UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science

COS 122 Operating Systems

Practical 1

Due: 2017-08-14 @ 14:00 PM

July 29, 2017

PLAGIARISM POLICY

UNIVERSITY OF PRETORIA

The Department of Computer Science considers plagiarism as a serious offence. Disciplinary action will be taken against students who commit plagiarism. Plagiarism includes copying someone else's work without consent, copying a friend's work (even with consent) and copying material (such as text or program code) from the Internet. Copying will not be tolerated in this course. For a formal definition of plagiarism, the student is referred to `http://www.ais.up.ac.za/plagiarism/index.htm` (from the main page of the University of Pretoria site, follow the *Library* quick link, and then click the *Plagiarism* link). If you have any form of question regarding this, please ask one of the lecturers, to avoid any misunderstanding. Also note that the OOP principle of code re-use does not mean that you should copy and adapt code to suit your solution.

Instructions

In this module it is advised that you have a Virtual Machine(VM) to do all your practicals due to the nature of this course it is easy to mistakenly destroy your Operating System. For all practicals you must make use of a Linux VM.

Follow the Upload and Demo instructions carefully at the end of the practical.

- All written answers must be in either txt or pdf format no other formats will be marked.
- Only upload your source code, makefile, and supporting documents. Do not upload large, binary, or OS files. Keep your uploaded archive small.
- Upload your work in a tar or gz file before 14:00 on the 14th of August 2017.
- Bring your VM on a flash disk to your booked practical demonstration session in the week starting on the 15th of August, so you may be marked.
- On non-demo days, there will still be teaching assistants available in the labs to help you.
- Make use of a Makefile for your program.
- You makefile **must** have these commands:
  - # make

Task 1 - Create your own VM [2 Marks]

For this task you have to be able to make sure you have created a Linux Virtual Machine with all the necessary configurations and portability, it is up to you to decide which distribution and version of Linux you want. Here are a few examples that you can use:

- Ubuntu https://www.ubuntu.com/download/desktop
- Gentoo https://www.gentoo.org/downloads
- Arch https://www.archlinux.org/download
- Manjaro - For those who want a customized Arch-Linux but don't know how to build it, this distribution is already pre-loaded with the essentials. https://manjaro.org/get-manjaro

You can use any Virtual Machine emulator of your choice as long as it can run on the lab computers. It is advised you use Virtual Box (https://www.virtualbox.org/wiki/Downloads) for this since it is free and the lab computers already have them installed. You can also use VMWare Workstation Player (https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_workstation_player/12_0) which has more features and better integration.

You can obtain some of the latest distributions mentioned above from the ftp server available at the address ftp://ftp.cs.up.ac.za/Courses/COS%20122/. The ftp server is only available on the campus computers and wireless networks, you will not have access to the server off campus.

You have received a document in COS132 that can help you set-up a VM. Remember this module is about Operating Systems so you will gain experience from making a VM and learn how to configure one. You will also learn the trade-offs of having a VM vs Physical machine. You are expected to have your VM portable so that you are able to bring it on a flash disk for demo's. You will be using it for the remainder of this course, so you should spend some time getting comfortable with the OS and be able to relate certain aspects with the theory you are learning. In order to get the marks for this task you will need answer any question(s) the teaching assistants ask you to show that you understand the basics.

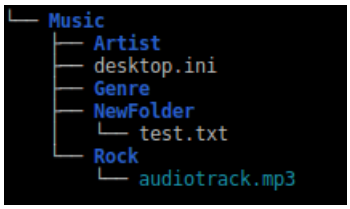
Note: Since you creating your own VM you need to make special care that your settings for the VM doesn't exceed the hardware capabilities of the host machine. Inspect the hardware of the lab computers to determine the settings of the VM.
If you make use of the VM on the ftp server you will not get full marks.

Task 2 - Understanding basic commands [5 Marks]

Task 2.1

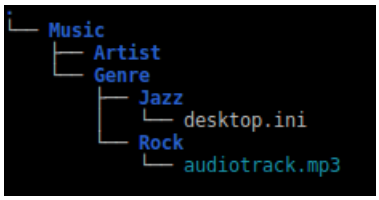
In this task you are required to understand the following commands (**ls, mkdir, touch rm, rmdir, mv, cd**). You can also refer to the [man] pages of the commands for quick reference and examples.

Make the file structure given below and write down the commands that you used to achieve this structure. Assume that there is a directory with no files and no directories in it. The files you create should be empty.



Task 2.2

Now that you have completed Task 2.1 finish off this task by performing the necessary commands to get the structure below. Also write down the commands you used.



Task 3 - Superuser power [3 Marks]

An Operating System can have several users accounts and at least one of them should have elevated rights to modify system files, etc. One command that grants you privileges is sudo or su meaning "superuser do". Learn more about the capabilities of sudo and why it can be a dangerous command.

For this task you are required to delete all user account files, give the command you used and restart your VM.
You will then notice something unexpected, ask yourself why a VM is better than an installed OS and why having backups is important.
If not done already follow Task 1 again since the VM is no longer usable.

Task 4 - Interprocess Communication [10 Marks]

For this task you will be required to read up on pipes and interprocess communication, please consult the following link for some basic information about pipes (http://www.linfo.org/pipes.html). You will then have to write your own C++ program. Your program will simply convert all input from any case to upper-casing each word. However, you should be able to pipe input into your program, as well as pipe output out.

If your program is called "UpCase", you should be able to execute it as in the following example:

- # echo "HelLo wOrld" | ./UpCase > output.txt

Implementation

The following steps show how your program should work:

1. Your program receives input.
2. It creates a child process with the fork() function.
3. The parent sends the input to the child process via a pipe.
4. The child converts the input to upper-case and sends it back to the parent via a pipe.
5. The parent outputs the received data.

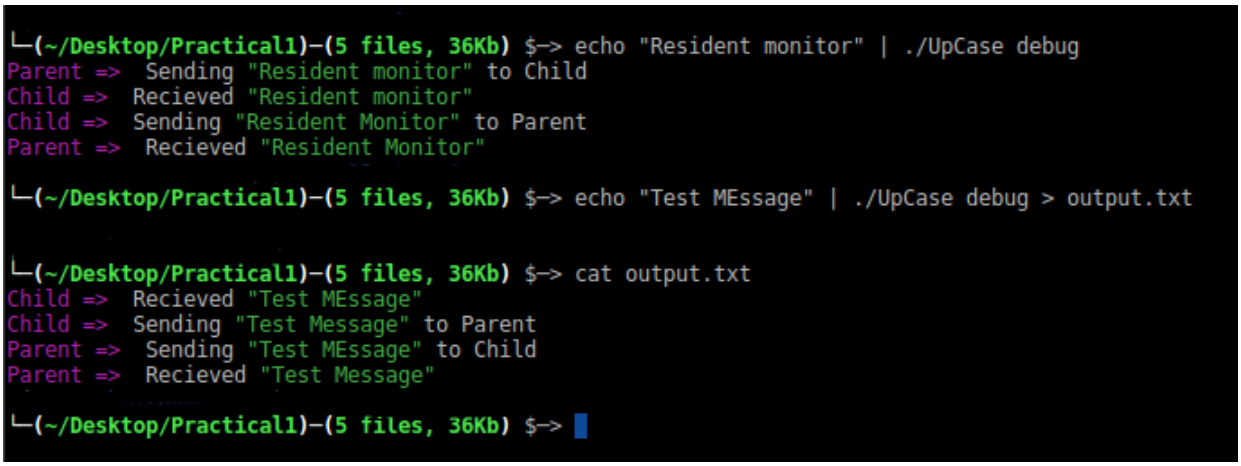
Example Output

The screen-shot below shows what the output should look like. Familiarize yourself with the package manager you have on your VM and install the necessary packages as needed. The example below makes use of a package called "fortune-mod" for the command [fortune].

```
└─(~/Desktop/Practical1)─(5 files, 36Kb) $-> gcc main.cpp -o UpCase

└─(~/Desktop/Practical1)─(5 files, 36Kb) $-> echo "cOs122" | ./UpCase
Cos122

└─(~/Desktop/Practical1)─(5 files, 36Kb) $-> echo "mY pipe program is cOOl" | ./UpCase > output.txt

└─(~/Desktop/Practical1)─(5 files, 36Kb) $-> cat output.txt
My Pipe Program Is Cool

└─(~/Desktop/Practical1)─(5 files, 36Kb) $-> fortune | ./UpCase
You Definitely Intend To Start Living Sometime Soon.

└─(~/Desktop/Practical1)─(5 files, 36Kb) $->
```

Full Marks

To obtain full marks for this question, you should make it easy to debug your application. You can implement this by simply sending an argument to your application and checking its value. When debugging is enabled, your output should look like this:



Important!

- You have to communicate by using pipes and the read() and write() functions.
- You are not allowed to execute other commands from within your application by using functions like **execve()** and friends.
- Your output **does not** have to be in colour to get full marks.
- Make a makefile for yourself. It will speed up your work-flow.

Upload Instructions

As it would be impractical to upload your Virtual Machine to the CS website for the practicals, you are required to submit your code and answers to the written questions:

- Upload your work in a tar or gz file to the Practical 1 assignment slot on the COS 122 course website before 14:00 on Monday 14 August 2017. No late submissions will be accepted.
- You have to demo what you uploaded.
- All written answers must be in either txt or pdf format no other formats will be marked.
- Every file you submit should contain your name, surname, and student number at the top of the file in comments.
- Only upload your source code, makefile, and supporting documents. Do not upload large, binary, or OS files. Keep your uploaded archive small.
- **Failure to upload your work will result in 0 marks being awarded for your practical.**

Demo Instructions

At the demo, we will check your uploaded work. Only students who have uploaded on time will be allowed to demo. **You may only demo during the slot where you are booked**. You should be able to demo the practical within Virtual Box.

- This practical must be marked by a tutor or teaching assistant during your booked practical demonstration sessions.
- You have to be there at the start of the session until you get marked. If you come just before the session ends you will not get marked.
- You have a limited time to demo approximately 4 minutes. Please ensure that your VM is running before you get marked to speed up the process.
- If you are getting marked next and you are not ready to demo you will receive 0 marks for this practical.
- If your program doesn't compile due to syntax errors you will get 0 marks.
- If your program receives a runtime error you will lose marks.
- You have to be prepared to answer any question(s) the teaching assistant or tutor may ask you. You should not have to waste time looking for the answer.
- Make sure that you sign the attendance register for the session and ensure that the teaching assistant has entered your mark on the mark sheet correctly.

- **Failure to demo your work will result in 0 marks being awarded for this practical.**

Total: [20]