

# **ANEL DE PROCESSOS**

## **UTILIZANDO ERLANG**

Anna Alícia

Caio Andrade

Fabrizio Honda

José Jordan

Vanessa Câmara

# PROBLEMA:

- Escrever um anel de processos.
- Criar  $N$  processos em um anel. Enviar uma mensagem ao redor do anel  $M$  vezes para que um total de  $N * M$  mensagens sejam enviadas.
- Cronometrar quanto tempo isso leva para diferentes valores de  $N$  e  $M$ .

# CÓDIGO:

```
-module(ring).
-export([send/2]).

%% @doc Send M messages through a ring of N processes.
send(M, N) ->
    statistics(runtime),
    H = lists:foldl(
        fun(Id, Pid) -> spawn_link(fun() -> loop(Id, Pid, M) end) end,
        self(),
        lists:seq(N, 2, -1)),
    {_, Time} = statistics(runtime),
    io:format("~p processes spawned in ~p ms~n", [N, Time]),
    statistics(runtime),
    H ! M,
    loop(1, H, M).

loop(Id, Pid, M) ->
    receive
    1 ->
        {_, Time} = statistics(runtime),
        io:format("~p messages sent in ~p ms~n", [M, Time]),
        exit(self(), ok);
    Index ->
        Pid ! Index - 1,
        loop(Id, Pid, M)
    end.
```

# EXPLICAÇÃO:

Um anel de  $N$  processos é criado e a mensagem é enviada por ele  $M$  vezes.

- criamos uma cadeia de processos
- enviamos uma mensagem para o primeiro processo

Ela contém:

- a mensagem
- o id do último processo do anel
- número de vezes que a mensagem deve ser lançada no anel

Criamos um novo processo que irá enviar mensagens para o processo com ID  
Durante o loop, a contagem é repetida, formando uma cadeia de processos.  
O ID do último processo criado é retornado.

Cada processo é executado em loop.

Se um pedido de lançamento for recebido, enviamos para o anel.

Até que o loop se encerre, a mensagem for recebida, colocamos novamente no anel

Ao fim do loop, enviamos um pedido de dado para finalizar os processos

## PARA 100 PROCESSOS E 100 MENSAGENS:

```
Eshell V10.7.2.1 (abort with ^G)
1> c(ring).
{ok,ring}
2> ring:send(100, 100).
100 processes spawned in 0 ms
100 messages sent in 0 ms
** exception exit: ok
```

## PARA 1000 PROCESSOS E 1000 MENSAGENS:

```
3> ring:send(1000, 1000).  
1000 processes spawned in 7 ms  
1000 messages sent in 8 ms  
** exception exit: ok
```

## PARA 10000 PROCESSOS E 1000 MENSAGENS:

```
6> ring:send(1000, 10000).  
10000 processes spawned in 56 ms  
1000 messages sent in 7 ms  
** exception exit: ok
```

## PARA 100000 PROCESSOS E 10000 MENSAGENS:

```
7> ring:send(10000, 100000).  
100000 processes spawned in 549 ms  
10000 messages sent in 57 ms
```