

Trabajo Práctico N° 14:

Módulo Concurrente (Realizar programas en R-info con distintos tipos de robots. Utilizar memoria compartida para la comunicación y sincronización entre robots. Combinar problemas con memoria compartida y pasaje de mensajes. Distinguir modelos de algoritmos a desarrollar de acuerdo al problema planteado).

Ejercicio 1: Clientes y Servidores.

Existe un robot que sirve de flores a tres robots clientes. Cada cliente solicita al servidor que le deposite en su esquina siguiente una cantidad de flores aleatoria (entre 1 y 4). Por ejemplo, si el cliente se encuentra en la esquina (2,1), le solicitará que coloque x cantidad de flores en la esquina (2,2). Cuando el robot servidor deposita las flores en la esquina solicitada, el cliente las junta y las deposita una a una a lo largo de la avenida en la que se encuentra. El programa finaliza cuando todos los robos clientes completan su avenida. Asumir que el robot servidor tiene flores suficientes en su bolsa. El robot servidor se inicia en la esquina (100,100). Los robots clientes inician en las esquinas (1,1), (2,1) y (3,1), respectivamente.

```
programa TP14_E1
procesos
    proceso juntarFloresEsquina
        comenzar
            mientras (HayFlorEnLaEsquina)
                tomarFlor
            fin
        proceso depositarFloryAvanzar (E flores: numero)
            comenzar
                mientras ((flores>0) & (PosCa<100))
                    depositarFlor
                    flores:=flores-1
                    mover
                fin
            areas
                area1: AreaPC(1,1,1,100)
                area2: AreaPC(2,1,2,100)
                area3: AreaPC(3,1,3,100)
                areaS: AreaP(100,100,100,100)
            robots
                robot cliente
                variables
                    rob, av, ca, flores: numero
                    ok: boolean
                comenzar
                    RecibirMensaje(rob,robotS)
                    mientras (PosCa<100)
```

```
Random(flores,1,4)
av:=PosAv
ca:=PosCa+1
EnviarMensaje(rob,robotS)
EnviarMensaje(flores,robotS)
EnviarMensaje(av,robotS)
EnviarMensaje(ca,robotS)
RecibirMensaje(ok,robotS)
Pos (av,ca)
juntarFloresEsquina
Pos (av,ca-1)
depositarFlorAvanzar(flores)
EnviarMensaje(rob,robotS)
EnviarMensaje(0,robotS)
fin
robot servidor
variables
rob, avIni, caIni, av, ca, flores, term: numero
ok: boolean
comenzar
avIni:=PosAv
caIni:=PosCa
term:=0
ok:=V
EnviarMensaje(1,robot1)
EnviarMensaje(2,robot2)
EnviarMensaje(3,robot3)
mientras (term<3)
    RecibirMensaje(rob,*)
    si (rob=1)
        RecibirMensaje(flores,robot1)
        si (flores<>0)
            RecibirMensaje(av,robot1)
            RecibirMensaje(ca,robot1)
            Pos (av,ca)
            repetir flores
                depositarFlor
            Pos (av,ca)
            EnviarMensaje(ok,robot1)
        sino
            term:=term+1
    sino
    si (rob=2)
        RecibirMensaje(flores,robot2)
        si (flores<>0)
            RecibirMensaje(av,robot2)
            RecibirMensaje(ca,robot2)
            Pos (av,ca)
            repetir flores
                depositarFlor
```

```
Pos (avIni,caIni)
EnviarMensaje(ok,robot2)
sino
    term:=term+1
sino
    RecibirMensaje(flores,robot3)
    si (flores<>0)
        RecibirMensaje(av,robot3)
        RecibirMensaje(ca,robot3)
        Pos (av,ca)
        repetir flores
            depositarFlor
        Pos (avIni,caIni)
        EnviarMensaje(ok,robot3)
    sino
        term:=term+1
fin
variables
robot1, robot2, robot3: cliente
robotS: servidor
comenzar
AsignarArea(robot1,area1)
AsignarArea(robot2,area2)
AsignarArea(robot3,area3)
AsignarArea(robotS,areaS)
AsignarArea(robotS,area1)
AsignarArea(robotS,area2)
AsignarArea(robotS,area3)
Iniciar(robot1,1,1)
Iniciar(robot2,2,1)
Iniciar(robot3,3,1)
Iniciar(robotS,100,100)
fin
```

Ejercicio 2: Productores y Consumidores.

Existen dos robots productores que recorren las avenidas 5 y 10, respectivamente, juntando todos los papeles de su avenida. A lo largo del recorrido, cada vez que juntan 5 papeles, los depositan en la esquina (50,50). Además, existen dos robots consumidores que intentan tomar una cantidad aleatoria de papeles (entre 2 y 5) de la esquina (50,50) para depositarla en su esquina de origen. Si la esquina (50,50) no posee la cantidad de papeles requerida, vuelven a su esquina de origen sin tomar ningún papel. Si luego de 8 intentos seguidos, detectan que la esquina (50,50) no tiene papeles suficientes para juntar, entonces, asumirán que los productores ya han completado su trabajo y, por lo tanto, terminarán su tarea también. Los consumidores inician en las esquinas (11,1) y (12,1), respectivamente.

```
programa TP14_E2
areas
areaC: AreaC(50,50,50,50)
areaP1: AreaP(5,1,5,100)
areaP2: AreaP(10,1,10,100)
areaC1: AreaP(11,1,11,1)
areaC2: AreaP(12,1,12,1)
robots
robot productor
variables
av, ca, papeles: numero
comenzar
repetir 99
    av:=PosAv
    ca:=PosCa
    mientras (HayPapelEnLaEsquina)
        tomarPapel
        papeles:=papeles+1
        si (papeles=5)
            papeles:=0
            BloquearEsquina(50,50)
            Pos (50,50)
            repetir papeles
                depositarPapel
                Pos (av,ca)
            LiberarEsquina(50,50)
        mover
    fin
    robot consumidor
variables
avIni, caIni, papeles, papelesReq, int: numero
comenzar
avIni:=PosAv
caIni:=PosCa
int:=0
mientras (int<8)
```

```
papeles:=0
Random(papelesReq,2,5)
BloquearEsquina(50,50)
Pos (50,50)
mientras (HayPapelEnLaEsquina & (papeles<papelesReq))
    tomarPapel
    papeles:=papeles+1
    si (papeles<papelesReq)
        repetir papeles
            depositarPapel
            Pos (avIni,caIni)
            LiberarEsquina(50,50)
            int:=int+1
    sino
        Pos (avIni,caIni)
        LiberarEsquina(50,50)
        repetir papeles
            depositarPapel
            int:=0
    fin
variables
robotP1, robotP2: productor
robotC1, robotC2: consumidor
comenzar
AsignarArea(robotP1,areaC)
AsignarArea(robotP1,areaP1)
AsignarArea(robotP2,areaC)
AsignarArea(robotP2,areaP2)
AsignarArea(robotC1,areaC)
AsignarArea(robotC1,areaC1)
AsignarArea(robotC2,areaC)
AsignarArea(robotC2,areaC2)
Iniciar(robotP1,5,1)
Iniciar(robotP2,10,1)
Iniciar(robotC1,11,1)
Iniciar(robotC2,12,1)
fin
```

Ejercicio 3: Sincronización Barrera.

Tres robots deben vaciar de papeles su avenida, comenzando por la calle 1 y terminando en la calle 100. El trabajo lo deben realizar todos juntos y en etapas: los robots inician juntos y, cuando todos completan una etapa del trabajo, pueden avanzar a la siguiente, lo que significa que, para poder pasar de etapa, los robots deben esperar que todos hayan completado la etapa en curso. Se proponen dos posibles soluciones a este problema (etapas homogéneas o etapas heterogéneas):

En cada solución, analizar cómo debería finalizar el programa. Los robots inician en las esquinas (1,1), (2,1) y (3,1), respectivamente. Existe un robot coordinador, cuya única tarea es asignar identificadores a cada robot.

(a) Implementar el programa considerando que cada robot completa una etapa cada 5 esquinas.

```
programa TP14_E3a
procesos
    proceso juntarPapelesEsquina
        comenzar
            mientras (HayPapelEnLaEsquina)
                tomarPapel
            fin
        areas
            area1: AreaP(1,1,1,100)
            area2: AreaP(2,1,2,100)
            area3: AreaP(3,1,3,100)
            areaC: AreaP(5,5,5,5)
        robots
            robot sincronizado
        variables
            rob: numero
            ok: boolean
        comenzar
            ok:=V
            RecibirMensaje(rob,robotC)
            repetir 19
                repetir 5
                    juntarPapelesEsquina
                    mover
                    si (rob=1)
                        EnviarMensaje(ok,robot2)
                        EnviarMensaje(ok,robot3)
                    sino
                        si (rob=2)
                            EnviarMensaje(ok,robot1)
                            EnviarMensaje(ok,robot3)
                        sino
                            EnviarMensaje(ok,robot1)
```

```
EnviarMensaje(ok,robot2)
repetir 2
    RecibirMensaje(ok,*)
repetir 4
    juntarPapelesEsquina
    mover
juntarPapelesEsquina
si (rob=1)
    EnviarMensaje(ok,robot2)
    EnviarMensaje(ok,robot3)
sino
    si (rob=2)
        EnviarMensaje(ok,robot1)
        EnviarMensaje(ok,robot3)
    sino
        EnviarMensaje(ok,robot1)
        EnviarMensaje(ok,robot2)
repetir 2
    RecibirMensaje(ok,*)
fin
robot coordinador
comenzar
    EnviarMensaje(1,robot1)
    EnviarMensaje(2,robot2)
    EnviarMensaje(3,robot3)
fin
variables
robot1, robot2, robot3: sincronizado
robotC: coordinador
comenzar
    AsignarArea(robot1,area1)
    AsignarArea(robot2,area2)
    AsignarArea(robot3,area3)
    AsignarArea(robotC,areaC)
    Iniciar(robot1,1,1)
    Iniciar(robot2,2,1)
    Iniciar(robot3,3,1)
    Iniciar(robotC,5,5)
fin
```

(b) Implementar el programa considerando que cada robot completa una etapa luego de juntar N papeles. El valor de N (entre 1 y 5) lo calcula cada robot antes de iniciar cada etapa.

programa TP14_E3b
procesos
 proceso juntarPapelEsquina (ES papeles: numero)
 comenzar

```
si (HayPapelEnLaEsquina)
    tomarPapel
    papeles:=papeles+1
sino
    mover
fin
areas
area1: AreaP(1,1,1,100)
area2: AreaP(2,1,2,100)
area3: AreaP(3,1,3,100)
areaC: AreaP(5,5,5,5)
robots
robot sincronizado
variables
rob, papeles, papelesEtapa: numero
ok, ok1, ok2: boolean
comenzar
ok:=V
ok1:=V
ok2:=V
RecibirMensaje(rob,robotC)
mientras ((PosCa<100) & (ok1) & (ok2))
    papeles:=0
    Random(papelesEtapa,1,5)
    mientras ((papeles<papelesEtapa) & (PosCa<100))
        juntarPapelEsquina(papeles)
        si (papeles<papelesEtapa)
            ok:=F
            si (rob=1)
                EnviarMensaje(ok,robot2)
                EnviarMensaje(ok,robot3)
            sino
                si (rob=2)
                    EnviarMensaje(ok,robot1)
                    EnviarMensaje(ok,robot3)
                sino
                    EnviarMensaje(ok,robot1)
                    EnviarMensaje(ok,robot2)
            RecibirMensaje(ok1,*)
            RecibirMensaje(ok2,*)
        fin
    robot coordinador
    comenzar
        EnviarMensaje(1,robot1)
        EnviarMensaje(2,robot2)
        EnviarMensaje(3,robot3)
    fin
variables
robot1, robot2, robot3: sincronizado
robotC: coordinador
```

comenzar

```
AsignarArea(robot1,area1)
AsginrarArea(robot2,area2)
AsginrarArea(robot3,area3)
AsginrarArea(robotC,areaC)
Iniciar(robot1,1,1)
Iniciar(robot2,2,1)
Iniciar(robot3,3,1)
Iniciar(robotC,5,5)
```

fin

Ejercicio 4: Jefe y Trabajadores - Master / Slave.

Un robot jefe asigna tareas a los trabajadores. Las tareas consisten en (1) recoger flores, (2) recoger papeles, (3) vaciar bolsa, (4) finalizar. Existen 2 robots trabajadores que reciben solicitudes de tareas del robot jefe. Para cada solicitud, reciben la tarea y la esquina donde deben realizarla (salvo cuando la tarea es 4, que no deben acceder a una esquina). Luego de recibir la tarea, los robots van a la esquina indicada, realizan la tarea, avisan al jefe que ya la completaron y quedan a la espera de una nueva tarea. El robot jefe inicia en la esquina (1,1) y los robots trabajadores inician en las esquinas (2,1) y (3,1), respectivamente. Las tareas se asignan aleatoriamente a cualquier esquina dentro del cuadrante comprendido entre las esquinas (2,2) y (100,100). El robot jefe envía 10 tareas aleatorias a trabajadores aleatorios y termina. Analizar: ¿Existe el riesgo de que el programa quede bloqueado y que ningún robot trabajador pueda realizar su tarea? ¿En qué caso puede suceder esto? ¿Qué resulta necesario considerar para evitar esta situación?

```
programa TP14_E4
procesos
    proceso recogerFlores
        comenzar
            mientras (HayFlorEnLaEsquina)
                tomarFlor
            fin
        proceso recogerPapeles
        comenzar
            mientras (HayPapelEnLaEsquina)
                tomarPapel
            fin
        proceso vaciarBolsa
        comenzar
            mientras (HayFlorEnLaBolsa)
                depositarFlor
            mientras (HayPapelEnLaBolsa)
                depositarPapel
            fin
areas
areaPC: AreaPC(2,2,100,100)
area1: AreaP(2,1,2,1)
area2: AreaP(3,1,3,1)
areaM: AreaP(1,1,1,1)
robots
robot slave
variables
    avIni, caIni, av, ca, tarea: numero
comenzar
    avIni:=PosAv
    caIni:=PosCa
    RecibirMensaje(tarea,robotM)
    mientras (tarea<>4)
```

```
RecibirMensaje(av,robotM)
RecibirMensaje(ca,robotM)
BloquearEsquina(av,ca)
Pos (av,ca)
si (tarea=1)
    recogerFlores
sino
    si (tarea=2)
        recogerPapeles
    sino
        vaciarBolsa
Pos (avIni,caIni)
LiberarEsquina(av,ca)
RecibirMensaje(tarea,robotM)
fin
robot master
variables
rob, av, ca, tarea: numero
rob1Act, rob2Act: boolean
comenzar
rob1Act:=V
rob2Act:=V
repetir 10
rob:=0
si ((rob1Act) & (rob2Act))
    Random(rob,1,2)
sino
    si (rob1Act)
        rob:=1
    sino
        si (rob2Act)
            rob:=2
    si (rob<>0)
        Random(tarea,1,4)
        Random(av,2,100)
        Random(ca,2,100)
    si (rob=1)
        EnviarMensaje(tarea,robot1)
    si (tarea<>4)
        EnviarMensaje(av,robot1)
        EnviarMensaje(ca,robot1)
    sino
        rob1Act:=F
    sino
        EnviarMensaje(tarea,robot2)
    si (tarea<>4)
        EnviarMensaje(av,robot2)
        EnviarMensaje(ca,robot2)
    sino
        rob2Act:=F
```

```
si (rob1Act)
    EnviarMensaje(4,robot1)
si (rob2Act)
    EnviarMensaje(4,robot2)
fin
variables
robot1, robot2: slave
robotM: master
comenzar
AsignarArea(robot1,areaPC)
AsignarArea(robot1,area1)
AsignarArea(robot2,areaPC)
AsignarArea(robot2,area2)
AsignarArea(robotM,areaM)
Iniciar(robot1,2,1)
Iniciar(robot2,3,1)
Iniciar(robotM,1,1)
fin
```