Lab One: Erlang Questions

Anastasia LaPeruta
Anastasia.Laperuta1@marist.edu

September 14, 2023

1 What is single assignment?

Single assignment means that within an environment, variables can only be bound to a single value that cannot change.

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

A bound variable is one that has a value attached to it, while an unbound variable is one that no value attached to it.

3 How does variable scope work in the Erlang environment?

In the Erlang environment a variable's scope is local rather than global, meaning that it cannot be accessed outside of the function it is in.

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

Erlang implements an immutable memory state because within its environment once a value is bound to a variable it remains bound to it.

5 Describe Erlang's memory management system.

Erlang uses a real-time garbage collector. Memory is allocated automatically when needed and taken away when it no longer has a use (Armstrong et al., n.d.).

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

Erlang is named after Danish mathematician Agner Krarup Erlang. A man who had ties to the telephony industry that the language originated from (Armstrong, n.d.).

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

"Soft real-time" is best effort within a certain time frame, while "hard real time" is guaranteed.

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

Erlang is so well suited for concurrency-oriented programming because we do not have to worry about variables changing values.

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

Erlang handles errors remotely, not locally (Armstrong and Markus 22:35). Erlang does not catch any exceptions because it is a waste of resources.

10 What’s the difference between a tuple and a list?

A tuple is used to store ”a fixed number of items into a single entity,” while a list is used to store ”an arbitrary number of things” (Armstrong, 2013),

11 What’s BEAM?

BEAM is the Erlang virtual machine that compiles code in the Erlang Runtime System (Högberg, 2020).

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

All processes contain multiple threads within each individual process. In general, many threads per process allows Erlang more processes than cores.

Works Cited

Armstrong, Joe. "A History of Erlang." 2007, *https://www.labouseur.com/courses/erlang/history-of-erlang-armstrong.pdf.*

Armstrong, Joe. *Programming Erlang, Second Edition: Software for a Concurrent World.* Pragmatic Programmers, LLC., 2013.

Armstrong, Joe. et al. *Concurrent Programming in ERLANG: Second Edition.*" New Jersey: Prentice Hall, n.d.

Högberg, John. "A brief introduction to BEAM." *ERLANG*, Oct, 20, 2020. *https://www.erlang.org/blog/a-brief-beam-primer/.*

Markus. "Episode 89: Joe Armstrong on Erlang." *Software Engineering Radio*, Armstrong, Joe, Bernd, Mar. 12, 2008. *http://www.se-radio.net/2008/03/episode-89-joe-armstrong-on-erlang/.*