Department of Computer Science

COS284 Practical and Assignment 1: Space and speed

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

1 Introduction

This document contains both practical 1 and assignment 1. In general the assignments will build upon the work of the current practical - due to this the practical will be due a week before the assignment to ensure you have enough time with the necessary skills to do the assignment.

1.1 Submission

The Practical portion is **due at 22:00 on Friday 27 August**.

The Assignment is **due at 22:00 on Friday 03 September**. You may use the practical sessions to ask for assistance if it is required, by visiting the module's discord server whose link can be found in the study guide.

1.2 Plagiarism policy

It is in your own interest that you, at all times, act responsibly and ethically. As with any work done for the purpose of your university degree, remember that the University of Pretoria will not tolerate plagiarism. Do not copy a friend's work or allow a friend to copy yours. Doing so constitutes plagiarism, and apart from not gaining the experience intended, you may face disciplinary action as a result.

For more on the University of Pretoria's plagiarism policy, you may visit the following webpage: `http://www.library.up.ac.za/plagiarism/index.htm`

1.3 Practical 1 [10%]

You must first complete this practical before attempting the assignment. For this task you have to implement the following 64-bit hello world assembly program in a file called **hello.asm**:

```
   segment .data
hello:  db "hello planet!",0x0a
   segment .text
   global _start
_start:
   mov eax,1
   mov edi,1
   mov edx, 13 ; The number of characters
   lea rsi,[hello]
   syscall
   mov eax,60
   xor edi, edi
   syscall
```

You are not at all expected to fully understand the code at this point in time. You are simply required to alter it to display the following:

```
XXXXXXXX is my student number.
```

Where the X's must be replaced with your student number.

When you have finished, upload your **hello.asm** file to the ff.cs.up.ac.za website, using the **Practical 1** upload link.

1.4 Assignment 1 [90%]

There exist many claims about the speed and space utilisation of assembler. This assignment will focus on trying in validate or invalidate these claims in a very simplified context.

You are required to implement a "hello world" esque program where the string **The quick brown fox jumps over the lazy dog.** is output using a number of programming languages. For each language you must record the **execution time** and the **size** of the "executable" file.

The languages and compilers to be used are:

- Java using javac.
- Python
- c++ using gcc
- Assembler using yasm

Tutorials on how to install each of these compilers can be found below:

- JavaC: `https://www3.cs.stonybrook.edu/~amione/CSE114_Course/materials/resources/installJava.html`
- Python: `https://realpython.com/installing-python/`
- GCC: `https://linuxhint.com/install_gcc_ubuntu/`
- Yasm: the instruction for the Yasm install will be available in the course slides.

While each of these compilers can theoretically be installed on windows it is HIGHLY recommended that you install them on a Linux machine or virtual machine, due to possible complications that may arise in later projects that will occur in Windows installations.

For c++ you must test using `cout` (not `printf`). For all languages utilise the default optimisation flag (i.e. don't set a optimisation flag).

Given that the program that you plan on executing is so small many external factors could substantially alter the completion time. For example your program being prioritised lower than some system maintenance task. To mitigate this, calculate the time required for the program to execute 500 times, call this value λ. Now the time one execution would take is on average $\lambda/500$. It is still possible that one run out of the 500 took drastically longer to execute due to external influences, and as a result the average time might be biased heavily. It is advised that you calculate 50 λs and report the minimum and average of these 50 λs. It is also highly advisable to use a high precision timer.

Effectively you will need to have run your program in each language **25000 times**, for a total of **100 000** iterations across all languages. This will take a long time to run, please account for this and allow enough time before the deadline to perform this operation such that you will have adequate time to write up the report.

You must write a report about these experiments. This report must include the following information:

- Introduction with system information such as processor and operating system
- Script/program you used to run your experiments must be explained and justified
- Experiment results - preferably in table/graphical form
- Conclusion about your findings

The report should not exceed 3 pages because you are not being marked on how much content you produce, rather the quality thereof.

One you have completed this assignment you must create an archive containing everything that would be required to run your program and obtain your empirical results as well as your report. This archive must be uploaded to the slot **Assignment 1** on ClickUp. This assignment will be hand marked by the course assistants.

2 Submission Details

2.1 Practical 1

Below is the required structure and naming scheme for the submission.

```
task1.tgz
-- hello.asm
-- makefile
```

2.2 Assignment 1

Below is an example submission, naming and structure do not have to match exactly but should make sense.

```
uXXXXXXXX_A1.zip
-- report.pdf
-- programs
  -- Python
    -- hello.py
  -- Java
    -- hello.java
  -- C++
    -- hello.cpp
  -- Assembly
    -- hello.asm
  -- tester.sh
```

3 Mark Distribution

Activity	**Mark**
Practical 1	10
Assignment 1	90
Total	**100**