

Fiber の

サンプルをかいてみた

2013/02/02

広島**Ruby**勉強会 #027

Fiber

別名

マイクロスレッド

コルーチン

状態管理を意識せずに行える
ため、協調的処理、イテレー
タ、無限リスト、パイプな
ど、継続状況を持つプログラ
ムが容易に記述できる。

ゲームプログラミングでみたことある

ということで
それ系のサンプルをかいてみた

*

<http://www.youtube.com/watch?v=U0I0VLrBDEA>

もっと派手にすればよかった

どんなコードを書きたいか

10回 下に移動する

10回 右に移動する

10回 上に移動する

10回 左に移動する

```
10.times do  
  down  
end
```

```
10.times do  
  right  
end
```

```
10.times do  
  up  
end
```

```
10.times do  
  left  
end
```

そうはうまくいかぬ

Fiber を使わずかいてみよう

```
def step
  if @time < 10
    @x += 1
  elsif @time >=10 and @time < 20
    @y += 1
  elsif @time >= 20 and @time < 30
    @x -= 1
  elsif @time >= 30 and @time < 40
    @y -= 1
  else
    @x = 0
    @y = 0
    @time = 0
  end
  @time += 1
end
```

状態を管理する必要がある

```
def step
  if @time < 10
    @x += 1
  elsif @time >= 10 and @time < 20
    @y += 1
  elsif @time >= 20 and @time < 30
    @x -= 1
  elsif @time >= 30 and @time < 40
    @y -= 1
  else
    @x = 0
    @y = 0
    @time = 0
  end
  @time += 1
end
```

状態の更新が必要

どの状態か判断が必要

```
def step
  if @time < 10           下に移動する時
    @x += 1
  elsif @time >= 10 and @time < 20  右に移動する時
    @y += 1
  elsif @time >= 20 and @time < 30  上に移動する時
    @x -= 1
  elsif @time >= 30 and @time < 40  左に移動する時
    @y -= 1
  else
    @x = 0
    @y = 0
    @time = 0
  end
  @time += 1
end
```



下に移動する時

右に移動する時

上に移動する時

左に移動する時

Fiber を使うと？

```
def step  
    fiber.resume  
end
```

```
@fiber ||= Fiber.new do
  loop do
    @x = 5
    @y = 5

    10.times do
      @x += 1
      Fiber.yield
    end

    10.times do
      @y += 1
      Fiber.yield
    end

    10.times do
      @x -= 1
      Fiber.yield
    end

    10.times do
      @y -= 1
      Fiber.yield
    end
  end
end
```

```
10.times do  
  down  
end
```

```
10.times do  
  right  
end
```

```
10.times do  
  up  
end
```

```
10.times do  
  left  
end
```

```
@fiber ||= Fiber.new do
  loop do
    @x = 5
    @y = 5

    10.times do
      @x += 1
      Fiber.yield
    end

    10.times do
      @y += 1
      Fiber.yield
    end

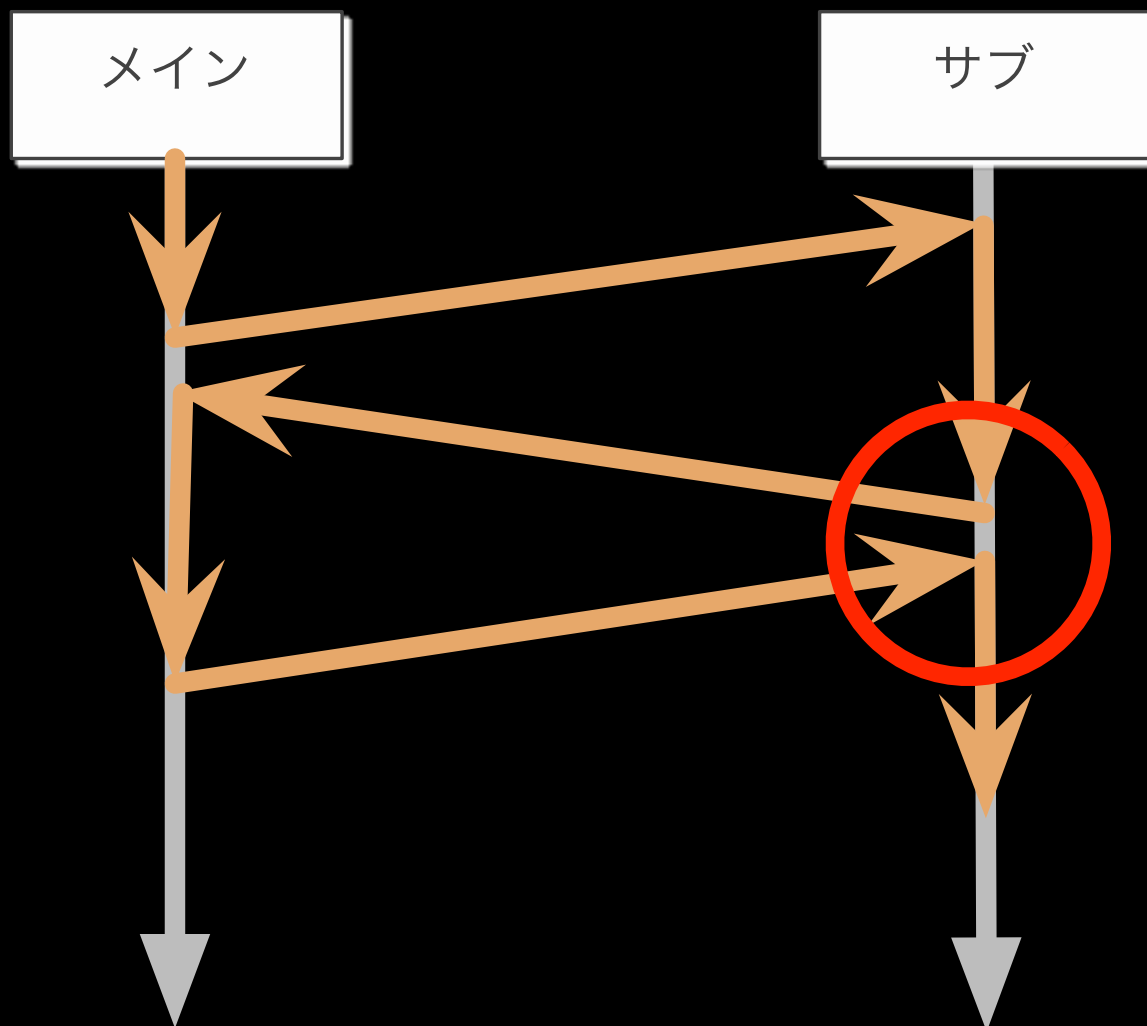
    10.times do
      @x -= 1
      Fiber.yield
    end

    10.times do
      @y -= 1
      Fiber.yield
    end
  end
end
```

概ね希望どおりかける

Fiber.yield を呼んだところで止めたままにできる

再開するには **Fiber#resume** を使用する



止めておいたところ
からそのまま再開で
きる

step メソッドで **Fiber** を再開させるだけ
Fiber の中では必要なところで止まる
状態をそのまま維持

実際にはあちこちまわるけど
プログラムは

真っ直ぐ書ける

おまけ

描画には

OpenGL とかつかってがんばる方法もあるけど
Curses を使いました

Curses

<http://ja.wikipedia.org/wiki/Curses>

curses（カーシス、カーズィス）は
UNIX系システムでの端末制御ライブラリで
ある。

テキストユーザインタフェース（TUI）アプ
リケーションを作成するのに使われる。

名称は“cursor optimization”に由来する。

端末の画面の特定の位置に文字を置く


```
Curses.setpos(x,y)  
Curses.addstr( '*' )
```

ソースコード

<https://github.com/eiel/Fiber-Sample>

ご清聴ありがとうございます