CS 377 : Operating Systems Laboratory
Assignment #5, 13 February 2015

1. Answer the following questions:

Q1. How to execute a user program: Where should we put the source code file and how should we execute that program? Which are all the files which can be included in the user space.

Q2: To understand the code of a system call, we have to study (i) Header file of that system call, (ii) Source file for that systemcall , (iii) include/libc/syscall.h, and (iv) src/libc/syscall.c. For example there is a systemcall Get_Key(), which is very similar to scanf() in C. For that system call we have to study include/libc/conio.h, src/libc/conio.c, include/libc/syscall.h and, src/libc/syscall.c.

(a) Describe what is the purpose of each of these 4 files.
(b) Explain what changes have to be made in these four files to add a new system call.

Q3. While adding a system call, how do we tell the OS about number of parameters passed in the call? Follow the hint in Q2 and find which variables contain relevant information.

Q.4 Find where the pid of a process is stored**.**
(*Hint:* In systcall.c we can see a variable CURRENT_THREAD. CURRENT_THREAD is a pointer to Kernel_Thread structure declared in kthread.h. Look into Kernel_Thread structure.)

2. Use kernel facilities to accept input from the keyboard

Write a user level program which will accept a string from the keyboard and print it on the console. The string should be terminated by a "@" symbol.
(*Hint:* Try to find answer for Q.1 and also look at include/libc/syscall.h and src/libc/syscall.c for various system calls that are supported by GeekOS.)

3. Add a new system call "Get_NewTOD"

Add new system call to provide the Time of day.

```
void Get_NewTOD(int *);
```

For example if we call "Get_NewTOD(&xyz)" from the user program then "xyz" should contain the value of time of day. We want this system call to be added to sched.h & sched.c. Note that the value should be deposited in "xyz" before control exits the kernel.

(*Hint:*
1. There is already a system call GetTimeOfDay() in those files. See its implementation for help. Also *ve*rify that our system call is working properly.

2. There are two functions in user.h

```
bool Copy_From_User(void *destInKernel, ulong_t srcInUser, ulong_t bufSize);
bool Copy_To_User(ulong_t destInUser, void *srcInKernel, ulong_t bufSize);
```

These functions are used to access the address space of the process, i.e., values in the memory

space allocated to a process from within the kernel. Try to figure out how these functions will be helpful for us.)

4. Add some system calls to collect information about execution of a process

Add new system calls to provide the following functionalities:

(a) Count the number of system calls made by each process,
(b) Count the number of files opened by each process,
(c) Count the number of read system calls made by each process,
(d) Display this information on the console.

(*Hint:* Think of where you should maintain this information and how/when you should update it.)