CS 377 : Operating Systems Laboratory
Assignment #6, 27 February 2015

Question 1:

Boot Process is started from GeekOS/src/geekos/main.c. There are many function calls are there in main.c . Give a brief decription for each of the function call and where is that function defined. (Make yourself familar with those functions as some basic questions will be asked about those functions .)

(Hint: See All the header files inherited by main.c and search for function definition. Or Try to use cscope for this purpose.)

Question 2:

Add a kernel thread on the bootup of the GeekOS, just after the init() process. Kernel state will be maintained for that thread. Traverse the whole kernel_thread structure and tell what does each element in the kernel_thread structure mean ?

Add a function that asks for a password and compares it with some stored password to decide whether to allow access.

(Hint: Passwords should be stored in a file with some standard name, say Password.txt. Put this file in GeekOS/build/user/Password.txt and in Open function path will be “/c/Password.txt” .See fileio.h and string.h)

Question 3:

Many operating systems do not use terms “Kernel Thread and User Thread”strictly. So we have to study kernel threads of Geekos and decide whether it is a kernel-level thread or user-level thread. Give proper reasons for your answer.

(Refer Class notes to get the differenece between User Thread and Kernel Thread)

Question 4:

In Question 2, you are waiting for the keyboard input i.e waiting for an I/O.Say, how is a process put to sleep or activated when it is engaged in doing I/O?

(Describe from both I/O as well as shceduling point of view.)

Question 5:

In GeekOS source files they have given TODO_P(PROJECT_NAME,”Message”) for some of the functions instead of giving actual definition. For example in syscall.c

```
static int Sys_Fork(struct Interrupt_State *state)
{
   TODO_P(PROJECT_FORK, "Fork system call");
   return EUNSUPPORTED;
}
```

Give detailed implementataion level work flow for fork() system call.