

Lightweight Language Saturday



P3-3

じゃんけんクライアント  
(Ruby版)

喜多 淳一郎

## じゃんけんクライアントの処理の流れ

---

1. じゃんけんサーバにアドレス/ポート番号を通知
  - TCPSocket
2. サーバからのCALLを受けて, 手を送信
  - TCPServer
3. サーバからの勝敗通知を受信
  - UDPSocket

→ これをそのままふつーにRubyで書いてみる

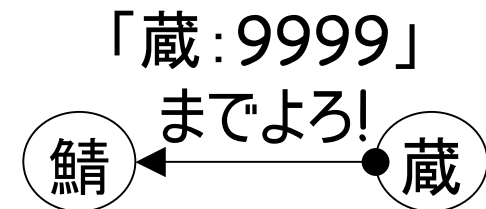
# 1. アドレス/ポート番号を通知

```
TCPSocket.open(server, 2003) do |s|  
  s.print "#{client}:9999¥r¥n"  
end
```

- TCPSocketはIOのサブクラス
- ソケットに関するもろもろの情報もsが保持
  - addrで自分の, peeraddrで相手のソケットの情報(IPアドレス, port等)が取れる
- ruby1.6系では

```
s = TCPSocket.open(server, 2003)  
s.print .....
```

  - としないとダメ → きたはこれで半日はまった...

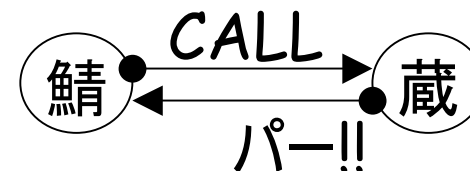


## 2. CALL→手の送信

```
gs = TCPServer.new(client, 9999)
s = gs.accept
while 1 = s.gets
  if /¥ACALL¥r¥n¥z/ === 1
    s.print "PAPER¥r¥n"
  end
end
s.close
```

```
gs = TCPServer.new(client, 9999)
while true
  Thread.start(gs.accept) do |s|
    while 1 = s.gets
      s.puts .....
    end
    s.close
  end
end
end
```

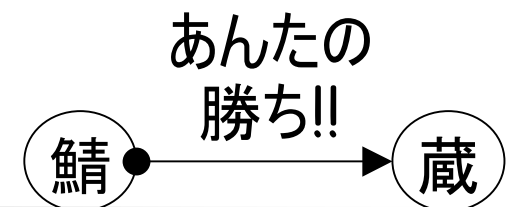
- TCPServerはTCPSocketのサブクラス
  - ただしI/Oには使われない
  - 待ち受け専門
- sはTCPSocketのインスタンス → こいつを使って相手とやりとり
- ちなみに、ふつーのサーバプログラムでは右側のように書く



### 3. 勝敗通知の受信(1/2)

```
s = UDPSocket.new
s.bind(client, 9999)
while l = s.gets # (A)
  result = l[/¥A¥d{3}/]
  show_result(result)
  break if result == CHAMPION or result == LOSE
end
s.close
```

- UDPSocketはIOのサブクラス
- 受信したデータグラムの送信元情報を知りたい場合はUDPSocket#recvfrom
- データグラムの送信はUDPSocket#connect後ソケットに書き込み
- ところでちょっと問題が...
  - もしかしたら勝敗通知は届かないかもしれない
  - すると(A)から先に進めない



### 3. 勝敗通知の受信(2/2)

- 選択肢
  - IO.select
  - IO#fcntlでNon-Blocking IO  
どちらも面倒
  - Thread
    - 今のRubyではThreadを用いても高速化できない
    - むしろやりたいことをシンプルに記述するために利用
    - というか、Thread使うとなんとか楽しい

```
Thread.start(UDPSocket.new) do |s|
  s.bind(client, 9999)
  while 1 = s.gets # (A)
    ...
    # 必要ならbreak
  end
  s.close
  Thread.main.raise("やめれ")
end
```

main thread側では  
rescueでこの例外を補足

```
begin
  # CALL→手の送信
rescue
  終了処理
end
```

# 全体の流れ

## 1. アドレス/ポート番号を通知

```
TCPSocket.open(server, 2003) do |s|
  s.print "#{client}:9999¥r¥n"
end
```

スレッド生成

## 2. CALL → 手の送信

```
while true
  gs = TCPServer.new(client, 9999)
  s = gs.accept
  while l = s.gets
    if /¥ACALL¥r¥n¥z/ === l
      s.print "PAPER¥r¥n"
    end
  end
  s.close
end
```

## 3. 勝敗通知の受信

```
Thread.start(UDPSocket.new) do |s|
  s.bind(client, 9999)
  while l = s.gets
    result = l[/¥A¥d{3}/]
    show_result(result)
    if result == CHAMPION or
       result == LOSE
      break
    end
  end
  s.close
  Thread.main.raise("やめれ")
end
```

終了通知

×