

Processus concurrents et parallélisme

Chapitre 2 - Concurrence et Parallélisme

Exemples

Gabriel Girard

2007

Chapitre 2 - Exemples

1 Exemples

2 Exemples de coroutine

Chapitre 2 - Exemples

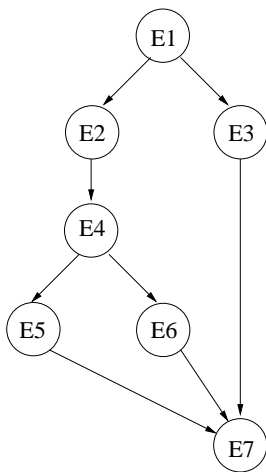
1 Exemples

2 Exemples de coroutine

Exemple 1

```
a = x + y;      // E1
b = z + a;      // E2
c = a - b;      // E3
d = b + 1;      // E4
e = d + 4;      // E5
f = d - 1;      // E6
r = c * (e - f) // E7
```

Exemple 1 : graphe de précédance



Example 1 : fork/join

```
a = x + y;
compte := 3;
fork L1;
b = z + a;
fork L2;
e = d + 4;
goto L3;
L2: f = d - 1;
goto L3;
L1: c = a - b;
d = b + 1;
join compte;
L3: r = c * (e - f)
```

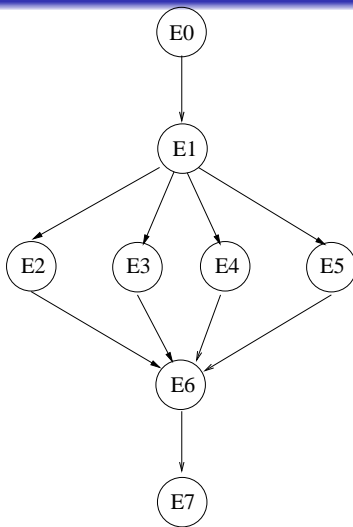
Exemple 1 : parbegin/parend

```
a = x + y;
parbegin
  c = a - b;
  begin
    b = z + a;
    d = b + 1;
    parbegin
      e = d + 4;
      f = d - 1;
    parend;
  end;
parend;
r = c * (e - f)
```

Exemple 2

```
z = 2;           // E0
a = x + y;      // E1
b = z + a;      // E2
r = a ** a;     // E3
c = a - 1;     // E4
d = z * a;     // E5
r = d * r;     // E6
printf("",r..); // E7
```


Exemple 2 : graphe de précédance



Exemple 2 : fork/join

```
z = 2;
a = x + y;
compte := 4;
fork L1;
fork L2;
fork L3;
b = z + a;
goto L4;
L1: r = a ** a;
    goto L4;
L2: c = a - 1;
    goto L4;
L3: d = z * a;
L4: join compte;
    r = (d * r) + (c ** b)
    printf("",r..);
```

Exemple 2 : parbegin/parend

```
z = 2;
a = x + y;
parbegin
    b = z + a;
    r = a ** a;
    c = a - 1;
    d = z * a;
parend;
r = (d * r) + (c ** b);
printf("",r..);
```

Chapitre 2 - Exemples

1 Exemples

2 Exemples de coroutine

Producteur/consommateur

```
var q := new queue

coroutine produce
  loop
    while q is not full
      create some new items
      add the items to q
    yield to consume

coroutine consume
  loop
    while q is not empty
      remove some items from q
      use the items
    yield to produce
```

Exemple 1 dans le langage LUA

```
function gimmeval()
    me = 1243
    while (me > 1234) do
        coroutine.yield()
        me = me - 1
        print ("Dans coroutine")
    end
end

-- main code
print ("Simple co-routine")
instream = coroutine.create(gimmeval)
while coroutine.status(instream) ~= "dead" do
    coroutine.resume(instream)
    print (me, "Here - at ")
end
...

```

Sortie...

```
Simple co-routine
1243   Here - at
Dans coroutine
1242   Here - at
Dans coroutine
1241   Here - at
Dans coroutine
1240   Here - at
Dans coroutine
1239   Here - at
Dans coroutine
1238   Here - at
...
```

Exemple 2 dans le langage LUA

```
> function foo()
>>   print("foo", 1)
>>   coroutine.yield()
>>   print("foo", 2)
>> end
> co = coroutine.create(foo)
> = type(co)
thread
> = coroutine.status(co)
suspended
> = coroutine.resume(co)
foo 1
> = coroutine.resume(co)
foo 2
> = coroutine.status(co)
dead
> = coroutine.resume(co)
false    cannot resume dead coroutine
```