

Elixir, La Turra II

César Guzmán Alpízar

Vipera Ibérica

Octubre de 2020

Index

1 Actor Model and Process

Actor Model

- Each Actor is a process
- Each process performs a specific task
- To tell a process to do something, we need to send a message
- Messages are pattern matched
- Other than that, process do not share any other information

From: Benjamin Tan Wei Hao. "The Little Elixir & OTP Guidebook MEAP v10".

Spawn a process

```
pid = spawn(fn -> 1 + 1 end)
```

Spawn 100 processes

```
Enum.each(1..100, fn _ ->  
  spawn(fn -> 1 + 1 end)  
end)
```

Receive messages

```
receive do
  {:pattern, n} -> IO.puts("The number is #{n}")
  _ -> IO.puts("Discarded message")
end
```

Receive messages

```
pid = spawn(fn ->
  receive do
    {:sum, a, b, pid} -> send(pid, {:result, a + b})
    _ -> IO.puts("Discarded message")
  end
end)
```

Send a message

```
send(pid, message)
```

```
send(pid, {:sum, a, b, self()})
```


Mantain a state with recursion

```
def loop(n) do
  receive do
    {:add, a} ->
      loop(n + a)

    :show_state ->
      IO.puts("The current state is #{n}")
      loop(n)

    _ ->
      IO.puts("Unrecognized error")
      loop(n)
  end
end
```

GenServer

```
defmodule Module do
  use GenServer

  # Client API
  def start_link(default) do
    GenServer.start_link(__MODULE__, default)
  end

  # Server callbacks
  def init(state) do
    {:ok, state}
  end
end
```

Thanks a lot!

Questions?