Project 1—Adding a System Call to the Linux Kernel
(**Due date: 9/29/2009/Tuesday**)

Your name:	Date:

This project is based on the previous project Project#0. If you have not completed the Project#0, you need to (1) successfully complete the Project#0, or (2) work on your own Linux machine (***at your own risk*** such that this project will change your Linux Kernel and settings, and most likely, much more time is needed—because the following project steps are strictly based on the Project#0).

- If you have interests about Linux Kernel, read the text in the textbook (on pages 93-97) on how to add a system call to the Linux kernel (this is an *optional*).
- I am assuming that you have installed **Ubuntu8041** (its **kernel version: 2.6.24-19**) appliance and VMware player on your PC as required in Project#0.
- Work through the following steps and **take screenshots and save them for your submissions**.
    - Very important: **BE very CAREFULLY** when typing the commands and making changes to the (program) files throughout the project process!!!
        - If there is typo, sometimes, you have to wait for the completion of the Kernel installation to know that you made a mistake, which will take several hours.
        - **!!!**Pay particular attention to the **space** (“ ”), **dot** (“**.**”), **hyphen** (“**-**”), and **two consecutive hyphen** (“**--**”) in the commands **!!!**
- Follow the Submission instructions on how to submit your work for this project.

1. Start up the VMware player.

2. Boot Ubuntu8041 within the VMware player.
    a. The username and password is “user” (unless you use a different version of Linux).

3. Open a CLI/Termina:
    a. Type “pwd” command which will show the working directory of the “user” (at “/home/user”, if it is not, you need to use “cd” command to go there).
    b. Get the current Linux kernel version by typing:
        i. “date”
        ii. “uname –r”
        iii. (I am assuming that it is “2.6.24-19-generic”).
        iv. **Take a screenshot of this CLI (with the outputs of “date” and “uname –r”) for your project submission**.

4. Type command “sudo passwd root” to update/change *root* (super-user) password:
    a. First you need to provide user’s password (that is, “user”);
    b. Then type the new-password for root (REMEMBER this root password);
    c. Retype this new password again.

5. Log in as root by typing “su” command (you need to provide the root password for command prompt question). Now you are root user that can write anything to the system’s directories (BE CAREFULLY when using commands such as “rm”, “rename” …that will change the contents of the files…).

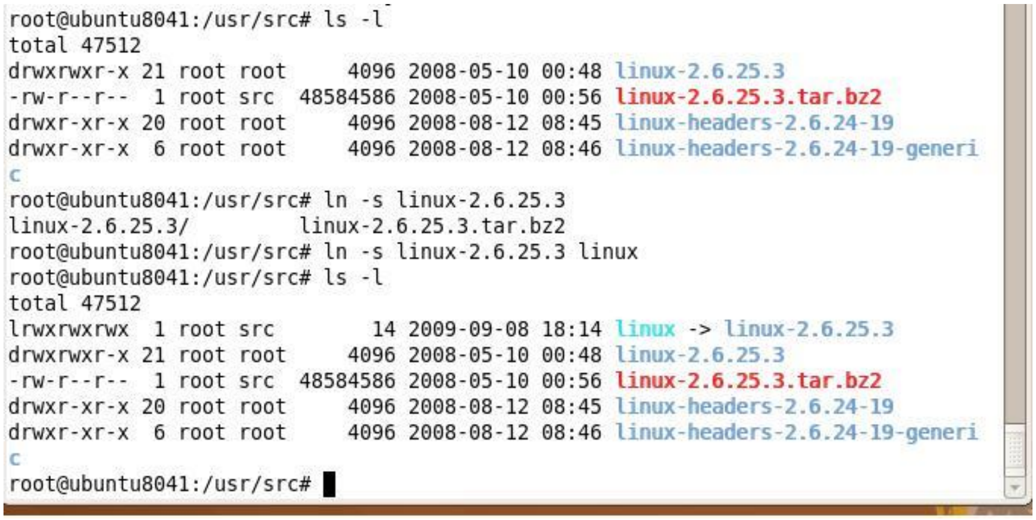
6. Update your Linux package database with the command
    a. “apt-get update”

7. Install all needed packages by typing the command:
    a. “apt-get install kernel-package libncurses5-dev fakeroot wget bzip2”
    b. When you are asked if you continue [Y/n]?
        i. Type “Y”—upper case!!
    c. Wait for sure till these packages have been updated successfully.

8. Enter directory “/usr/src” by typing command:
    a. “cd /usr/src”
    b. (make sure you are under this directory by using command “pwd”).

9. Download new kernel with the following command:
    a. “wget http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.25.3.tar.bz2”
    b. If the above command does not work, do the following:
        i. Use Firefox to download the above kernel
        ii. It will be (usually) saved on the “Desktop”
        iii. Copy that file to your current directory which is “/usr/src” (if not you need to enter there by “cd /usr/src”):
            1. “cp /home/user/Desktop/linux-2.6.25.3.tar.bz2 **•**” (do NOT forget the “**•**”)

10. Extract the file and prepare for compiling by using the following commands:
    a. “tar xjf linux-2.6.25.3.tar.bz2”
    b. “ls –l”
    c. “ln -s linux-2.6.25.3 linux”
    d. “ls –l”
    e. “cd linux”
    f. (**Take a screenshot that shows all the above results** such like the following one).



```
root@ubuntu8041:/usr/src# ls -l
total 47512
drwxrwxr-x 21 root root     4096 2008-05-10 00:48 linux-2.6.25.3
-rw-r--r--  1 root src  48584586 2008-05-10 00:56 linux-2.6.25.3.tar.bz2
drwxr-xr-x 20 root root     4096 2008-08-12 08:45 linux-headers-2.6.24-19
drwxr-xr-x  6 root root     4096 2008-08-12 08:46 linux-headers-2.6.24-19-generi
c
root@ubuntu8041:/usr/src# ln -s linux-2.6.25.3
linux-2.6.25.3/         linux-2.6.25.3.tar.bz2
root@ubuntu8041:/usr/src# ln -s linux-2.6.25.3 linux
root@ubuntu8041:/usr/src# ls -l
total 47512
lrwxrwxrwx  1 root src        14 2009-09-08 18:14 linux -> linux-2.6.25.3
drwxrwxr-x 21 root root     4096 2008-05-10 00:48 linux-2.6.25.3
-rw-r--r--  1 root src  48584586 2008-05-10 00:56 linux-2.6.25.3.tar.bz2
drwxr-xr-x 20 root root     4096 2008-08-12 08:45 linux-headers-2.6.24-19
drwxr-xr-x  6 root root     4096 2008-08-12 08:46 linux-headers-2.6.24-19-generi
c
root@ubuntu8041:/usr/src# █
```

11. Add a System Call to the Kernel:
    a. Go to directory “/usr/src/linux/kernel”
    b. Open and Edit the file “workqueue.c”:
        i. Type command “gedit workqueue.c” to open the file;

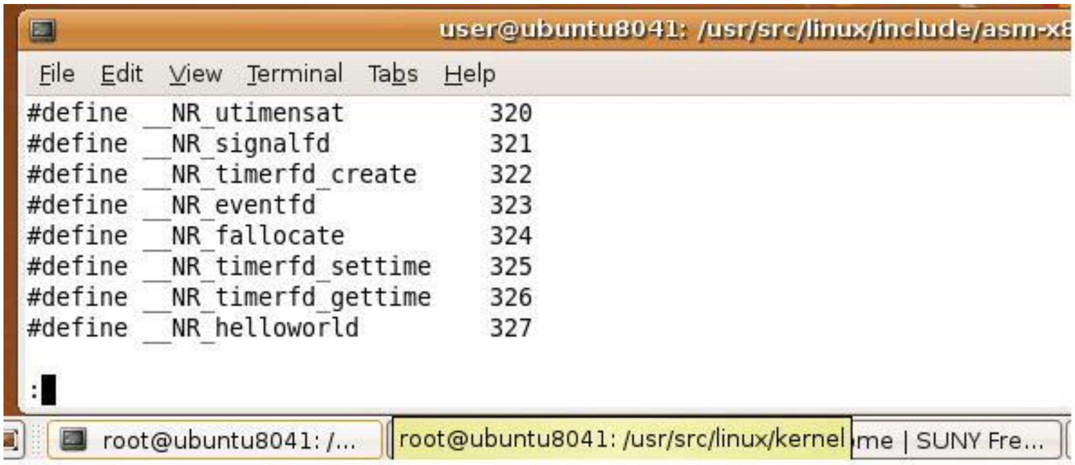
ii. Insert the following among the "#include…." at the beginning of that file:

```
#include <linux/linkage.h>
```

iii. Insert the following to the end of the file:

```
asmlinkage  int  sys_helloworld() {
        printk(KERN_EMERG  "hello world!");

        return  826;
}
```

12. Define a new system call number for __NR_helloworld in file "unistd_32.h" with the command:
    a. "gedit /usr/src/linux/include/asm-x86/unistd_32.h"
    b. Assign a **unique number** to __NR_helloworld, as shown in the following (here, the unique number for it is 327):



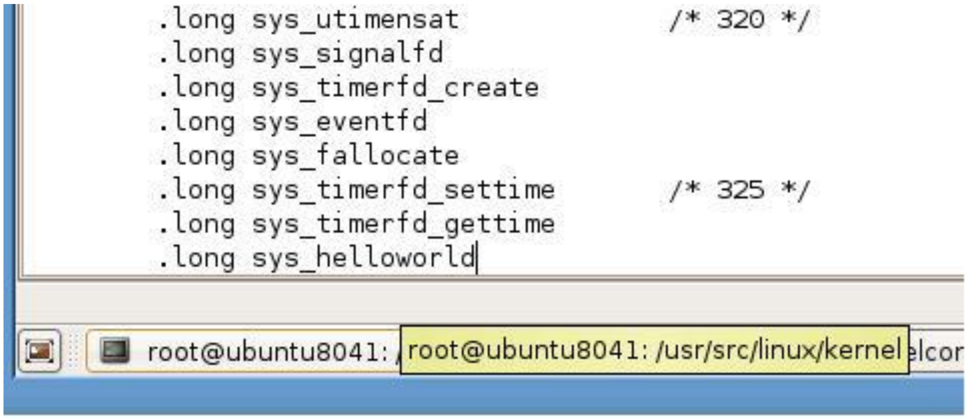
user@ubuntu8041: /usr/src/linux/include/asm-x8
File Edit View Terminal Tabs Help
root@ubuntu8041: /...
root@ubuntu8041: /usr/src/linux/kernel
me | SUNY Fre...

```
#define __NR_utimensat          320
#define __NR_signalfd           321
#define __NR_timerfd_create     322
#define __NR_eventfd            323
#define __NR_fallocate          324
#define __NR_timerfd_settime    325
#define __NR_timerfd_gettime    326
#define __NR_helloworld         327

:
```

13. Add an entry ".long sys_helloworld" to the file "syscall_table_32.S" with

    a. "gedit /usr/src/linux/arch/x86/kernel/syscall_table_32.S";

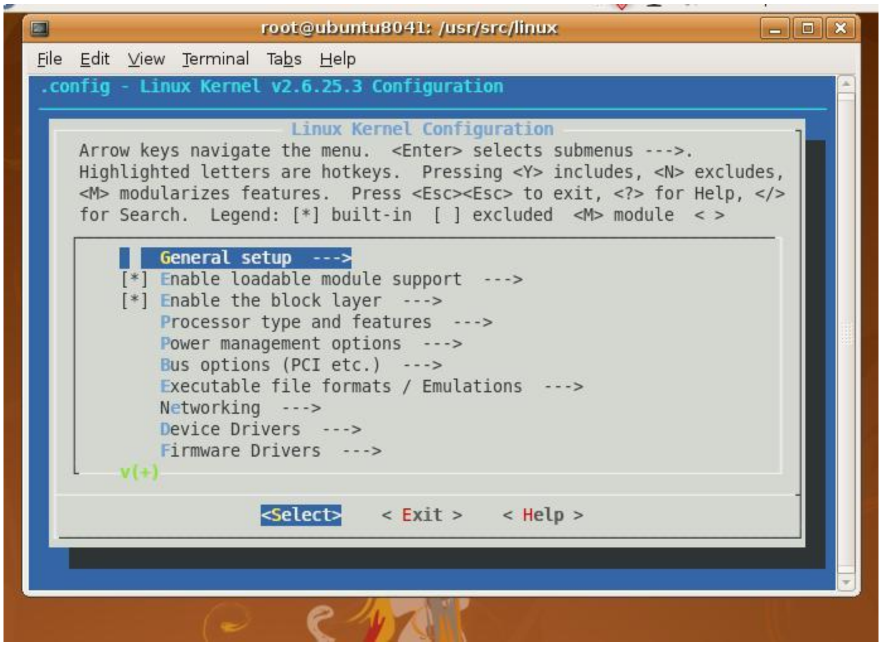
    b. Add the entry as shown in the following:



```
        .long sys_utimensat             /* 320 */
        .long sys_signalfd
        .long sys_timerfd_create
        .long sys_eventfd
        .long sys_fallocate
        .long sys_timerfd_settime       /* 325 */
        .long sys_timerfd_gettime
        .long sys_helloworld
```

14. Configure the kernel (now you are in the directory **"/usr/src/linux")** through the following commands:
    a. Now you should be in the directory **"/usr/src/linux"** ;
    b. Type command "cp /boot/config-`uname -r` **./.**config"
        i. Do NOT forget the two "**.**"
        ii. The "`" key is at the **upper-left corner** of the keyboard!!!

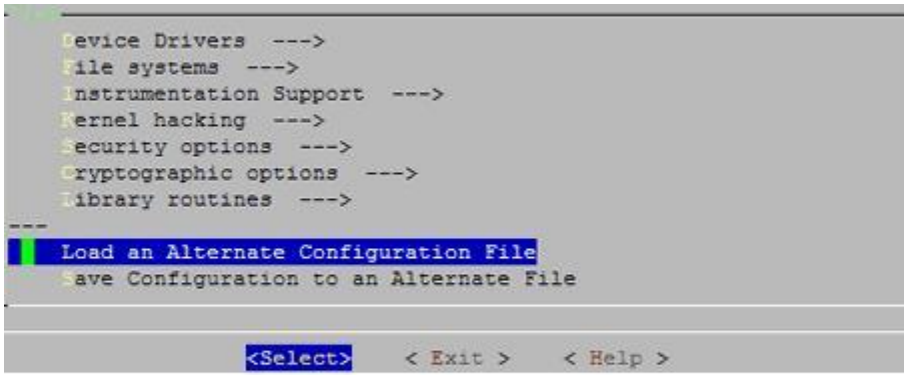
    c. Type command "make menuconfig", which will display the kernel configuration menu:



root@ubuntu8041: /usr/src/linux
File Edit View Terminal Tabs Help

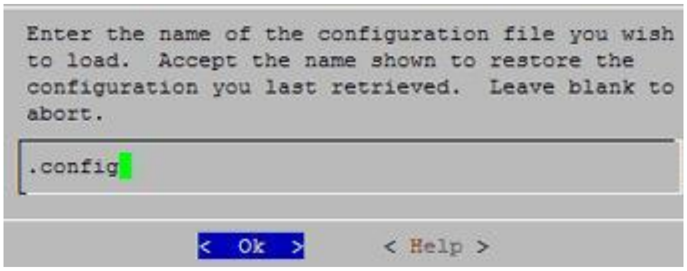
```
.config - Linux Kernel v2.6.25.3 Configuration

                    Linux Kernel Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus --->.
  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes,
  <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </>
  for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < >

        General setup  --->
    [*] Enable loadable module support  --->
    [*] Enable the block layer  --->
        Processor type and features  --->
        Power management options  --->
        Bus options (PCI etc.)  --->
        Executable file formats / Emulations  --->
        Networking  --->
        Device Drivers  --->
        Firmware Drivers  --->
    v(+)

              <Select>    < Exit >    < Help >
```

    d. Using the arrow keys to go to **Load an Alternate Configuration File** as shown in the following:



```
    Device Drivers  --->
    File systems  --->
    Instrumentation Support  --->
    Kernel hacking  --->
    Security options  --->
    Cryptographic options  --->
    Library routines  --->
---
    Load an Alternate Configuration File
    Save Configuration to an Alternate File

              <Select>    < Exit >    < Help >
```

e. Choose (or input) the "**.config**" file:

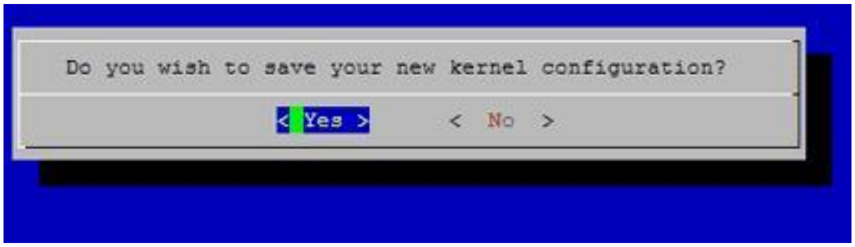


```
Enter the name of the configuration file you wish
to load.  Accept the name shown to restore the
configuration you last retrieved.  Leave blank to
abort.

.config
```

f. After selecting the above .config file, you may go through the kernel configuration (if you don't know about them, do NOT make any changes!).

g. Now you can exit by pressing <ESC><ESC> or selecting Exit.

h. Answer the following question with YES:



```
Do you wish to save your new kernel configuration?
```

i. On you CLI, there will be messages saying that configurations have been written to "**.config**". Otherwise, you need to start the above process.

15. Build the new kernel with the following commands:

a. "make-kpkg clean"

b. "fakeroot make-kpkg **--**initrd **--**append-to-version=**-beifang** kernel_image kernel_headers" (this is a single one command; do NOT type Enter key until you have typed this command for execution—sometimes, the command will automatically go the "next line" due to the limited width of CLI!!!).

i. In the above command, after "**--**append-to-version=", you can add any string that helps you identify the new kernel (you must begin with a minus (**-**)! Here is "-beifang").

1. You need to use your firstname (NO upper case) for this new version string (I used "beifang")

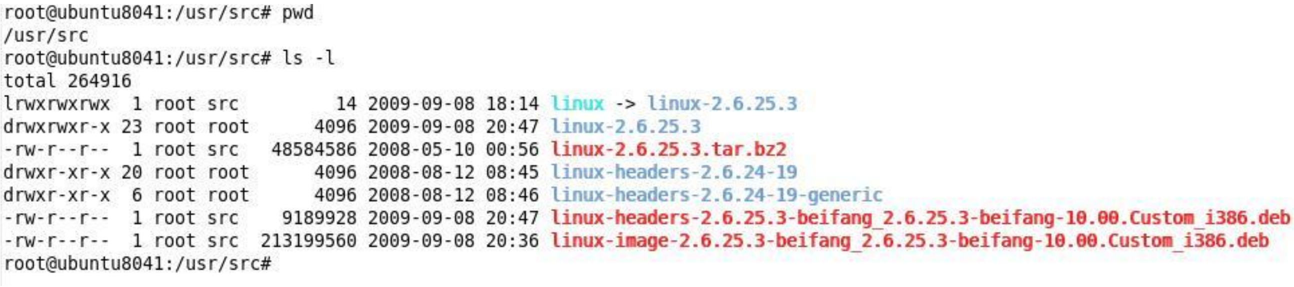
ii. **!!!*Be PATIENT***: the compiling process will take **several hours** (2-5 hours) depending on your machine's configuration.

iii. Examine the outputs (when finished) for any possible exit/errors.

16. Install the new kernel:

    a. Type command “cd /usr/src”;

    b. Type command “ls –l”—The following will be displayed on the CLI:



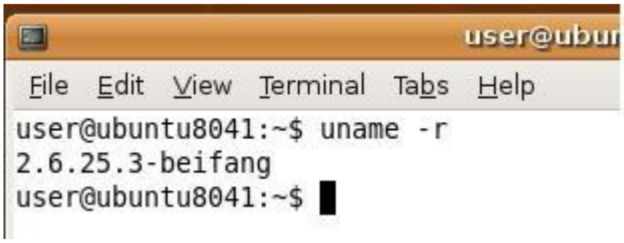
```
root@ubuntu8041:/usr/src# pwd
/usr/src
root@ubuntu8041:/usr/src# ls -l
total 264916
lrwxrwxrwx  1 root src         14 2009-09-08 18:14 linux -> linux-2.6.25.3
drwxrwxr-x 23 root root      4096 2009-09-08 20:47 linux-2.6.25.3
-rw-r--r--  1 root src   48584586 2008-05-10 00:56 linux-2.6.25.3.tar.bz2
drwxr-xr-x 20 root root      4096 2008-08-12 08:45 linux-headers-2.6.24-19
drwxr-xr-x  6 root root      4096 2008-08-12 08:46 linux-headers-2.6.24-19-generic
-rw-r--r--  1 root src    9189928 2009-09-08 20:47 linux-headers-2.6.25.3-beifang_2.6.25.3-beifang-10.00.Custom_i386.deb
-rw-r--r--  1 root src  213199560 2009-09-08 20:36 linux-image-2.6.25.3-beifang_2.6.25.3-beifang-10.00.Custom_i386.deb
root@ubuntu8041:/usr/src#
```

    c. **Take a screenshot like the above for submission**.

    d. There are two files ended with “.deb” with new version script “..beifang…”. Now install them with the following commands:
        i. “dpkg -i linux-image-2.6.25.3-beifang_2.6.25.3-beifang-10.00.Custom_i386.deb”
        ii. “dpkg -i linux-headers-2.6.25.3-beifang_2.6.25.3-beifang-10.00.Custom_i386.deb”

    e. Reboot the system by using:
        i. “shutdown -r now”
        ii. Provide the username and password when asked.

    f. Now you have the new kernel installed/updated on your machine. Check the new version:
        i. “date”
        ii. “uname –r” and you will get something like the following:



```
user@ubuntu8041:~$ uname -r
2.6.25.3-beifang
user@ubuntu8041:~$ 
```

        iii. Take a screenshot of this CLI with the outputs of “date” and “uname –r” for your project submission.

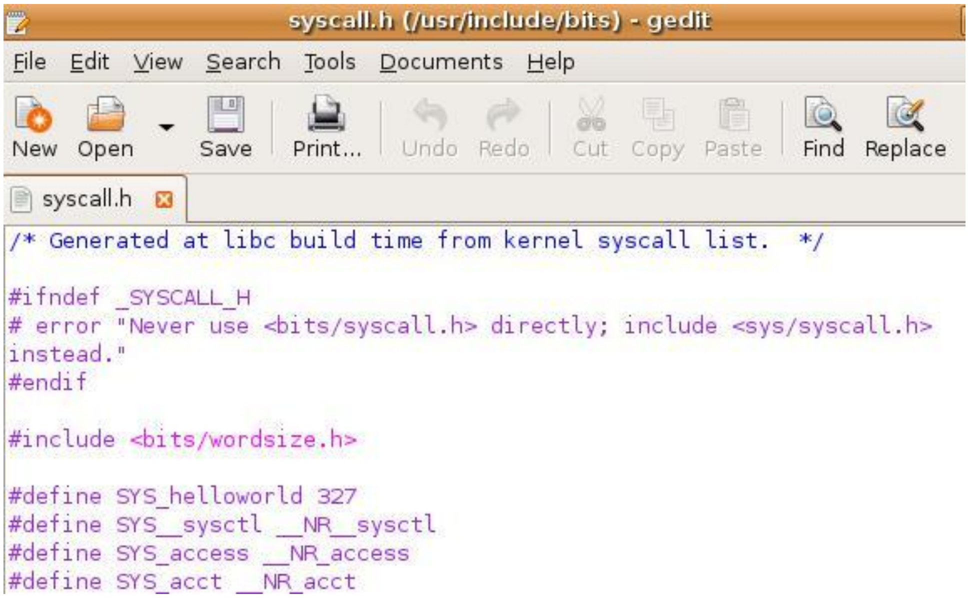
17. Clean the *.o files by using the following commands:
    a. “cd /usr/src/linux”
    b. “su”
    c. “make-kpkg clean”

18. Test the newly inserted system call in the updated Linux kernel:
    a. Remember: we have updated/installed the new Linux kernel but we have not renewed the system’s “include” files (*.h) in their corresponding directories. Thus, when we test the new system call function, we have to use those *.h under the “/usr/src/linux.” A makefile is needed to tell the compiler where to find them. To make things simple, we will take a shortcut illustrated in the following steps.

b. If you are not a root user, type “su” to become a superuser.

c. Edit the system’s syscall.h file by typing :
   i. “gedit /usr/include/bits/syscall.h”

   ii. Insert “#define SYS_helloworld 327” to that file as shown in the following picture (I assume that the unique number you used in **step 12** is 327).

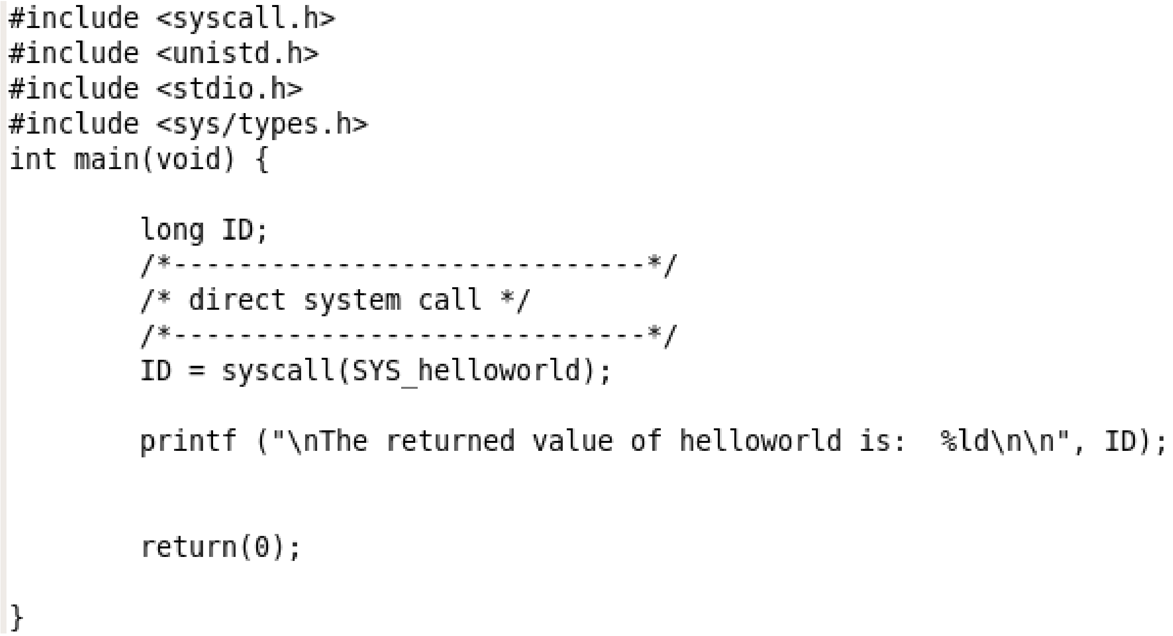

syscall.h (/usr/include/bits) - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
syscall.h

```
/* Generated at libc build time from kernel syscall list.  */

#ifndef _SYSCALL_H
# error "Never use <bits/syscall.h> directly; include <sys/syscall.h>
instead."
#endif

#include <bits/wordsize.h>

#define SYS_helloworld 327
#define SYS__sysctl __NR__sysctl
#define SYS_access __NR_access
#define SYS_acct __NR_acct
```

d. Create a test c program (see the following, with the name of “testNewKernel.c”):

   i. For security, you may need to open another CLI and go to the directory “csc280” which is a new directory (if not created. Use “mkdir csc280” to create it).

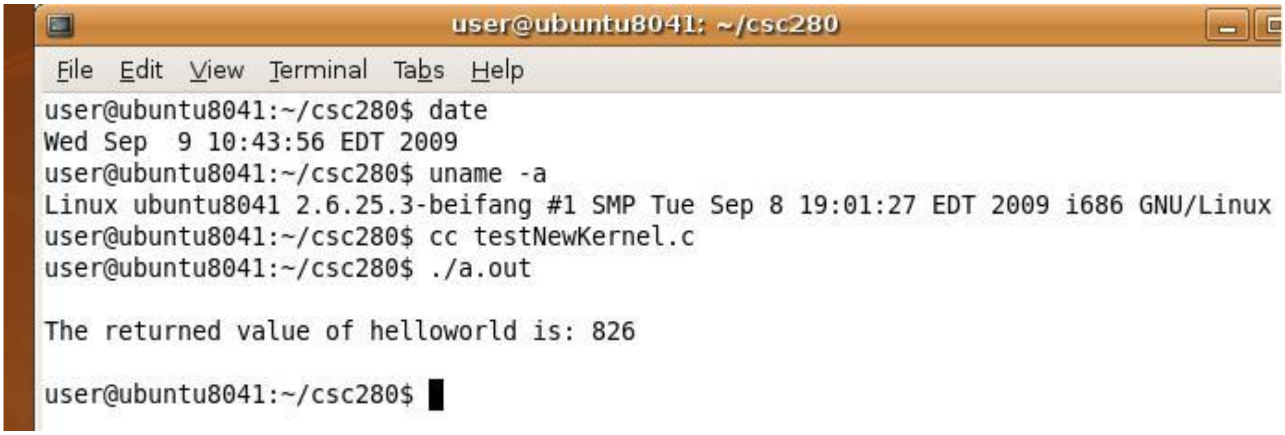
   ii. And then from there, type command “gedit testNewKernel.c” and input the following to the file.



```
#include <syscall.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
int main(void) {

        long ID;
        /*-------------------------------*/
        /* direct system call */
        /*-------------------------------*/
        ID = syscall(SYS_helloworld);

        printf ("\nThe returned value of helloworld is:  %ld\n\n", ID);


        return(0);

}
```

e. Compile this c program with “cc testNewKernel.c”

f. Run the executable file a.out with “./a.out” and you will get the results like the following:



```
user@ubuntu8041:~/csc280$ date
Wed Sep  9 10:43:56 EDT 2009
user@ubuntu8041:~/csc280$ uname -a
Linux ubuntu8041 2.6.25.3-beifang #1 SMP Tue Sep 8 19:01:27 EDT 2009 i686 GNU/Linux
user@ubuntu8041:~/csc280$ cc testNewKernel.c
user@ubuntu8041:~/csc280$ ./a.out

The returned value of helloworld is: 826

user@ubuntu8041:~/csc280$ █
```

g. As you can see, the new system call has been successfully called and it returned the value we defined in the system call code!

h. **Take screenshot** (similar to the above one) with the results of the following commands for your submission:

i. “date”
ii. “uname –a”
iii. “cc testNewKernel.c”
iv. “./a.out”

How to submit:

- Submit one electronic copy by email to the instructor by the due date:
- All your work should be included in **ONLY one SINGLE Word** or **PDF** file, which includes:
  - (The **file name** should be like “project1_YourLastName.pdf”…)
  - (The **subjec**t for the email should be like “CSC280 Project1”)
  - A cover page with your name, project title,…
  - Screenshots taken in the Project.
    - You need to take these shots **in this order as you work through the project**.
- ***Penalty*** will be applied if your submission that does *not* follow the above instructions.