

17 de novembre de 2014

Examen de PCTR

1h 30 min

**Exercici 1: Màquina expendedora (5 punts)**

L'objectiu d'aquest problema és el de dissenyar un sistema concurrent per controlar una màquina genèrica expendedora de productes (fins a 16 numerats de 0 a 15). La funcionalitat d'una d'aquestes màquines és:

- Inicialment el saldo per aconseguir un producte es 0.
- Quan s'introdueix una moneda: si hi ha canvi s'actualitza el saldo i es visualitza al display, si no hi ha canvi es retorna la moneda.
- Quan es prem el botó de retorn s'ha de retornar la quantitat del saldo i actualitzar-lo a zero.
- Quan es selecciona un producte, si el saldo es suficient es serveix el producte i es retorna la quantitat restant deixant el saldo a zero, si el saldo no es suficient s'informa en el display de la quantitat que falta i la màquina oblida la selecció.

Per desenvolupar aquest programari es disposa d'un paquet 'maquina' ja implementat per controlar els dispositius de la màquina:

Valor = detectar \_Moneda(). Provoca que el procés que l'utilitza quedi bloquejat fins que un usuari introdueixi una moneda.  
 (Retorna el valor de la moneda).

C = hi\_ha\_canvi(). Retorna un zero en el cas de que n'hi hagi i un u en cas contrari.

detectar\_Devolucio(). Provoca que el procés que el crida quedi bloquejat fins que l'usuari premi el botó de retorn dels diners acumulats.

retornar(Q). Retorna la quantitat de diners Q. L'operació resol el nombre de monedes i el tipus que s'ha de retornar.

Cost = preu(P). Retorna el preu del producte P.

P = seleccio(). Provoca que el procés que el crida quedi bloquejat fins que un usuari seleccioni un producte P. Retorna el n° del producte.

servir(P). Provoca que la màquina serveixi una unitat del producte P.

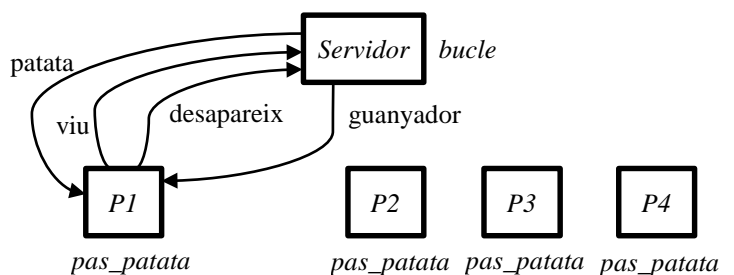
mostrar(Q). Mostra la quantitat Q en el display de la màquina.

**Exercici 2 (5 Punts): El joc de la patata**

Es vol implementar en Erlang un joc simple que consisteix a passar una patata (un nombre natural aleatori) entre els diferents processos en un entorn distribuït. Cada vegada que un dels participants rep la patata, si no val zero, es disminueix en un el valor i passar a un altre procés. Quan és zero, el participant (el procés) s'elimina i el joc continua amb la resta dels participants (a patata s'assigna un nou número natural aleatori). Aquí teniu un parell d'exemples d'execucions del sistema:

1> patata:inici(4). El procés 1 passa la patata El procés 3 passa la patata El procés 2 passa la patata El procés 1 passa la patata El procés 4 ha guanyat	2> patata:inici(4). El procés 4 passa la patata El procés 3 passa la patata El procés 1 ha guanyat
---	---

S'utilitza la següent arquitectura distribuïda (4 participants):



on hi ha un servidor que decideix aleatòriament a quin jugador (P<sub>i</sub>) passa la patata.

Una possible traça de la patata (mentre sigui diferent de zero) per a aquest sistema pot ser:

Servidor → P2 → Servidor → P1 → Servidor → P4 → Servidor → P2 → Servidor → P3 → ...

A la figura, només es mostren els termes (patata, guanyador, viu, desapareix) enviats entre el servidor i P1, però els mateixos termes s'envien entre el servidor i els altres processos. També es mostra a la figura les principals funcions executades en cada procés: el servidor executa el bucle de la funció i els participants executar la funció pas\_patata. Vosaltres heu de decidir la informació que s'intercanvia entre el servidor i els participants, juntament amb les condicions (per exemple, l'ID del servidor es pot enviar als participants per permetre tornar la patata).

Feu un programa amb Erlang per aquest sistema. N'hi ha prou en fer les funcions: **iniciar**, **bucle** i **pas\_patata**.

La funció iniciar(N): inicialitza tot el sistema per N jugadors.

Us poden ser útils les funcions d'Erlang:

random:uniform(N): retorna un natural entre 1 i N

lists:nth(N, L): que retorna l'element N de la llista L.