OS Lab 05

Anurag Shirolkar (120050003) Dheerendra Rathor (120050033)

February 13, 2015

Contents

1 Question 1

1. (a) where to put the user program : in the directory src/user
   (b) execute following commands
   - make (in the build directory)
   - geek
   - <name of the program> (in the geek terminal)

   (c) the header files in the directory **include/libc** can be used in the user programs
2. (a) what is the purpose?

file	purpose
syscall.h	declarations of all the systemcalls
syscall.c	definition of the systemcalls declared in syscall.h
conio.h	declaration of all the console input/output functions
conio.c	definitions of the functions declared in conio.h

(b) to add new syscall add the syscall to the enum which has list of all the syscalls write the declaration in the syscall.h file write the definition in the syscall.c file

3. In the wrapper of the syscall function the last argument is of the form **SYSCALL_REGS_<i>**
   where i is the number of parameters in the system call

4. There is a structure **kernel_thread** defined in the file **include/geekos/kthread.h** that structure has a member **int pid**. The $\text{current}_{\text{thread}}$ variable in the syscall.c file is the pointer to the structure of current thread which contains the pid member.

5.

2 Question 2

In the geek terminal Type the following command :
$ q2
The input will be read till @ and printed to the console.

3 Question 3

In the geek terminal Type the following command :
$ q3
produces the following output

Output from old get time of day syscall : <time>
Output from new get time of day syscall : <time>

4 Question 4

Created a new member **sys_call_count** in the kthread.h which is initialized as 0 in kthread.c. Whenever the syscall handler in **trap.c** is called the **sys_call_count** of the **CURRENT_THREAD** variable is incremented. Defined a new syscall which returns the count of syscalls from CURRENT_THREAD.
Similar method used for file count.