CS342 Operating Systems – Fall 2017
Homework 1

Assigned: Sep 26, 2017
Due date: Oct 4, 2017, 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).*

1) Install the following Linux distribution and release on your own computer: **Ubuntu Desktop 64-bit 16.04.X LTS**. X can be 3 or something else like 1. 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 will have a backup copy in case your computer does not boot up after installation.

You can also install Linux in a virtual machine created in your computer. In this case, you first need to install a virtualization software (called virtual machine monitor), 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 your installation choices and experience 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 our report, write down the names of 10 Linux commands that you learned.

2) Find out and write down where the Linux *kernel executable* is in your Linux directory tree. Find out the version of your running kernel by using the "uname -r" command. Write the version number in your report.

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. You can learn the version of your running kernel by typing "uname –r". Change into the root directory of the source code tree and in your report write the names of the subdirectories you see there.

4) 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.

5) Learn C Programming [1][2]. Write a simple C program to compare the cost of a simple procedure call to a simple system call (getpid() is a good candidate as a system call to test – it is returning the integer id of the process – running program - calling it). Call your program as cost.c. While compiling your program, turn off the compiler optimization so that procedure calls are not optimized and made inline. To measure the cost in terms of time, you can use the gettimeofday() system call of Linux. It can return the real time in microseconds granularity. Explain the time difference between

making a system call to a function in kernel and making a procedure call to a function in your program.

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
	cc –Wall –g  -o cost cost.c
```

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. Make sure you develop your code in C and Linux. You can use the gcc compiler.

Include the source code of your program. Do a lot of timing experiments and report the results as nice as possible. Measure the mean and standard deviation of time spent to call a procedure and the time spent to call a system routine (you may include the procedure and system call execution time – therefore make sure you use simple procedures or system calls).

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. Hence you will submit 3 files: report.pdf, cost.c and 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 his 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 Clarifications:

--- 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.