

Paradigmas Avanzados de Programación

Objetos activos

Juan Francisco Díaz Frias

Maestría en Ingeniería, Énfasis en Ingeniería de Sistemas y Computación
Escuela de Ingeniería de Sistemas y Computación,
home page: <http://eisc.univalle.edu.co>
Universidad del Valle - Cali, Colombia

Plan

- 1 **Objetos activos**
- 2 Un ejemplo sencillo
- 3 El problema de Flavio Josefo

Plan

- 1 Objetos activos
- 2 Un ejemplo sencillo
- 3 El problema de Flavio Josefo

Plan

- 1 Objetos activos
- 2 Un ejemplo sencillo
- 3 El problema de Flavio Josefo

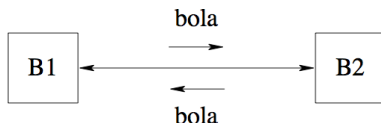
Objetos activos

¿Qué son?

- Un objeto activo es un objeto puerto cuyo comportamiento se define por medio de una clase.
- Objeto activo \equiv (Puerto, Hilo que lee del puerto, objeto).
- Combinan las capacidades de la POO (incluyendo polimorfismo y herencia) y las capacidades de la concurrencia por paso de mensajes (incluyendo concurrencia e independencia de los objetos).
- Los otros objetos los llamaremos pasivos.

Objetos activos

Ejemplo (1)



Ejemplo (3)

```

B1={CrearObjActivo
    JuegoDeBola inic(B2) }
B2={CrearObjActivo
    JuegoDeBola inic(B1) }
{B1 bola}
  
```

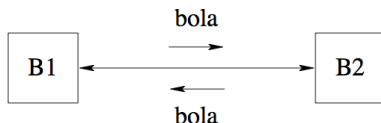
Ejemplo (2)

```

class JuegoDeBola
  attr otro cont:0
  meth inic(Otro)
    otro:=Otro
  end
  meth bola
    cont:=@cont+1
    {@otro bola}
  end
  meth obt(X)
    X=@cont
  end
end
  
```

Objetos activos

Ejemplo (1)



Ejemplo (3)

```

B1={CrearObjActivo
    JuegoDeBola inic(B2)}
B2={CrearObjActivo
    JuegoDeBola inic(B1)}
{B1 bola}
  
```

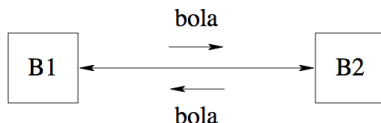
Ejemplo (2)

```

class JuegoDeBola
  attr otro cont:0
  meth inic(Otro)
    otro:=Otro
  end
  meth bola
    cont:=@cont+1
    {@otro bola}
  end
  meth obt(X)
    X=@cont
  end
end
  
```

Objetos activos

Ejemplo (1)



Ejemplo (3)

```

B1={CrearObjActivo
    JuegoDeBola inic(B2)}
B2={CrearObjActivo
    JuegoDeBola inic(B1)}
{B1 bola}
  
```

Ejemplo (2)

```

class JuegoDeBola
  attr otro cont:0
  meth inic(Otro)
    otro:=Otro
  end
  meth bola
    cont:=@cont+1
    {@otro bola}
  end
  meth obt(X)
    X=@cont
  end
end
  
```


Objetos activos

La abstracción `CrearObjActivo`

```
fun {CrearObjActivo Clase Inic}  
  Obj={New Clase Inic}  
  P  
in  
  thread S in  
    {NewPort S P}  
    for M in S do {Obj M} end  
  end  
  proc {$ M} {Send P M} end  
end
```

Ganamos asincronía

A diferencia de lo que sucede con los objetos pasivos, la invocación de un objeto activo es asincrónica: la invocación termina inmediatamente, sin esperar a que el mensaje haya sido manejado.

Objetos activos

La abstracción `CrearObjActivo`

```

fun {CrearObjActivo Clase Inic}
  Obj={New Clase Inic}
  P
in
  thread S in
    {NewPort S P}
    for M in S do {Obj M} end
  end
  proc {$ M} {Send P M} end
end

```

Ganamos asincronía

A diferencia de lo que sucede con los objetos pasivos, la invocación de un objeto activo es asincrónica: la invocación termina inmediatamente, sin esperar a que el mensaje haya sido manejado.

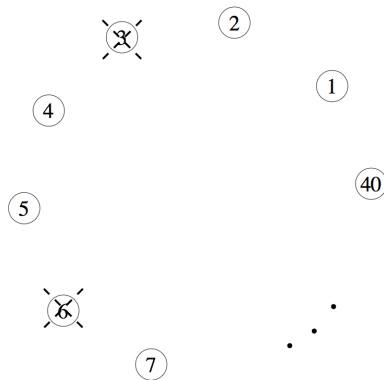
Objetos activos: El problema de Flavio Josefo

Gráficamente ...

La anécdota histórica

Flavio Josefo fue un historiador romano de origen judío.

Durante las guerras judío-romanas del primer siglo D.C., estaba en una cueva con otros soldados, cuarenta en total, rodeado por las tropas enemigas romanas. Ellos acordaron suicidarse, colocándose inicialmente en una fila en forma de círculo, y contando de tres en tres a partir del primero. Cada soldado que quedaba de tercero, se suicidaba, y el círculo se hacía más pequeño. Josefo, quien no quería morir, manipuló las cosas para colocarse, en el inicio, en la posición del último en suicidarse, que, al no hacerlo, resultó ser en realidad la posición del sobreviviente.



Objetos activos: El problema de Flavio Josefo

Versión general

n soldados numerados del 1 al n y cada k -ésimo soldado debe ser eliminado. El conteo comienza desde el primer soldado. ¿Cuál es la posición inicial del sobreviviente?

Modelo con objetos activos

- Soldado \equiv objeto activo
- Círculo de objetos, donde cada objeto conoce sus dos vecinos.
- Un mensaje `matar(x s)` circula alrededor del círculo, donde x cuenta el número de objetos vivos que ha atravesado el mensaje y s contiene el número total de objetos vivos restantes.
- Inicialmente, el mensaje `matar(1 N)` se envía al primer objeto.

Objetos activos: El problema de Flavio Josefo

Versión general

n soldados numerados del 1 al n y cada k -ésimo soldado debe ser eliminado. El conteo comienza desde el primer soldado. ¿Cuál es la posición inicial del sobreviviente?

Modelo con objetos activos

- Soldado \equiv objeto activo
- Círculo de objetos, donde cada objeto conoce sus dos vecinos.
- Un mensaje `matar(x s)` circula alrededor del círculo, donde x cuenta el número de objetos vivos que ha atravesado el mensaje y s contiene el número total de objetos vivos restantes.
- Inicialmente, el mensaje `matar(1 N)` se envía al primer objeto.

Objetos activos: El problema de Flavio Josefo

Protocolo de paso de mensajes

Cuando el objeto i recibe el mensaje `matar(X S)` hace lo siguiente:

- Si está marcado como vivo y $S = 1$, entonces él es el sobreviviente. El objeto indica esto ligando una variable global. No se reenvían más mensajes.
- Si está marcado como vivo y $x \bmod k = 0$, entonces el objeto se marca como muerto y envía el mensaje `matar($X+1$ $S-1$)` al objeto siguiente en el círculo.
- Si está marcado como vivo y $x \bmod k \neq 0$, entonces envía el mensaje `matar($X+1$ S)` al objeto siguiente en el círculo.
- Si está marcado como muerto, entonces reenvía el mensaje `matar(X S)` al objeto siguiente en el círculo.

Objetos activos: El problema de Flavio Josefo

La clase Víctima (1)

```
class Víctima
  attr ident paso sobrev suc pred vivo:true
  meth inic(I K L) ident:=I paso:=K sobrev:=L end
  meth asgnSuc(S) suc:=S end
  meth asgnPred(P) pred:=P end
  ...
```

Objetos activos: El problema de Flavio Josefo

La clase Víctima (2)

```
class Víctima
...
meth matar(X S)
  if @vivo then
    if S==1 then @sobrev=@ident
    elseif X mod @paso==0 then
      vivo:=false
      {@pred nuevosuc(@suc) }
      {@suc nuevopred(@pred) }
      {@suc matar(X+1 S-1) }
    else
      {@suc matar(X+1 S) }
    end
  else {@suc matar(X S) } end
end
...
```


Objetos activos: El problema de Flavio Josefo

La clase Víctima (3)

```
class Víctima
  ...
  meth nuevosuc(S)
    if @vivo then suc:=S
    else {@pred nuevosuc(S)} end
  end
  meth nuevopred(P)
    if @vivo then pred:=P
    else {@suc nuevopred(P)} end
  end
end
```

Objetos activos: El problema de Flavio Josefo

La función Josefo

```
fun {Josefo N K}  
  A={NewArray 1 N null}  
  Sobrev  
in  
  for I in 1..N do  
    A.I:={CrearObjActivo Víctima  
          inic(I K Sobrev)}  
end  
  for I in 2..N do {A.I asgnPred(A.(I-1))} end  
  {A.1 asgnPred(A.N)}  
  for I in 1..(N-1) do {A.I asgnSuc(A.(I+1))} end  
  {A.N asgnSuc(A.1)}  
  {A.1 matar(1 N)}  
  Sobrev  
end
```