



# Pràctica 4: Sistema de control per a un ascensor (I)

Programació Concurrent i en Temps Real — iTIC

Antoni Escobet Canal  
Sebastià Vila-Marta

15 d'octubre de 2014

## Índex

### 1 Organització

Aquesta sessió i les que segueixen tenen com a objectiu dissenyar i implementar un sistema de control per a una màquina elevadora (ascensor). La part física de l'ascensor que es controla (sistema motriu, cabina, portes, sensors, etc.) se simularà usant processos **Erlang**. L'objectiu final s'assolirà iterativament, cada nova pràctica constitueix un pas en el projecte.

La pràctica d'aquesta sessió es dedica a dissenyar i implementar un sistema de control per a un ascensor simple dotat únicament de cabina i botonera de cabina. Aquest ascensor no disposarà de memòria per “recordar” els diferents pisos en que s'ha d'aturar durant el trajecte.

Per desenvolupar aquesta pràctica us cal el mòdul **bcab.erl** que podeu obtenir de l'OCW iTIC, <http://ocw.itic.cat>. Aquest mòdul implementa una botonera de cabina més sofisticada que la que coneixíeu fins ara.

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. La durada estimada d'aquesta pràctica és de 2 setmanes.

### 2 Descripció del sistema

EL sistema que cal controlar és una màquina elevadora simple. Aquesta màquina està formada per una cabina que es mou en sentit vertical a través d'un sistema mecànic impulsat per un motor elèctric. El motor elèctric pot girar en dos sentits, la qual cosa provoca que la cabina pugi o baixi. La cabina té un recorregut màxim que ve donat per la longitud de les guies que la suporten. Aquest recorregut va des dels 0 m, que correspon a la posició més baixa, fins als

MAXREC mque correspon a la màxima posició superior. La cabina, quan el motor està actiu, es desplaça a una velocitat constant de  $VELCAB \text{ ms}^{-1}$ .

El fossat de la màquina està dotat de sensors que determinen quan la cabina passa per certs punts del recorregut. Particularment, hi ha un sensor a cada planta de manera que s'activa quan el terra de l'ascensor està exactament a nivell amb el terra de la planta corresponent. La posició de cada sensor la determina la forma de l'edifici al que serveix. En general la distància entre plantes, i per tant entre sensors, és diferent. La primera planta, arran de carrer, normalment no correspon a la posició 0 de la cabina atès que la cabina encara pot moure's un pèl més avall.

La cabina disposa d'una botonera de comandament que conté un botó il·luminable per cada pis, un display per a mostrar informació a l'usuari i un parell de botons per a obrir i tancar portes (que en aquesta pràctica encara no s'usaran).

En aquesta pràctica assumirem que la màquina elevadora no té portes, com en els antics elevadors industrials.

El funcionament de la màquina és el següent:

1. En engegar, la màquina fa un reset i situa al cabina a la planta 1. Durant el reset, en el display mostra el missatge **RESET**; a partir d'aquest instant, el display sempre mostra el pis on es troba la cabina.
2. Quan una persona entra a l'elevador, pot polsar un dels botons corresponents a un pis
  - En cas que sigui el mateix pis on es troba la cabina, aquesta acció no té cap efecte.
  - En cas que es tracti d'un pis diferent, s'il·lumina el botó polsat i l'elevador inicia el viatge corresponent.
3. El viatge d'un pis a un altre implica engegar el motor en la direcció escaient i estar atent als sensors que van indicant per on passa la cabina. Cada vegada que el sensor d'una planta s'activa, s'actualitza el pis que consta al display de la botonera. Quan s'arriba a la planta destí, es para el motor i s'apaga el botó que correspon al pis.

### 3 Disseny de la solució

El programari d'aquesta pràctica s'estructura en base a processos. Alguns processos simulen part del sistema electromecànic de l'ascensor mentre que altres constitueixen el sistema de control. En aquesta pràctica es posa més èmfasi en els processos que simulen el sistema electromecànic.

Els processos que constitueixen el sistema i les seves responsabilitats són els següents:

**motor** És el responsable de simular el conjunt sistema motriu – cabina. Coneix la posició física de la cabina en el recorregut i disposa d'un motor que la pot fer moure a velocitat constant.

**sensor** És el sistema que simula el conjunt de sensors de posició de la cabina. Obté la informació d'on és la cabina del procés motor i informa dels sensors que s'activen en cada moment.

**bcab** Simula la botonera de cabina, és a dir la botonera que es troba dins de la cabina de l'ascensor.

**ascensor** És el sistema de control. Supervisa el funcionament de tot el sistema electromecànic.

Les comunicacions entre els diferents processos són les que mostra el graf de la figura ???. En aquest graf el nodes corresponen als processos i els arcs orientats indiquen quin procés envia

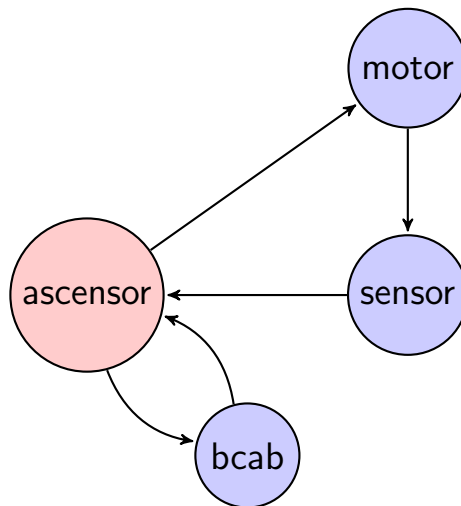


Figura 1: Comunicacions entre els processos del sistema.

missatges a quin altre. El color blau indica un procés que simula un aparell de la màquina elevadora; el color vermell marca els processos de control.

A continuació s'especifiquen breument cadascun dels processos que formen el sistema.

### 3.1 Procés motor

El procés motor simula el conjunt motor-cabina. Les operacions principals que admet són doncs les que engeguen i paren el motor en un o altre sentit de la marxa. Mentre el motor està engegat, la cabina es desplaça a una velocitat constant i, per tant, va variant la seva posició. Aquest procés fa una simulació discreta d'aquest mecanisme.

Els missatges que admet el procés són els següents:

- **run\_up**

Aquest missatge només es pot enviar si el motor està aturat. Engega el motor i desplaça la cabina amunt a velocitat **VELCAB** fins a nova ordre. Si la cabina supera el recorregut màxim **MAXREC**, executa una aturada d'emergència automàticament de forma autònoma, i envia el missatge **at\_top**.

- **run\_down**

L'ordre simètrica de l'anterior. En aquest cas l'aturada d'emergència es produeix si ultrapassa la posició 0 i el missatge que envia és **at\_bottom**.

- **stop**

Atura el motor i deixa la cabina en repòs. Només es pot enviar aquest missatge si el motor està engegat.

- **kill**

Acaba el procés.

El procés també pot informar de certs esdeveniments i ho fa enviant els missatges següents:

- {ready}

S'envia una única vegada just després que el ascensor s'inicialitzi per indicar que la màquina està a punt per funcionar.

- {at,P}

S'envia cada RESOL ms mentre el motor està engegat per indicar que la cabina està a la posició P.

- at\_top i at\_bottom

Missatges que indiquen una aturada d'emergència.

El codi corresponent a aquest procés s'allotja en un únic mòdul anomenat **motor** i té un únic punt d'entrada que correspon a la funció **start(Pid)** que arrenca el procés. Pid és el pid del procés a qui s'enviaran els missatges generats pel motor.

El funcionament d'aquest mòdul té un punt de dificultat atès que està simulant un procés continuu, el desplaçament de la cabina, de manera discreta.

Assumim que la cabina es troba a la posició  $h_0 \ll \text{MAXREC}$  i que arriba un missatge **run\_up**. El principi de funcionament és el següent:

1. Esperem RESOL ms.
2. Al cap de RESOL ms la cabina s'haurà desplaçat un espai  $\delta$  que depèn de RESOL i de VELCAB. La nova posició de la cabina és doncs  $h_1 = h_0 + \delta$ . Enviem una notificació indicant la nova posició {at, h1}.
3. Repetim el procés *ad infinitum* fins que aturen el motor o actua la parada d'emergència.

### 3.2 Procés sensor

El procés sensor rep les comunicacions de posició {at,P} que va enviant el motor i determina si la cabina activa algun dels sensors de posició. En cas afirmatiu, envia un missatge reflectint l'esdeveniment.

Els missatges que pot rebre el procés són els següents:

- {at,P}

Informa que la cabina es troba en la posició P.

- at\_top

Informa que la cabina ha fet una aturada d'emergència al final del recorregut.

- at\_bottom

Informa que la cabina ha fet una aturada d'emergència al principi del recorregut.

- ready

Informa que el sistema cabina-motor està a punt per funcionar.

- reset

Indica que el sistema de sensors ha de re-sincronitzar la posició de la cabina. Vegeu més endavant una descripció detallada.

- kill

Atura el procés.

Els missatges que envia són els següents:

- at\_top

Informa que la cabina ha fet una aturada d'emergència al final del recorregut.

- at\_bottom

Informa que la cabina ha fet una aturada d'emergència al principi del recorregut.

- ready

Informa que el sistema cabina-motor-sensor està a punt per funcionar.

- {sens\_pl,P}

Indica que s'ha activat el sensor de posició corresponent a la planta P.

El sistema de sensors ha de ser conscient de quina posició ocupa la cabina en tot moment. En certes ocasions, per exemple a l'arrencada, aquesta posició es desconeguda. En aquest cas ha de poder fer un reset. El reset és un procés que segueix la següent estratègia:

1. Arrenca el motor i fa baixar la cabina fins la parada d'emergència `at_bottom`.
2. Arrenca el motor i fa pujar la cabina fins la primera planta.

Amb això, sigui quina sigui la posició inicial de la cabina, s'acaba situant a la planta 1.

El fet que el motor se simuli de forma discreta introdueix alguns detalls que cal considerar. El sistema de sensors sempre és conscient de la posició actual de la cabina. Suposem que la cabina es troba a la posició  $h_0$ , quan el procés rep un missatge `{at,H1}`, sabem que la cabina s'ha mogut d' $h_0$  a  $h_1$ , ja sigui pujant o baixant. Assumim que és pujant i per tant  $h_0 < h_1$ . Si el sensor de la planta  $P$  se situa en la posició  $h_P$ , l'esdeveniment `{sens_pl,P}` s'haurà d'activar en cas que  $h_P$  es trobi dins de l'interval  $[h_0, h_1)$ .

### 3.3 Procés bcab

La botonera de cabina és un artefacte gràfic que simula una botonera de cabina 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 `bcab` (de Botonera de CABina). El mòdul `bcab` exporta una única funció:

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

Crea una nova botonera amb  $N$  botons, que corresponen a  $N$  plantes, un display i dos botons pel control de portes. 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 similar a la que havíeu usat en la pràctica anterior.

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.

- {display, M}

Mostra M pel display de la botonera. M pot ser un enter entre 0 i 99 o bé una cadena de caràcters.

- kill

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 pulsat el botó corresponent al pis Pis.

- open\_doors

Indica que s'ha pulsar el botó d'obrir portes.

- close\_doors

Indica que s'ha pulsar el botó de tancar portes.

- abort

Indica que s'ha destruït la botonera per una acció de l'usuari (per exemple tancant la finestra).

### 3.4 Procés ascensor

El procés ascensor és l'encarrega de controlar el sistema i implementar les polítiques de funcionament que s'han descrit prèviament. El seu disseny concret és responsabilitat de l'estudiant.

## 4 Desenvolupament de la pràctica

**EXERCICI 4.1** Després de llegir amb cura el sistema que es proposa feu un primer disseny del procés ascensor. Tingueu en compte que, abans de començar, ha d'inicialitzar correctament tot el sistema.

**EXERCICI 4.2** Implementeu el sistema motor i comproveu el seu correcte funcionament. Per comprovar-ne el funcionament us serà de molta utilitat tenir a mà un mòdul **logger**, que simplement arrenca un procés que tot el que rep ho escriu pel canal de sortida. Podeu usar aquest mòdul per “espiar” tot el que va enviant el motor mentre el proveu a la shell d'**Erlang**.

En la implementació les constants **VELCAB**, **RESOL** i **MAXREC** han d'implementar-se com a tals. Invetigueu com es defineixen constants a **Erlang**. Inicialment assumiu les següents dades:

Constant	Valor
VELCAB	0,6 m s <sup>-1</sup>
RESOL	100,0 ms
MAXREC	22,0 m

EXERCICI 4.3 Implementeu i proveu el mòdul sensor. En la implementació d'aquest mòdul s'ha d'usar una constant que contingui la taula de posicions dels sensors, de manera que resulti immediat re-configurar la posició que ocupen. Inicialment suposeu que l'edifici té 4 plantes i que els sensors estan distribuïts tal i com s'indica aquí:

Sensor	Alçada (m)
sens_pl 1	0,5
sens_pl 2	5,5
sens_pl 3	8,5
sens_pl 4	12,5

EXERCICI 4.4 Implementeu el mòdul ascensor. Comproveu que tot el sistema funciona correctament.