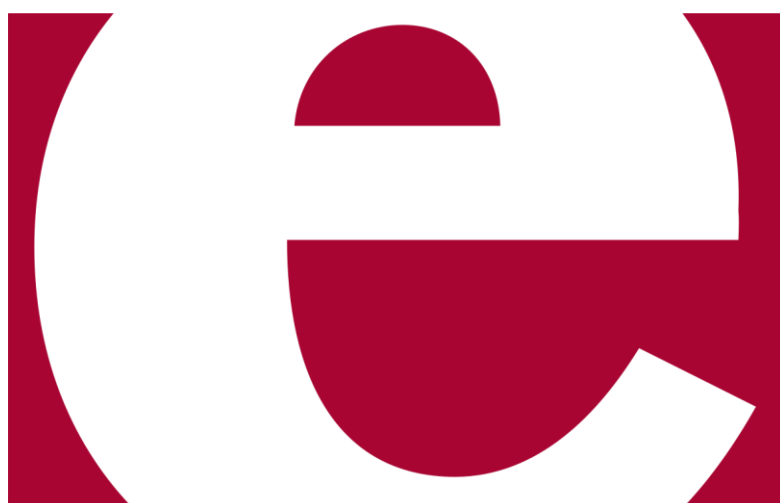


Programação Concorrente



ERLANG

Manuel da Silva Sá

up201805273

Faculdade de Ciências da
Universidade do Porto

Introdução

A linguagem Erlang:

Erlang é uma linguagem de programação interpretada desenvolvida pela empresa Ericsson em 1986, a qual tinha o objetivo de implementar sistemas de tempo real. Em 1998 foi lançada a versão Open-Source. A versão da empresa mencionada acima consistia na interpretação do código numa VM (Virtual Machine), mas em algumas máquinas possui também acesso a um compilador.

O trabalho:

O objetivo deste trabalho, para além de introduzir-nos à linguagem de programação Erlang, consiste em criar uma base de dados e um serviço de servidor/cliente referente à dita base de dados. Esta última contém a informação necessária para guardar requisições de livros, os próprios livros e as pessoas que os requisitaram. Também contém algumas funções que nos ajudam a analisar dados e alterá-los. O servidor vai ter como objetivo receber os pedidos do cliente e retornar uma resposta: um determinado estado da base de dados. O cliente terá a possibilidade de fazer esses mesmos pedidos.

A estrutura

O trabalho está dividido em dois ficheiros: database.erl (a base de dados) e clientserver.erl (o servidor e o cliente). No database temos as três tabelas: pessoas, livros e requisições. Também temos as seguintes funções:

- do_this_once: cria as tabelas;
- start: inicializa a ase de dados;
- reset_tables: limpa alterações anteriores;
- livros: dá-nos os livros requisitados por uma pessoa;
- empréstimos: lista de pessoas que requisitaram um livro;
- Requisitado: verifica se um livro está ou não requisitado;
- Requisitado1- função auxiliar da anterior que cria a lista com o livro em questão;
- códigos: dá-nos o id de livros com o mesmo nome;
- NumRequisicoes: n de livros requisitados pela pessoa;
- add_request: adiciona uma requisição;
- remove_request: remove uma requisição;

No que toca ao clientserver, temos as referências a todas as funções não auxiliares, o rpc e o loop do servidor.

Análise de dados

Exemplos de output:

Tendo em consideração a base de dados criada no ficheiro, foi testado um ou mais exemplos para cada função:

livros(CC):

```
(client@derpro)4> rpc:call('server@derpro',database,livros,[2000]).  
["Centenario: Idade das Cinzas"]
```

Empréstimos(BookName):

```
(client@derpro)5> rpc:call('server@derpro',database,emprestimos,["Contra Batida: Ofensiva Global"]).  
["Farao"]
```

Requisitado(BookCode):

```
(client@derpro)8> rpc:call('server@derpro',database,requisitado,[1]).  
true
```

```
(client@derpro)9> rpc:call('server@derpro',database,requisitado,[2]).  
false
```

Códigos(BookName):

```
(client@derpro)10> rpc:call('server@derpro',database,codigos,["Attack do Titã: Temporada Final"]).  
[6,5]
```

NumRequisições(CC):

```
(client@derpro)14> rpc:call('server@derpro',database,numRequisicoes,[2000]).  
1
```

Add_Request(CC,BookId):

```
(client@derpro)16> rpc:call('server@derpro',database,add_request,[2000,4]).  
{atomic,ok}  
(client@derpro)17> rpc:call('server@derpro',database,livros,[2000]).  
["Centenario: Idade das Cinzas","O Cavaleiro de Arkham"]
```

Remove_Request(CC,BookId):

```
(client@derpro)16> rpc:call('server@derpro',database,add_request,[2000,4]).  
{atomic,ok}  
(client@derpro)17> rpc:call('server@derpro',database,livros,[2000]).  
["Centenario: Idade das Cinzas","O Cavaleiro de Arkham"]  
(client@derpro)18> rpc:call('server@derpro',database,remove_request,[2000,4]).  
{atomic,ok}  
(client@derpro)19> rpc:call('server@derpro',database,livros,[2000]).  
["Centenario: Idade das Cinzas"]
```

Nota: o servidor não permite adicionar uma requisição de um livro já requisitado.

Conclusões finais:

Após a realização de todo o trabalho relatado acima, retiro que, como estudante, obtive os conhecimentos necessários para me abrir os horizontes a nível profissional, no que toca à linguagem Erlang. Pessoalmente prefiro a programação em C++ ou Java, que são linguagens compiladas e imperativas. No entanto, Erlang fez-me descobrir um maior interesse no que toca a linguagens semelhantes a ela. Sinto-me mais confortável a programar em Erlang do que em Haskell ou Prolog por exemplo. Quem sabe, um dia talvez haja a possibilidade de aprofundar as capacidades de programação esta linguagem no futuro.