Pavel Dvořák, FI MU

ERLANG
A CONCURRENT PROGRAMMING LANGUAGE

A Brief History

- In the eighties, the Ericsson company needed a very reliable and highly parallel programming language for their telephone exchanges.
- Research and development was started by Joe Armstrong and Bjarne Däcker in 1986; the first version was written in Prolog.
- The language was open sourced in 1998 and it is widely used nowadays.

Main Features Overview

- general-purpose programming language
- mostly interpreted (via virtual machine)
- functional paradigm
- strict evaluation
- strong dynamic type system
- fault tolerant (up to *nine nines*)
- hot swapping
- concurrent programming

Concurrency

- Instead of threads, lightweight processes are used. They live in a virtual machine.
- The processes communicate with each other by passing asynchronous messages.
- There is no memory sharing, the data must be copied. Consequently, the applications do not have to bother with synchronization.
- It is very easy to scale Erlang programs.

Elementary Examples (I)

Factorial

```
-module(example).
-export([fact/1]).

fact(0) -> 1;
fact(N) when N > 0 -> N * fact(N - 1).
```

Elementary Examples (II)

Quicksort

```
quicksort([]) -> [];
quicksort([Pivot|Rest]) ->
    quicksort([Smaller || Smaller <- Rest,
                Smaller <= Pivot])
    ++ [Pivot] ++
    quicksort([Larger || Larger <- Rest,
                Larger > Pivot]).
```

Software Written in Erlang

eJabberd

- XMPP server
- reliability, modules, clustering
- Nokia Ovi, Facebook Chat

CouchDB

- document-oriented DBMS
- collection of JSON documents, REST API
- Apache Software Foundation, IBM

Sources

- http://erlang.org/
- http://learnyousomeerlang.com/

The End

Any questions?

Thanks for your attention.