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.

1.1 Submission

The Practical portion is due to be uploaded on 14 August.

The Assignment will be marked in later practical sessions and is due at 17:00 on Friday 21 August. You may use the practical sessions to ask for assistance if it is required, by visiting the module 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 responsible 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 Introductory component [28%]

You must first complete all 3 tasks of this practical. Note as the questions will state, the first two tasks must be completed on ClickUp.

1.3.1 Task 1: Exit program [10 %]

See the ClickUp assignment marked Practical 1 Task 1.

1.3.2 Task 2: Data example[10 %]

See the ClickUp assignment marked Practical 1 Task 2.

1.3.3 Task 3: Hello World [8%]

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 assignments.cs.up.ac.za website, using the **Practical 1 Task 3** upload link.

1.4 Substantive component [72%]

There exist many claims about the speed and space utilization of assembler. This assignment will focus on trying in validate or invalidate these claim 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://www.ntu.edu.sg/home/ehchua/programming/howto/JDK_HowTo.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 utilize the default optimization flag (i.e. don't set a optimization 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 prioritized 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.

You must write a 2 page report. This report must include information about your system such as processor and processor core number and operating system. It must also include the results of the tests described above and any conclusions that you have drawn from those results.

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**. This assignment will be hand marked by the course assistants.

2 Mark Distribution

Activity	**Mark**
Task 1	10
Task 2	10
Task 3	8
Substantive component	72
Total	**100**