

Programación con memoria compartida

Andrés Becerra Sandoval
abecerra@puj.edu.co

20 de abril de 2007

Contenido

Introducción

TADs imperativos

Arreglos

Diccionarios

Tuple Spaces

Definición

Ejemplo con Tuple Spaces

Implementación

Locks

Definición

Implementación

Monitores

Definición

Ejemplo

Implementación

Asignación

- ▶ $V = \{\text{NewCell Valor_inicial}\}$

Asignación

- ▶ $V = \{\text{NewCell Valor_inicial}\}$
- ▶ $V := \text{Nuevo_valor}$

Asignación

- ▶ $V = \{\text{NewCell Valor_inicial}\}$
- ▶ $V := \text{Nuevo_valor}$
- ▶ $@V$ El valor de la celda

Asignación

- ▶ $V = \{\text{NewCell Valor_inicial}\}$
- ▶ $V := \text{Nuevo_valor}$
- ▶ $@V$ El valor de la celda
- ▶ $\{\text{Exchange } V \text{ Old New}\}$

Asignación

- ▶ $V = \{\text{NewCell Valor_inicial}\}$
- ▶ $V := \text{Nuevo_valor}$
- ▶ $@V$ El valor de la celda
- ▶ $\{\text{Exchange } V \text{ Old New}\}$
 - ▶ Liga Old al valor viejo de la celda

Asignación

- ▶ $V = \{\text{NewCell Valor_inicial}\}$
- ▶ $V := \text{Nuevo_valor}$
- ▶ $@V$ El valor de la celda
- ▶ $\{\text{Exchange } V \text{ Old New}\}$
 - ▶ Liga Old al valor viejo de la celda
 - ▶ Pone el valor New en la celda actualizándola

Ejemplo Cola (Declarativa)

```
fun {NewQueue}
  X C={NewCell q(0 X X)}
  proc {Insert X}
    N S E1 in
      q(N S X|E1)=@C
      C:=q(N+1 S E1)
  end
  fun {Delete}
    N S1 E X in
      q(N X|S1 E)=@C
      C:=q(N-1 S1 E)
      X
  end
in
  queue(insert:Insert delete:Delete)
end
```

Ejemplo Cola (Imperativa)

```
fun {NewQueue}
  X C={NewCell q(0 X X)}
  proc {Insert X}
    N S E1 N1 in
      {Exchange C q(N S X|E1) q(N1 S E1)}
      N1=N+1
  end
  fun {Delete}
    N S1 E N1 X in
      {Exchange C q(N X|S1 E) q(N1 S1 E)}
      N1=N-1
      X
  end
in
  queue(insert:Insert delete:Delete)
end
```

Ejemplo Cola (con Locks)

```
fun {NewQueue}
  X C={NewCell q(0 X X)}      L={NewLock}
  proc {Insert X}
  N S E1 in
    lock L then
      q(N S X|E1)=@C
      C:=q(N+1 S E1)
    end
  end
  fun {Delete}
  N S1 E X in
    lock L then
      q(N X|S1 E)=@C
      C:=q(N-1 S1 E)
    end
    X
  end
in
  queue(insert:Insert delete:Delete)
end
```

Ejemplo Cola (con clase)

```
class Queue
  attr queue
  prop locking
  meth init
    queue:=q(0 X X)
  end
  meth insert(X)
    lock N S E1 in
      q(N S X|E1)=@queue
      queue:=q(N+1 S E1)
    end
  end
  meth delete(X)
    lock N S1 E in
      q(N X|S1 E)=@queue
      queue:=q(N-1 S1 E)
    end
  end
end
```

Contenido

Introducción

TADs imperativos

Arreglos

Diccionarios

Tuple Spaces

Definición

Ejemplo con Tuple Spaces

Implementación

Locks

Definición

Implementación

Monitores

Definición

Ejemplo

Implementación

Arreglos

- ▶ Un arreglo es un conjunto de celdas indexadas
- ▶ $A = \{\mathbf{NewArray} \ +LowI \ +HighI \ InitX \}$
- ▶ $\{\mathbf{Array.put} \ A \ I \ X\} \triangleright A_i := X$
- ▶ $\{\mathbf{Array.get} \ A \ I \} \triangleright A_i$
- ▶ $\{\mathbf{Array.exchange} \ A \ +I \ OldVal \ NewVal\} \triangleright$ Actualiza A_i

Contenido

Introducción

TADs imperativos

Arreglos

Diccionarios

Tuple Spaces

Definición

Ejemplo con Tuple Spaces

Implementación

Locks

Definición

Implementación

Monitores

Definición

Ejemplo

Implementación

Diccionarios

- ▶ Un diccionario es una lista de parejas (llave, valor)
- ▶ $D = \{\mathbf{NewDictionary}\}$
- ▶ $\{\text{Dictionary.put } D +LI X\}$ En la llave LI coloca el valor X
- ▶ $\{\text{Dictionary.get } +D +LI X\}$ Liga X al valor que tiene la llave LI

Contenido

Introducción

TADs imperativos

Arreglos

Diccionarios

Tuple Spaces

Definición

Ejemplo con Tuple Spaces

Implementación

Locks

Definición

Implementación

Monitores

Definición

Ejemplo

Implementación

Tuple Spaces

- ▶ Un espacio de tuplas es un conjunto de tuplas etiquetadas
- ▶ **write:** {TS write T} Escribe la tupla T en el espacio TS
- ▶ **read:** {TS read(L T)} Espera a que el espacio TS tenga una tupla con etiqueta L, cuando suceda esto liga T a dicha tupla

Contenido

Introducción

TADs imperativos

Arreglos

Diccionarios

Tuple Spaces

Definición

Ejemplo con Tuple Spaces

Implementación

Locks

Definición

Implementación

Monitores

Definición

Ejemplo

Implementación

Cola (con Tuple Spaces)

```
fun {NewQueue}
  X TS={New TupleSpace init}
  proc {Insert X}
    N S E1 in
      {TS read(q q(N S X|E1))}
      {TS write(q(N+1 S E1))}
  end
  fun {Delete}
    N S1 E X in
      {TS read(q q(N X|S1 E))}
      {TS write(q(N-1 S1 E))}
      X
  end
in
  {TS write(q(0 X X))}
  queue(insert:Insert delete:Delete)
end
```

Contenido

Introducción

TADs imperativos

Arreglos

Diccionarios

Tuple Spaces

Definición

Ejemplo con Tuple Spaces

Implementación

Locks

Definición

Implementación

Monitores

Definición

Ejemplo

Implementación

Implementación de Tuple Spaces

```
class TupleSpace
  prop locking
  attr tupledict
  meth init tupledict:={NewDictionary} end
  meth EnsurePresent(L)
    if {Not {Dictionary.member @tupledict L}}
    then @tupledict.L:={NewQueue} end
  end
  meth Cleanup(L)
    if {@tupledict.L.size}==0
    then {Dictionary.remove @tupledict L} end
  end
  meth write(Tuple)
    lock L={Label Tuple} in
      {self EnsurePresent(L)}
      {@tupledict.L.insert Tuple}
    end
  end
end
```

Implementación de Tuple Spaces

```
meth write(Tuple)
  lock L={Label Tuple} in
    {self EnsurePresent(L)}
    {@tupledict.L.insert Tuple}
  end
end
meth read(L ?Tuple)
  lock
    {self EnsurePresent(L)}
    {@tupledict.L.delete Tuple}
    {self Cleanup(L)}
  end
  {Wait Tuple}
end
```

Implementación de Tuple Spaces

```
meth readnonblock(L ?Tuple ?B)
  lock
    {self EnsurePresent(L)}
    if {@tupledict.L.size}>0 then
      {@tupledict.L.delete Tuple} B=true
    else B=false end
    {self Cleanup(L)}
  end
end
```


Contenido

Introducción

TADs imperativos

Arreglos

Diccionarios

Tuple Spaces

Definición

Ejemplo con Tuple Spaces

Implementación

Locks

Definición

Implementación

Monitores

Definición

Ejemplo

Implementación

Locks - Candados

- ▶ Los candados permiten proteger varias regiones críticas
- ▶ Podemos implementarlos, aunque Oz tiene una implementación a bajo nivel (C++) más eficiente

Contenido

Introducción

TADs imperativos

Arreglos

Diccionarios

Tuple Spaces

Definición

Ejemplo con Tuple Spaces

Implementación

Locks

Definición

Implementación

Monitores

Definición

Ejemplo

Implementación

Lock (no reentrante)

```
fun {SimpleLock}
  Token={NewCell unit}
  proc {Lock P}
    Old New in
      {Exchange Token Old New}
      {Wait Old}
      {P}
      New=unit
    end
  in
    'lock' ('lock':Lock)
  end
end
```

Lock (no reentrante)

Con manejo de excepciones:

```
fun {CorrectSimpleLock}
  Token={NewCell unit}
  proc {Lock P}
    Old New in
      {Exchange Token Old New}
      {Wait Old}
      try {P} finally New=unit end
    end
  in
    'lock' ('lock':Lock)
  end
```

Lock (reentrante)

```
fun {NewLock}
  Token={NewCell unit}
  CurThr={NewCell unit}
  proc {Lock P}
    if {Thread.this}==@CurThr then
      {P}
    else Old New in
      {Exchange Token Old New}
      {Wait Old}
      CurThr:={Thread.this}
      try {P} finally
        CurThr:=unit
        New=unit
      end
    end
  end
end
in
  'lock'('lock':Lock)
end
```

Contenido

Introducción

TADs imperativos

Arreglos

Diccionarios

Tuple Spaces

Definición

Ejemplo con Tuple Spaces

Implementación

Locks

Definición

Implementación

Monitores

Definición

Ejemplo

Implementación

Monitores

- ▶ Un monitor es un tipo abstracto de datos que protege varias regiones críticas (al igual que un lock), pero permite un mayor control al programador (son más expresivos), ya que los hilos se pueden *notificar* eventos entre si
- ▶ Operaciones:
 - ▶ **Lock**
 - ▶ **Wait**
 - ▶ **Notify**
 - ▶ **NotifyAll**

Contenido

Introducción

TADs imperativos

Arreglos

Diccionarios

Tuple Spaces

Definición

Ejemplo con Tuple Spaces

Implementación

Locks

Definición

Implementación

Monitores

Definición

Ejemplo

Implementación

Bounded Buffer

Un ejemplo familiar:

```
class Buffer
  attr
    buf first last n i
  meth init(N)
    buf:={NewArray 0 N-1 null}
    first:=0 last:=0 n:=N i:=0
  end
  meth put(X)
    ... %wait until i<n
    %now add an element:
    @buf.@last:=X
    last:=(@last+1) mod @n
    i:=@i+1
  end
  meth get(X)
    ... %wait until i>0
    %now remove an element:
    X=@buf.@first
    first:=(@first+1) mod @n
```

Bounded Buffer (con monitor)

Un monitor ayuda a modelar:

```
class Buffer
  attr m buf first last n i
  meth init(N)
    m:={NewMonitor}
    buf:={NewArray 0 N-1 null}
    n:=N i:=0 first:=0 last:=0
  end
  meth put(X)
    {@m.'lock' proc {$}
      if @i>=@n then {@m.wait} {self put(X)}
      else
        @buf.@last:=X
        last:=(@last+1) mod @n
        i:=@i+1
        {@m.notifyAll}
      end
    end}
  end
end
```

Bounded Buffer (con monitor)

```
meth get(X)
  {@m. 'lock' proc {$}
    if @i==0 then {@m.wait} {self get(X)}
    else
      X=@buf. @first
      first :=( @first+1) mod @n
      i :=@i-1
      {@m.notifyAll}
    end
  end}
end
end
```

Contenido

Introducción

TADs imperativos

Arreglos

Diccionarios

Tuple Spaces

Definición

Ejemplo con Tuple Spaces

Implementación

Locks

Definición

Implementación

Monitores

Definición

Ejemplo

Implementación

Implementación de Monitor

```
fun {NewMonitor}
  Q={NewQueue}
  L={NewGRLock}
  proc {LockM P}
    {L.get} try {P} finally {L.release} end
  end
  proc {WaitM}
  X in
    {Q.insert X} {L.release} {Wait X} {L.get}
  end
  proc {NotifyM}
  U={Q.deleteNonBlock} in
    case U of [X] then X=unit else skip end
  end
  proc {NotifyAllM}
  L={Q.deleteAll} in
    for X in L do X=unit end
  end
in
  monitor('lock':LockM wait:WaitM notify:NotifyM
```

Cola extendida

```
fun {NewQueue}
  ...
  fun {Size}
    lock L then @C.1 end
  end
  fun {DeleteAll}
    lock L then
      X q(_ S E)=@C in
        C:=q(0 X X)
        E=nil S
      end
    end
  fun {DeleteNonBlock}
    lock L then
      if {Size}>0 then [{Delete}] else nil end
    end
  end
end
in
  queue(insert:Insert delete:Delete size:Size
        deleteAll:DeleteAll deleteNonBlock:DeleteNonBlock)
```

Lock(get-release)

```
fun {NewGRLock}
  Token1={NewCell unit}
  Token2={NewCell unit}
  CurThr={NewCell unit}
  proc {GetLock}
    if {Thread.this}\=@CurThr then Old New in
      {Exchange Token1 Old New}
      {Wait Old}
      Token2:=New
      CurThr:={Thread.this}
    end
  end
  proc {ReleaseLock}
    CurThr:=unit
    unit=@Token2
  end
in
  'lock'(get:GetLock release:ReleaseLock)
end
```