CRYSTAL

CodeCamp | SF2017

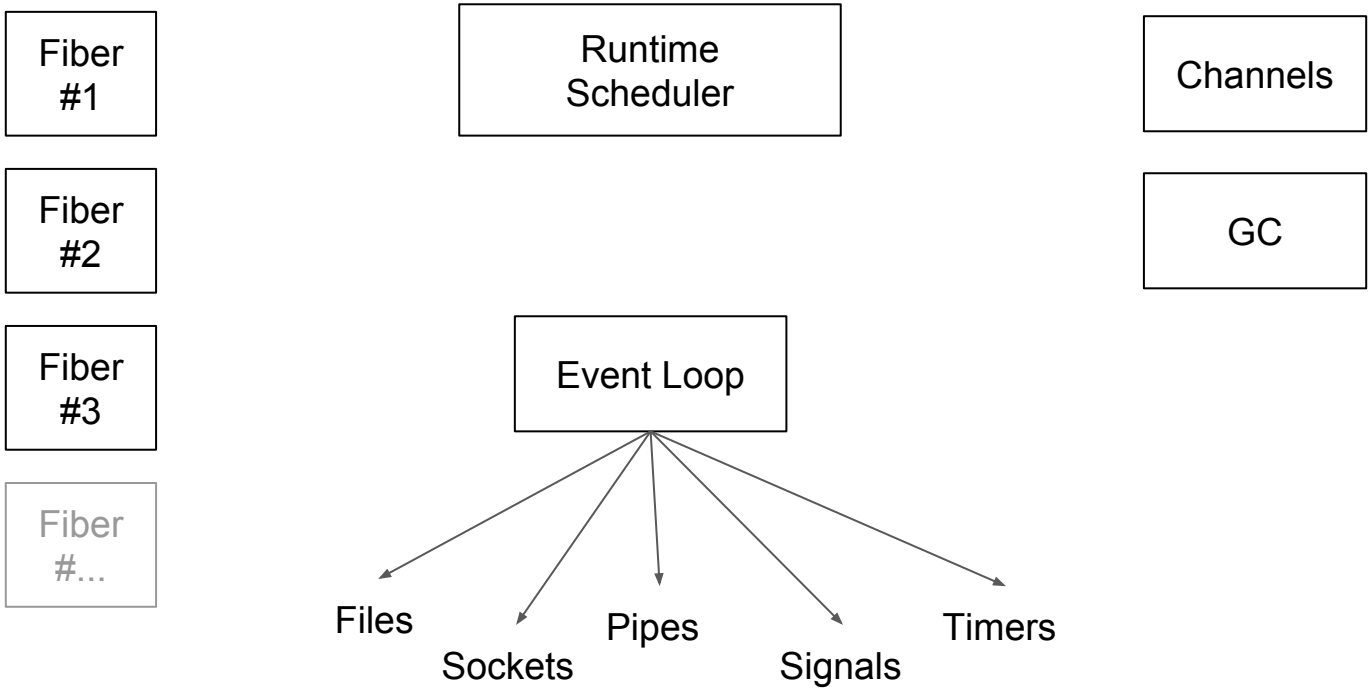
Concurrency and processes

Doing more than one thing at a time

Concurrency Model

- Inspirations: Go and Erlang
- Cooperative scheduling (Fiber)
- Event loop (IO, timers)
- Message channels (CSP)

A Crystal process



Fiber

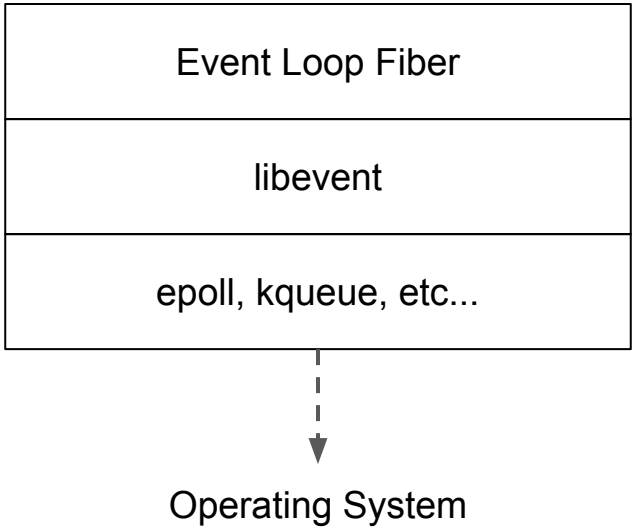
- Cooperative
- Lightweight (from 4KB)
- 32 bit arch.: max 512 fibers
- 64 bit arch.: 2^41 (~2 trillion)

Fibers

Creating a new fiber

```
spawn do
  # ...
end
```

Event Loop


Event Loop Fiber
libevent
epoll, kqueue, etc...
Operating System

Scheduling

```
spawn do
  # ...
  @socket.read(...)
  # ...
end
```

```
spawn do
  # ...
  sleep 5
  # ...
end
```

Channels

Unbuffered channel

```
ch = Channel(Int32).new

spawn do
  ch.send 123
end

spawn do
  x = ch.receive
end
```

Channels

Buffered channel

```
ch = Channel(Int32).new(10)

spawn do
  ch.send 123
end

spawn do
  x = ch.receive
end
```

Waiting on several channels

```
select
when x = ch1.receive
  # ...
when x = ch2.receive
  # ...
when ch3.send(123)
  # ...
end
```

Patterns

Synchronizing

```
ch = Channel(Nil).new

spawn do
  # ...
  ch.send nil
end

ch.receive
```

Patterns

Synchronizing

```
ch = Channel(Nil).new

10.times do
  spawn do
    # ...
    ch.send nil
  end
end

10.times { ch.receive }
```

Patterns

Job queue

```
ch = Channel(Job).new

10.times do
  spawn do
    loop do
      execute_job(ch.receive)
    end
  end
end
```

Higher level constructs

```
d = delay(1) { Process.kill(Signal::KILL, Process.pid) }

# ... long operations ...

d.cancel
```

Higher level constructs

```
f = future { HTTP::Client.get "http://..." }

# ... other actions ...

response = f.get
```

Higher level constructs

```
l = lazy { HTTP::Client.get "http://..." }

spawn { l.get }
spawn { l.get }
```

Higher level constructs

```
r1, r2 = parallel(
  HTTP::Client.get("http://www.google.com"),
  HTTP::Client.get("http://www.bing.com")
)
```

Playing with fibers and channels

Caveat: CPU-bound processes

- How to screw a cooperative scheduling system
- Avoiding long-running tasks

Spawning new processes

- fork/exec
- Awaiting for completion

Inter Process Communication

- Signals
- Pipes
- Sockets
- Files

Multi-thread support

- What about regular threads?
- Support status
- What to expect
- New problems/caveats it introduces

We are Manas.

We build unconventional software_

https://manas.tech