Lab One – Erlang's History

Luca Gristina
luca.gristina1@marist.edu

September 13, 2023

1 Lab 1 Questions

1. What is single assignment?

   It means that a variable can only be bound once.

2. What's the difference between a bound and unbound variable?

   A bound variable has some sort of value attached to it while an unbound variable doesn't have a value.

3. How does variable scope work in the Erlang environment?

   The scope of a variable is its function clause. Variables bound in a branch of an if, case, or receive expression must be bound in all branches to have a value outside the expression. Otherwise they are regarded as 'unsafe' outside the expression.

4. Does Erlang implement mutable or immutable memory state? Why?

   Variables in Erlang are immutable. Once bound it can not be mutated nor become a different value. The immutable variables allow for increased efficiency within the language.

5. Describe Erlang's memory management system?

   Each Erlang process has its own stack and heap which are allocated in the same memory block and grow toward each other. When the stack and the heap meet the garbage collector is triggered and memory is reclaimed.

6. What does "Erlang" mean or stand for, if anything?

   It stands for Earl's Language.

7. Contrast "soft real time" from "hard real time".

   "Soft" real time means that some requests can miss the deadline, while "Hard" real time means that all requests must be satisfied within a specified time.

8. Why is Erlang so well suited for concurrency-oriented programming?

   It can handle large number of requests and processes at once. and it was made for building building systems that need to scale to handle large amounts of traffic.

9. Explain Erlang's "let it crash" philosophy?

   It's an approach to error handling that seeks to preserve the integrity and reliability of a system by intentionally allowing certain faults to go unhandled.

10. What's the difference between a tuple and a list?

    Tuples can be made up of multiple types, while a list has to be made up of one type.

11. What's BEAM?

    BEAM is a register machine, where all instructions operate on named registers. Each register can contain any Erlang term such as an integer or a tuple.

12. How can it be that we can create more Erlang "processes" than are allowed for in the operating system?

    Erlang is designed for massive concurrency. Erlang processes are lightweight (grow and shrink dynamically) with small memory footprint, fast to create and terminate, and the scheduling overhead is low.