

CS342 - Spring 2019

Homework #1

Assigned: **Feb 08, 2019.**

Due date: **Feb 15, 2019, 23:55.**

Submit through Moodle. Make sure you start submitting one day before the deadline. You can overwrite your submission as many times as you wish. Late submissions will not be accepted (no excuse; no email will be accepted).

Please do the following tasks and write a report at the end.

1. Install the following Linux distribution and release on your own computer: Ubuntu Desktop 64-bit 18.04 LTS. It is essential that you install this distribution and release so that you will not have problems like "was working on my machine". You can install Linux on bare hardware, i.e., on a partition of your hard-disk. In this case, make sure you first backup all your important data so that you do not lose your data in case your computer does not boot up after installation.

You can also install Linux in a virtual machine created in your computer. For this, you first need to install a virtualization software, VMware or VirtualBox (or some other virtualization software), on your computer. VirtualBox is free to use. VMware Player is also free to use. If you have, you can also use VMware Fusion (for OS X) or VMware Workstation.

You can help each other in installing Linux.

Write briefly about your installation choices and experiences in your report. After installing Linux, start Linux and learn basic Linux usage. There are lots of guides and tutorials in Internet teaching basic Linux usage. You can benefit from them. In your report, write down the names of 10 Linux commands that you learned.

2. Find out and write down the location (pathname) where the kernel executable resides in the default directory tree (starting with /) of your Linux installation. Find out the version of your running kernel by using the "uname -r" command. Write the version number in your report.



```
/lib/modules  
version 4.15.0-34-generic
```

3. Download the source code of the Linux kernel (from kernel.org, for example). Download the version that is close to the version of your running kernel. After opening the tar package, change into the root directory of the downloaded kernel source code (it is in the directory where you downloaded the tar package), and in your report write the names of the subdirectories you see there.
4. In the source code of the kernel, find out the definition of the system call table. Write the pathname where you found it. Then, examine the table. Find out

```
/tmp/mozilla_parallels0/linux-4.14.98/arch/x86/entry/syscalls
5 common fstat sys_newfstat
43 common accept sys_accept
123 common setfsgid sys_setfsgid
220 common semtimedop sys_semtimedop
```

the system call names corresponding to system call numbers 5, 43, 123, and 220.

5. Use the **strace** command of Linux to trace the systems calls made by some simple programs like **cp**, **ls**, etc. Use the manual page of strace to learn more about it (type **man strace**). Include sample output in your report. The **man** command provides help pages about Linux commands, system calls, and C library functions.
6. Use the **time** command to measure the time required to execute some programs like **cp**, etc. It reports different times: real, user and sys. What are they? Write those values for different program executions.
7. Learn C Programming [1,2]. Write a simple C program to compare the cost (in terms of elapsed time) of different system calls with different parameter values. Different system calls can be: **getpid()**, **read()**, **write()**, **open()**, **mkdir()**, etc. Different parameter values, for example for the **read()** system call, can be: read 100 bytes, 1000 bytes, 10000 bytes, or 100000 bytes into a buffer of enough size. You can call the program as **cost.c**. To get the current time in the program you can use the **gettimeofday()** system call. It gives the current time in microseconds granularity. You can call this before and after a system call is made to measure the elapsed time.

Write a simple Makefile to compile your program. A Makefile is a set of directives and commands specified in a file to compile a project. The following can be a starting point for your Makefile content. Be careful about TAB characters.

```
all: cost
cost: cost.c
    gcc Wall g -o cost cost.c
clean:
    rm -fr cost cost.o *~
```

This is useful for you to warm up with C. Make sure you do it alone. Otherwise it will be very difficult to do the projects. You will develop your code in C and Linux. You will use the gcc compiler. Include the source code of your program in your report. Do a lot of timing experiments and report the results as nice as possible.

8. Read the first three chapters of the textbook.

Submission

Submit a pdf file as your report which will include the information required in each question above.

Submit also your `cost.c` file and your `Makefile`. Put these files into a directory named with your Student Id, and tar and gzip the directory. For example a student with ID 21404312 will create a directory named 21404312 and will put the files there. Then he will tar the directory (package the directory) as follows:

```
tar cvf 21404312.tar 21404312
```

Then he will gzip the tar file as follows:

```
gzip 21404312.tar
```

In this way he will obtain a file called `21404312.tar.gz`. Then he will upload this file in Moodle. Late submission will not be accepted (no exception). A late submission will get 0 automatically (you will not be able to argue it). Make sure you make a submission one day before the deadline. You can then overwrite it.

References

- [1] The C Programming Language. B. Kernighan and D. Ritchie. Second Edition. Prentice Hall. 1998. A must have book; very useful.
- [2] Any Book on C, available in Meteksan Bookstore.

Tips and Clarification

- Make sure you learn a debugger like `gdb`, `xxgdb`, or the debugger of the IDE (integrated development environment) that you are using to develop programs (for example Eclipse IDE).
- There are lot of documentation in Internet about how to develop, compile, run, and debug C programs in Linux OS. You can benefit from them.