



Taller de Programación



AGENDA



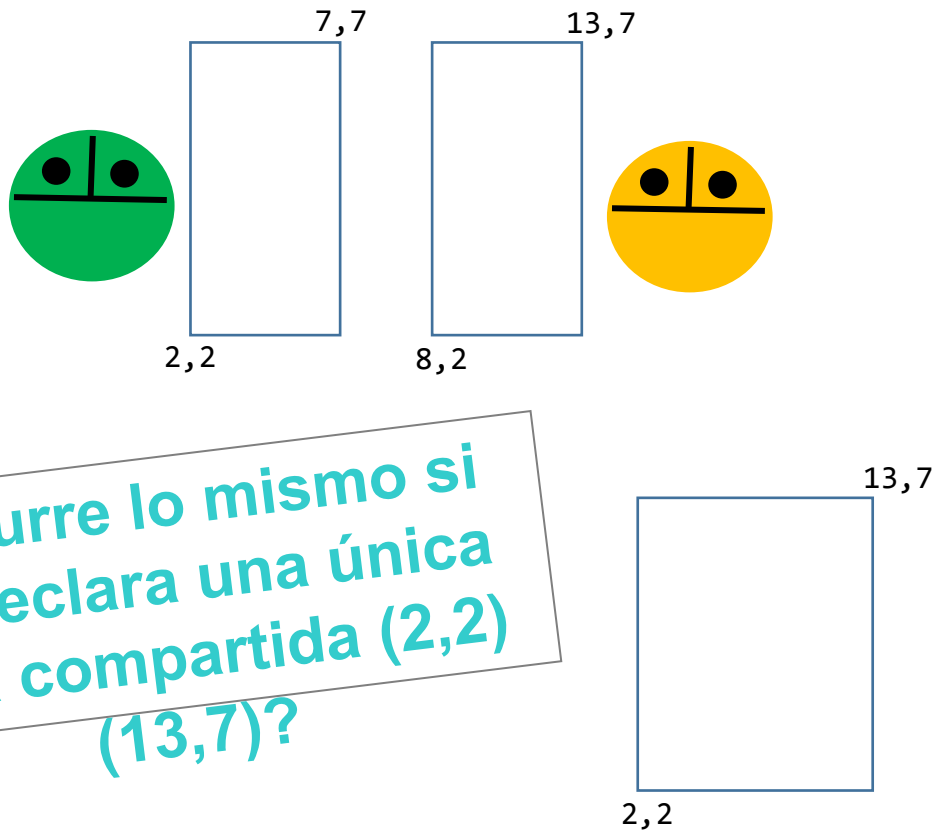
Mecanismos de Comunicación - Memoria Compartida

Ejemplos

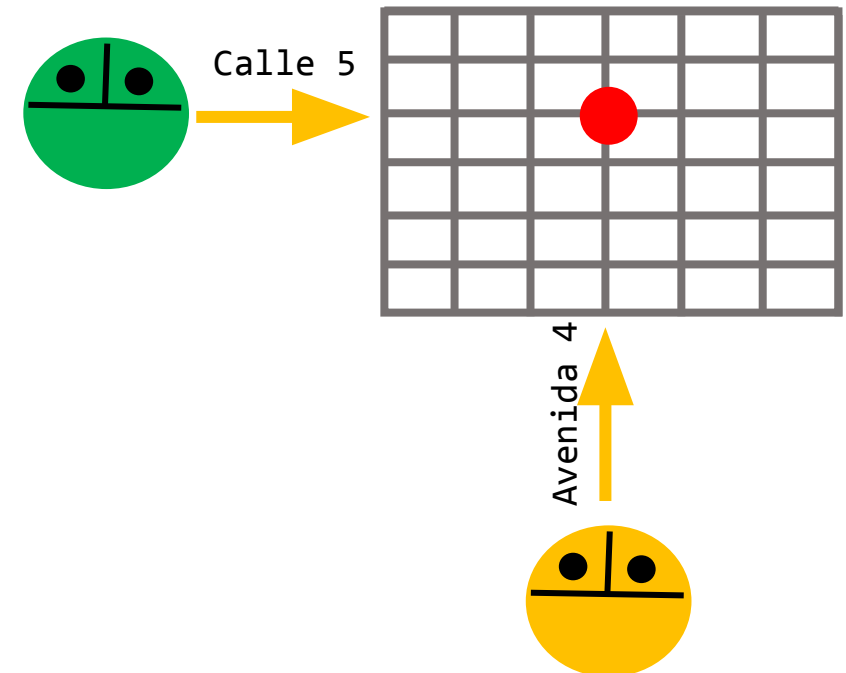


Mecanismos de Comunicación – MEMORIA COMPARTIDA

Realizar un programa donde existen **dos robots**. El robot 1 trabaja en su área privada delimitada por las esquinas (2,2) y (7,7) contando esquinas vacías y el robot 1 trabaja en su área privada delimitada por las esquinas (8,2) y (13,7) contando esquinas vacías.

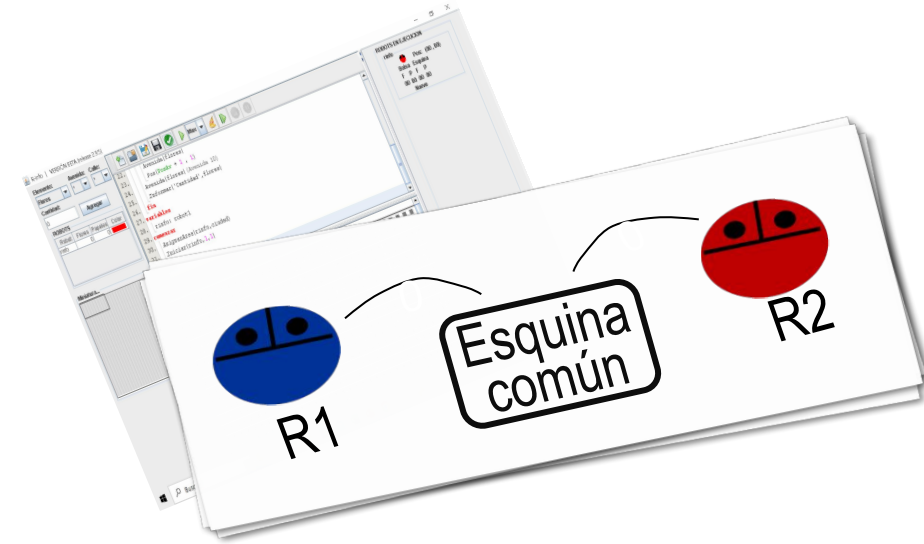


Realizar un programa donde existen **dos robots**. El robot 1 llamado avenida debe recorrer la avenida 10 y juntar las flores que encuentre. El robot 2 llamado calle debe recorrer la calle 5 juntando los papeles que encuentre.





COMUNICACIÓN – Memoria Compartida



BLOQUEAR RECURSO

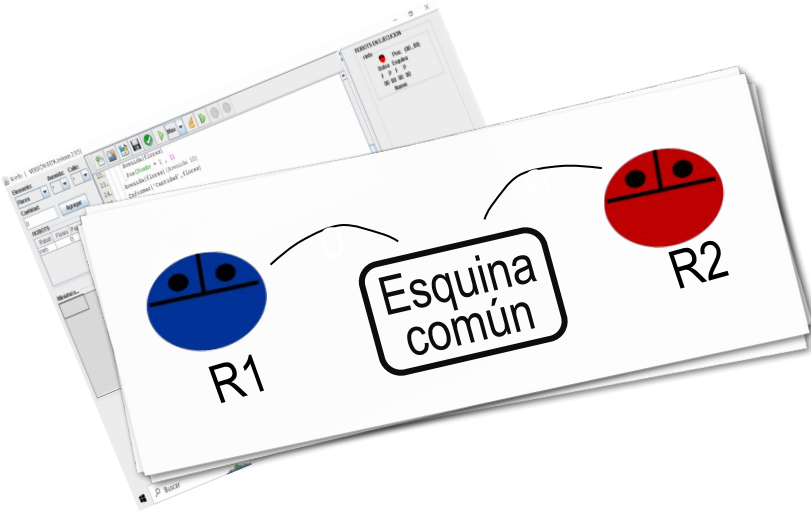
Dado un recurso compartido (por 2 o más procesos) que está **DISPONIBLE** se bloquea ese recurso para que otro proceso no pueda accederlo-

LIBERAR UN RECURSO

Dado un recurso compartido (por 2 o más procesos) **BLOQUEADO** el programador libera dicho recurso para que cualquier proceso pueda bloquearlo.



COMUNICACIÓN – Memoria Compartida



¿Cómo es
en CRME?

BLOQUEAR RECURSO - CONSIDERACIONES

- Puede realizarlo el programador o el Sistema Operativo
- Sólo se bloquea un recurso libre. Si el recurso ya está bloqueado no se debe intentarse hacerlo.
- Hay que bloquear un recurso cuando puede ser accedido por dos o más procesos de un programa.

LIBERAR UN RECURSO - CONSIDERACIONES

- Puede realizarlo el programador o el Sistema Operativo
- Sólo se libera un recurso ocupado. Si el recurso no está bloqueado no se debe intentarse hacerlo.
- Hay que liberar un recurso cuando puede ser accedido por dos o más procesos de un programa.



COMUNICACIÓN – Memoria Compartida

```
programa ejemploBloqueo
areas
  area1: AreaPC(1,1,20,20)

robots
  robot tipo1
  comenzar
  ...
  fin

variables
  robot1: tipo1
  robot2: tipo1

comenzar
  ...
  fin
```



Supongamos que el robot 1, y
el robot 2 deben acceder a
la esquina (5,10) en algún
momento

```
BloquearEsquina(avenida,calle)
```

```
BloquearEsquina(2,8)
```

```
BloquearEsquina(posAv+1,posCa)
```

```
BloquearEsquina(ave,ca)
```

*Cómo queda
el
programa?*



COMUNICACIÓN – Memoria Compartida

```
programa ejemploBloqueo
areas
    area1: AreaPC(1,1,20,20)
```

```
robots
    robot tipo1
    comenzar

    ...
    BloquearEsquina(5,10)
    ....
    fin
```

```
variables
    robot1: tipo1
    robot2: tipo1
comenzar
    ...
fin
```

- Si la esquina (5,10) está **desbloqueada**, entonces se marca como bloqueada y el robot continua ejecutando su código.
- Si la esquina (5,10) está **bloqueada**, entonces el robot “queda esperando” hasta que la esquina se libere y pueda ejecutar la instrucción de bloqueo.

Cómo se libera un esquina?



COMUNICACIÓN – Memoria Compartida

```
programa ejemploBloqueo
areas
  area1: AreaPC(1,1,20,20)

robots
  robot tipo1
  comenzar
  ...
  fin

variables
  robot1: tipo1
  robot2: tipo1

comenzar
  ...
  fin
```



Supongamos que el robot 1, y
el robot 2 deben liberar la
esquina (5,10) en algún
momento

```
LiberarEsquina(avenida,calle)
```

```
LiberarEsquina(2,8)
```

```
LiberarEsquina(posAv+1,posCa)
```

```
LiberarEsquina(ave,ca)
```

*Cómo queda
el
programa?*



COMUNICACIÓN – Memoria Compartida

```
programa ejemploBloqueo
areas
  area1: AreaPC(1,1,20,20)
robots
  robot tipo1
  comenzar

    ...
    BloquearEsquina(5,10)
    ....
    LiberarEsquina(5,10)
    ....
  fin
variables
  robot1:tipo1
  robot2: tipo1
comenzar
  ...
fin
```

- La esquina (5,10) está **bloqueada**, entonces el robot al ejecutar LiberarEsquina permite que pueda ser bloqueada nuevamente.



COMUNICACIÓN – MEMORIA COMPARTIDA

Las esquinas deben bloquearse sólo cuando dos o más robots pueden querer acceder.

Las esquinas deben permanecer bloqueadas el menor tiempo posible que garantice que el programa funcione.

Una esquina bloqueada siempre debe ser desbloqueada en algún momento del programa.

No debe bloquearse la esquina en que el robot ya se encuentra parado (ya que podría estar bloqueada previamente y podría haber otro robot). Siempre debe bloquearse primero la esquina y luego posicionarse en ella.



EJERCICIOS PARA ANALIZAR

Supongamos que tenemos un programa en el cual están declarados dos tipos robots y dos variables robot, una correspondiente a cada tipo. El robot 1 (de tipo 1) inicia su trabajo en la esquina (1,1) camina 5 cuadras y luego se posiciona en la esquina (10,10). El robot2 (de tipo 2) inicia su trabajo en la esquina (2,2) junta las flores de la esquina y luego se posiciona en la esquina (10,10). Ambos robots después de su trabajo vuelven a su esquina original.

ROBOT tipo1

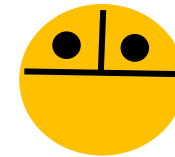


Repetir 5
mover

Posicionarse en la esquina (10,10)

Posicionarse en la esquina (1,1)

ROBOT tipo2



Mientras (hay flor en la esquina)
tomarFlor

Posicionarse en la esquina (10,10)

Posicionarse en la esquina (2,2)



EJERCICIOS PARA ANALIZAR

```
programa solucion1
areas
  area1: AreaPC(1,1,100,100)
```

```
robots
  robot tipo1
  comenzar
  ...
  fin
```

```
robot tipo2
  comenzar
  ...
  fin
```

```
variables
  robot1: tipo1
  robot2: tipo2
  ...
```

```
robot tipo1
comenzar
  repetir 5
    mover
  Pos (10,10)
  Pos (1,1)
fin
```

```
robot tipo2
comenzar
  mientras (hayFlorEnLaEsquina)
    tomarFlor
  Pos (10,10)
  Pos (2,2)
fin
```

Los robots podrían chocarse al querer posicionarse en la esquina (10,10)





EJERCICIOS PARA ANALIZAR

```
programa solucion2
areas
  area1: AreaPC(1,1,100,100)

robots
  robot tipo1
  comenzar
  ...
  fin
```

```
robot tipo2
comenzar
...
fin
```

```
variables
  robot1: tipo1
  robot2: tipo2
...
```

```
robot tipo1
comenzar
  repetir 5
    mover
  BloquearEsquina(10,10)
  Pos (10,10)
  LiberarEsquina(10,10)
  Pos (1,1)
fin
```

```
robot tipo2
comenzar
  mientras (hayFlorEnLaEsquina)
    tomarFlor
  BloquearEsquina(10,10)
  Pos (10,10)
  LiberarEsquina(10,10)
  Pos (2,2)
fin
```

Los robots podrían chocarse al querer posicionarse en la esquina (10,10)





EJERCICIOS PARA ANALIZAR

```
programa solucion2
areas
  area1: AreaPC(1,1,100,100)
```

```
robots
  robot tipo1
  comenzar
  ...
  fin
```

```
robot tipo2
  comenzar
  ...
  fin
```

```
variables
  robot1: tipo1
  robot2: tipo2
  ...
```

```
robot tipo1
  comenzar
    BloquearEsquina(10,10)
  repetir 5
    mover
    Pos (10,10)
    Pos (1,1)
    LiberarEsquina(10,10)
  fin
```

```
robot tipo2
  comenzar
    BloquearEsquina(10,10)
  mientras (hayFlorEnLaEsquina)
    tomarFlor
    Pos (10,10)
    Pos (2,2)
    LiberarEsquina(10,10)
  fin
```

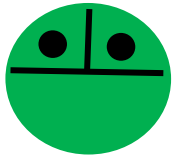
Los recursos
deben ser
bloqueados el
menor tiempo
posible que sea
necesario





COMUNICACIÓN – Memoria Compartida

ROBOT robot1



Realizar código seguro

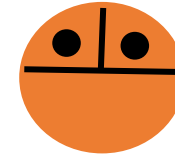
Bloquear la esquina compartida

Realizar la tarea en esa esquina

Moverme a una esquina segura

Liberar la esquina compartida

ROBOT robot2



Realizar código seguro

Bloquear la esquina compartida

Realizar la tarea en esa esquina

Moverme a una esquina segura

Liberar la esquina compartida





COMUNICACIÓN – Memoria Compartida

```
programa solucion2
areas
    area1: AreaPC(1,1,100,100)
```

```
robots
    robot tipo1
    comenzar
    ...
    fin
```

```
robot tipo2
    comenzar
    ...
    fin
```

```
variables
    robot1: tipo1
    robot2: tipo2
    ...
```

```
robot tipo1
comenzar
    repetir 5
        mover
        BloquearEsquina(10,10)
        Pos (10,10)
        Pos (1,1)
        LiberarEsquina(10,10)
fin
```

```
robot tipo2
comenzar
    mientras (hayFlorEnLaEsquina)
        tomarFlor
        BloquearEsquina(10,10)
        Pos (10,10)
        Pos (2,2)
        LiberarEsquina(10,10)
fin
```