

Concurrent programming in Erlang

Grupo de Sistemas Distribuídos
Universidade do Minho

Objectives

Exploring Erlang concurrency primitives. Implementing concurrent abstractions for use by client processes, controlled by server processes.

Mechanisms

Process creation, message send, message receive.

Tasks

1. Write a module which implements a login manager, which allows account creation, and controls user login/logout. It should make available the functions:

```
create_account (Username, Passwd) -> ok | user_exists  
close_account (Username, Passwd) -> ok | invalid  
login (Username, Passwd) -> ok | invalid  
logout (Username) -> ok  
online() -> [Username]
```

2. Write a module which implements a read/write lock abstraction, which guarantees absence of starvation. Use select receive with pattern matching. It should make available the functions:

```
create() -> Lock  
acquire(Lock, read | write) -> ok  
release(Lock) -> ok
```

3. Write an alternative implementation for the read/write lock without using selective receive.