar els missatges següents:

- `new`  
Indica que s'ha premut el pis 1 i que s'ha de crear una nova botonera de 8 pisos.
- `close`  
Indica que s'ha premut el pis 0 i que s'han de tancar totes les botoneres de 8 pisos.
- `abort`  
Indica que s'ha destruït la botonera de control per una acció de l'usuari (per exemple tancant la finestra) i que s'ha de tancar tot el sistema

Noteu que el procés central no distingeix una botonera d'una altra. Dissenyeu i implementeu aquest procés en un mòdul. Comproveu que funciona correctament.