Assignment 1 – u20734621

Matthew Gotte

System Information

Text

Description automatically generated

All tests were executed and timed by tester.sh on the Windows Subsystem for Linux.

Script/Program to execute the experiment:

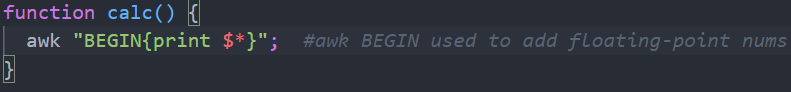
Explanation of approach:

The script begins with calling the functions to begin the tests on each language respectively. The test comprises of generating a λ, (with λ = single execution x 500). Each lambda was recorded and added to a running total to maintain the total execution time of the language, as well as compared to a variable to maintain which λ (500 executions) executed the fastest. The script repeats this for each language and then generates the following:

1. Executable file size in KB.
2. Total execution time for all λ in seconds. This total is purely the execution time of the executable file and does not factor any overhead code such as comparison in each iteration to maintain the lowest λ, this makes it very accurate in terms of the total execution time of the language.
3. Average λ for that language in seconds. This value is given as (Average λ) = (Total λ) / 50, due to 50 λ being calculated
4. The fastest λ that executed in seconds. This is managed by assuming first λ being fastest then a comparison with each λ generated after and maintaining a value of the lowest λ that executed

The Script (refer to comments in script for explanations):

A picture containing text, monitor, screen, screenshot

Description automatically generated

Text

Description automatically generated

Experiment Results:

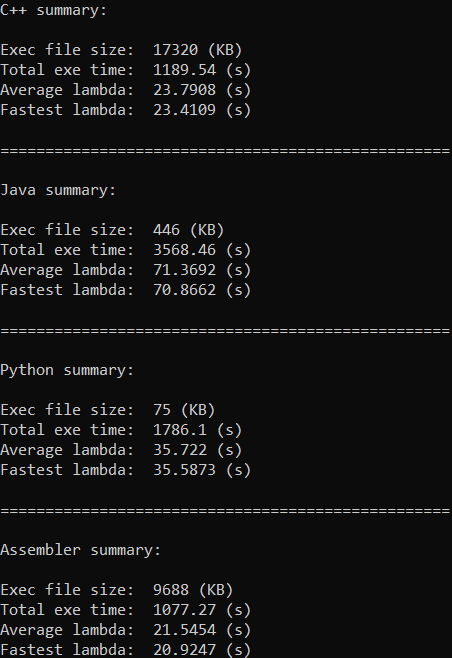
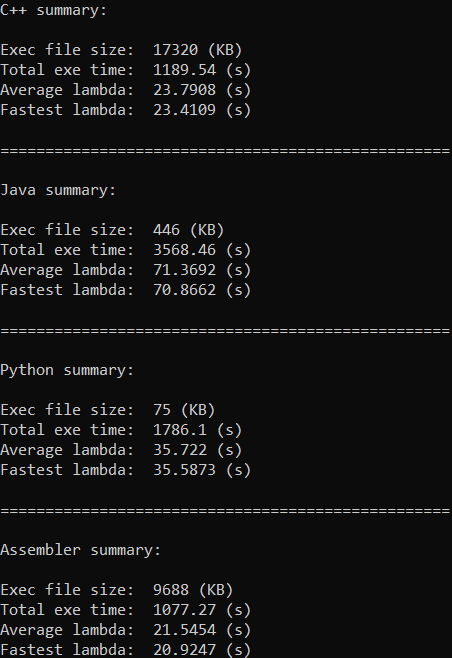


Table of results:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | C++ | Java | Python | Assembly |
| Total time [50 λ] (s) | 1189.54 | 3568.46 | 1786.1 | 1077.27 |
| Average λ | 23.7908 | 71.3692 | 35.722 | 21.5454 |
| Minimum λ | 23.4109 | 70.8662 | 35.5873 | 20.9247 |
| Average per 25000 executions | 0.0475816 | 0.1419384 | 0.071444 | 0.0430908 |
| File size (KB) | 17320 | 446 | 75 | 9688 |

Note: λ = 500 executions of a language Average per execution = Total time ÷ (50 x 500)

Graph of results:

Conclusion and Observations:

The slower languages (comparatively speaking) were the languages that compile into bytecode and are then interpreted. This presents in the results with Java and Python being the two slowest languages of the four.

The faster languages are the ones that are compiled languages, languages in this category are compiled into machine code by the compiler. This makes them faster than those converted to bytecode then interpreted. This presents in the results of C++ being faster than Java and Python.

The fastest language (according to the results produced in this experiment) is Assembly, this is due to assembly being purely assembled opposed to being compiled. Assembly is the lowest level language out of the four languages that were tested and thus it has shown that a purely assembled language is faster. With the results of this experiment, it can be said that in terms of increasing execution time, the order is, Assembly, C++, Python then Java.