

# Erlang Academy

## Лекция 6

# План

- Паралеллизм
- Процессы
- Оператор receive
- Типы связей между процессами
- Поведения (behaviours)
- Поведения встроенные в ОТР
- Фреймворк для тестирования Common Test

# Параллелизм

Виды параллелизма:

- Параллельность — взаимодействие параллельных процессов
- Конкурентность — взаимодействие последовательно выполняемых процессов

Конкурентность базируется на обработке прерываний

# Процесс

Процесс Erlang следует рассматривать, как комбинацию стека, кучи и очереди хранящей сообщения (почтовый ящик)

# Процессы

```
1> X = 99.
```

```
99
```

```
2> F = fun() -> X + 1 end.
```

```
#Fun<erl_eval.20.50752066>
```

```
3> Pid = spawn(F).
```

```
<0.37.0>
```

# Процессы

```
1> F = fun() ->
1>     receive {ping, Pid} -> Pid ! pong end
1> end.
#Fun<erl_eval.20.50752066>
2> Pid = spawn(F).
<0.36.0>
3> Pid ! {ping, self()}.
{ping,<0.33.0>}
4> receive
4>     Msg -> io:format("Receive msg:~p~n", [Msg])
4> end.
Receive msg:pong
ok
```

# Рекурсивное чтение сообщений

```
process_loop() ->  
  _ = receive  
    {ping, Pid} ->  
      Pid ! pong;  
end,  
process_loop().
```

# Эволюция процесса

Пример



# Типы связей между процессами

- monitor - односторонняя связь
- link - двусторонняя связь

# Поведения встроенные в OTP

gen\_server

supervisor

application

gen\_statem (Заменял gen\_fsm)

gen\_event

# OTP Design Principles

[OTP Design Principles](#)

# Common Test

[Common Test User's Guide](#)

[Пример тестов](#)