“ADDING A SYSTEM CALL TO LINUX KERNEL”

PROJECT REPORT

Submitted for the course: Operating Systems (CSE-2005)

By

Vignesh Kumar S	**16BCE0566**
Yarlagadda Nikhitha	**16BCI0071**
Sowmya Saraswathi	**16BCI0095**

Slot: A2

Name of faculty: Professor Vijayarajan V

(SCHOOL OF COMPUTER SCIENCE AND ENGINEERING)

Nov, 2017

CERTIFICATE

This is to certify that the project work entitled "*ADDING A SYSTEM CALL TO LINUX KERNEL*" that is being submitted by "***Yarlagadda Nikhitha, Sowmya Saraswathi, Vignesh Kumar S*** " for Operating Systems CSE(2005) is a record of bonafide work done under my supervision. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted for any other CAL course.

Place: Vellore

Date: November 5, 2016

Signature of Students:

Vignesh Kumar S
Yarlagadda Nikhitha
Sowmya Saraswathi

Signature of Faculty:

Professor Vijayarajan V

ACKNOWLEDGEMENTS

a) First and foremost, we would like to thank VIT University Management for giving us permission to carry out this project in the University.
b) We would also like to thank our esteemed school Dean, for providing us with proper facilities for the experimentation phase of the project.
c) We would like to thank our teacher Vijayarajan V., for giving us the opportunity to take up this project.

Vignesh Kumar S 16BCE0566
Yarlagadda Nikhitha 16BCI0071
Sowmya Saraswathi 16BCI0095

ABSTRACT

We would like to thank our guide, Dr. Vijayarajan V for the time, support and knowledge he has granted us. We would like to also thank the Head of the Department Prof. Senthilkumar R. and the Dean Dr. Arunkumar T. This project would not have been possible without their guidance. We are highly grateful to VIT University for providing a platform to achieve academic success. We would also like to express our gratitude to our loved ones who have supported us throughout the whole process.

INTRODUCTION

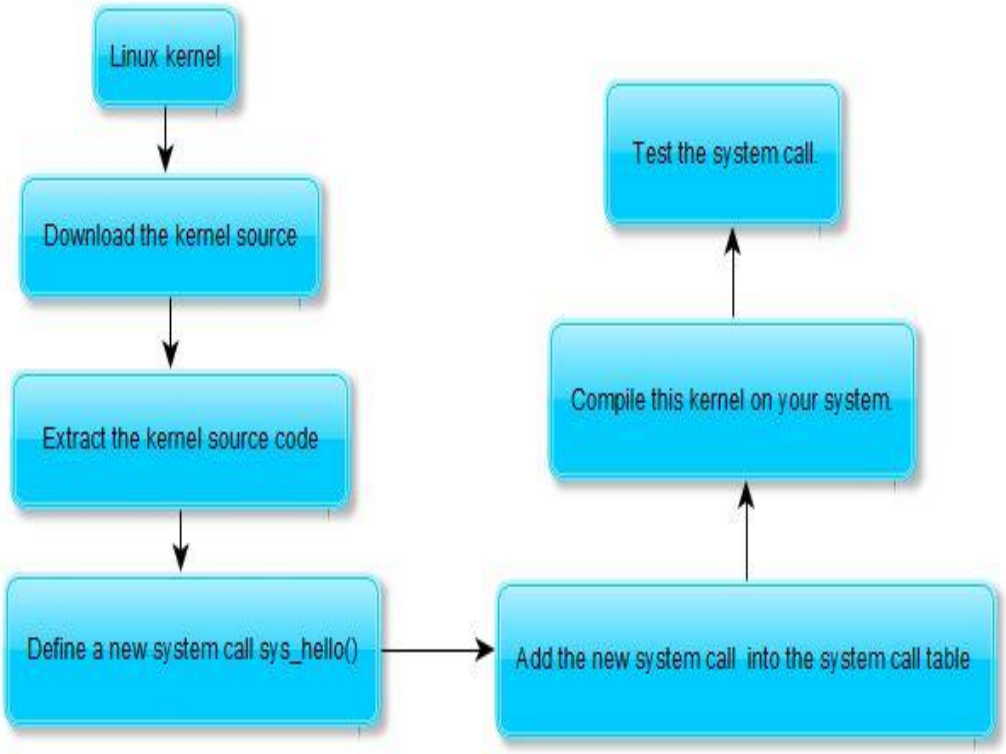
The Linux kernel is written in the version of the C programming language supported by GCC (which has introduced a number of extensions and changes to standard C), together with a number of short sections of code written in the assembly language of the target architecture. Because of the extensions to C it supports, GCC was for a long time the only compiler capable of correctly building the Linux kernel.

A system call is implemented in the Linux kernel. When a program makes a system call, the arguments are packaged up and handed to the kernel, which takes over execution of the program until the call completes. A system call isn't an ordinary function call, and a special procedure is required to transfer control to the kernel. However, the GNU C library (the implementation of the standard C library provided with GNU/Linux systems) wraps Linux system calls with functions so that you can call them easily. Low-level I/O functions such as open and read are examples of system calls on Linux.

PROJECT PLAN

We can customize the linux kernel by adding new system calls to make it more user friendly. We have chosen to implement the Hello world system call to the linux kernel.

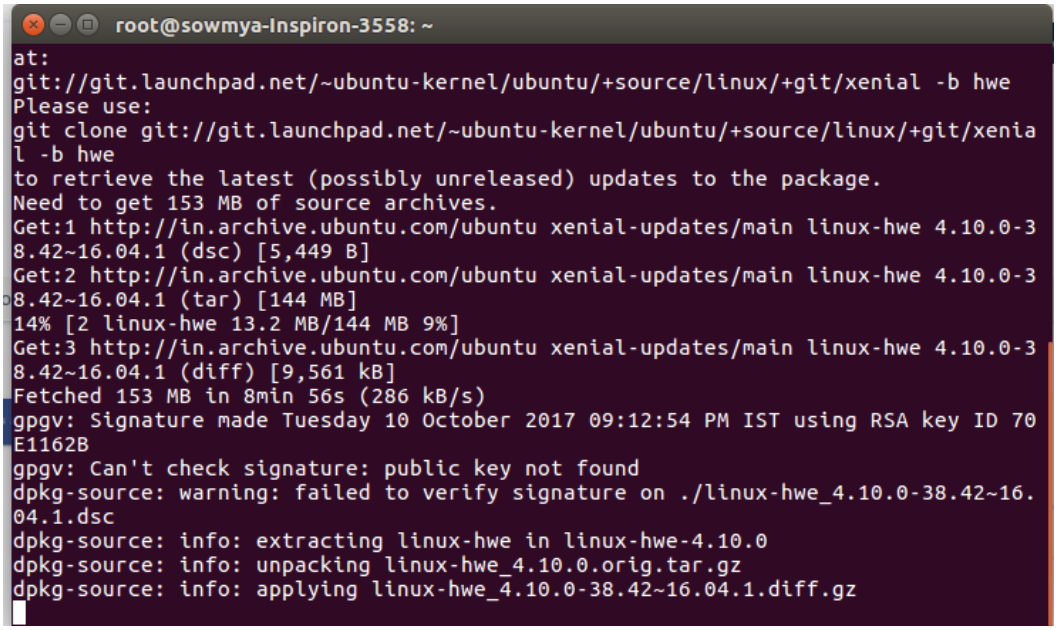
Design and architecture

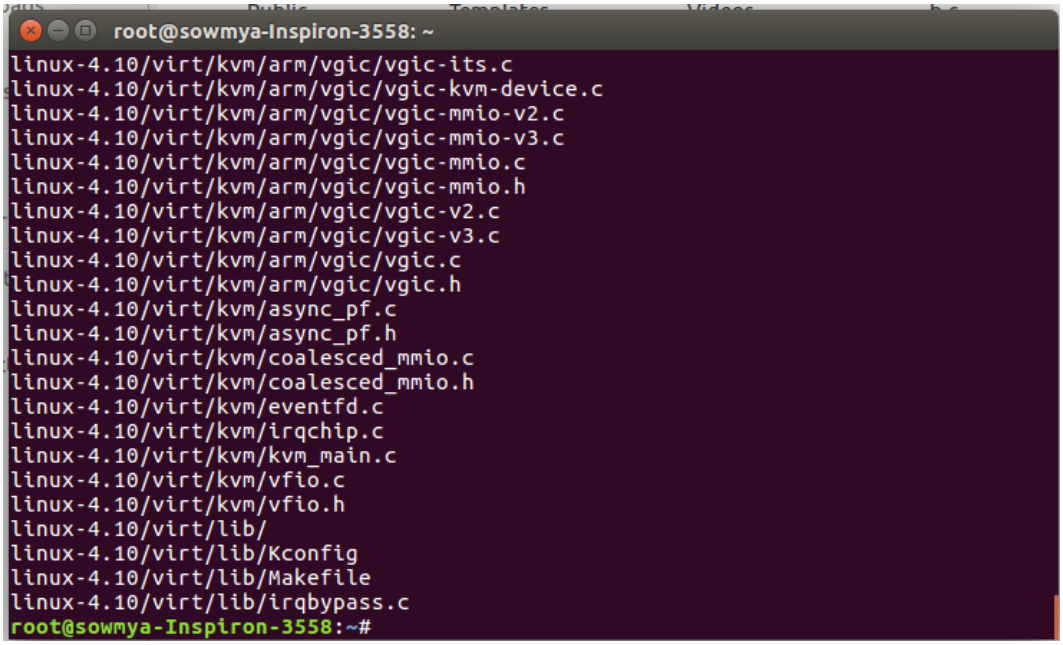


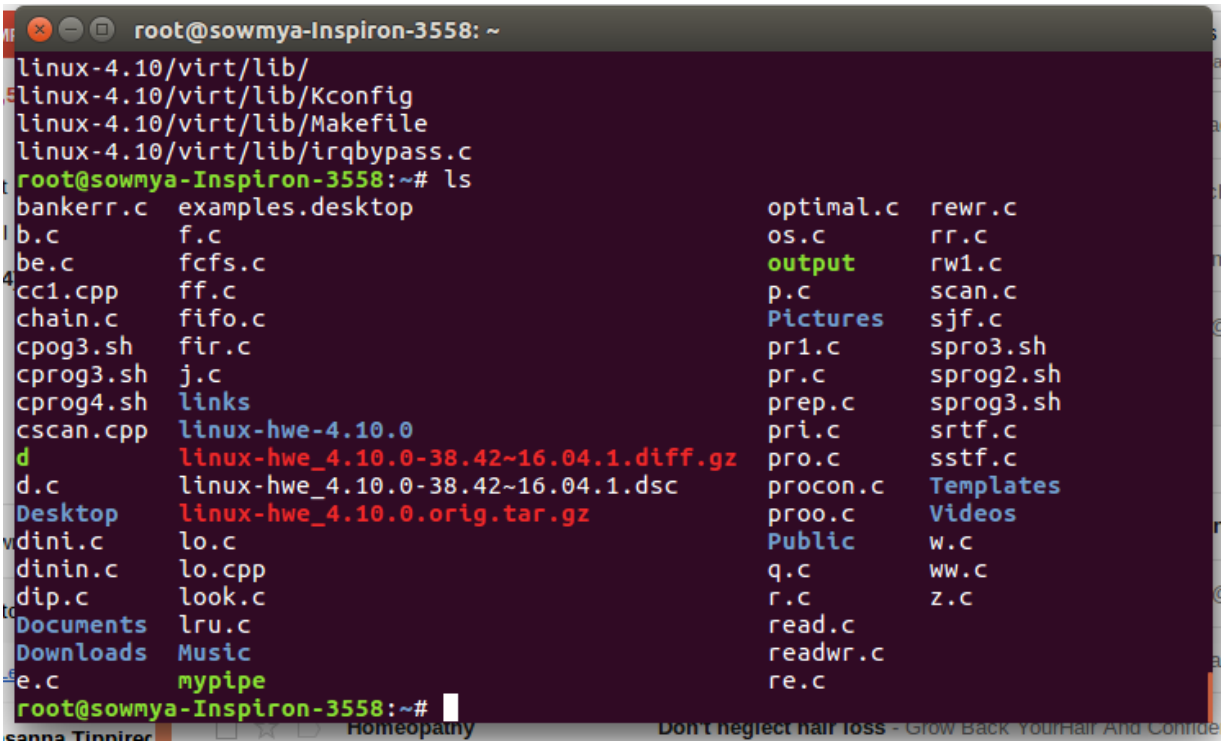
WORKING METHODOLOGY

The steps followed for adding the system calls are as follows:

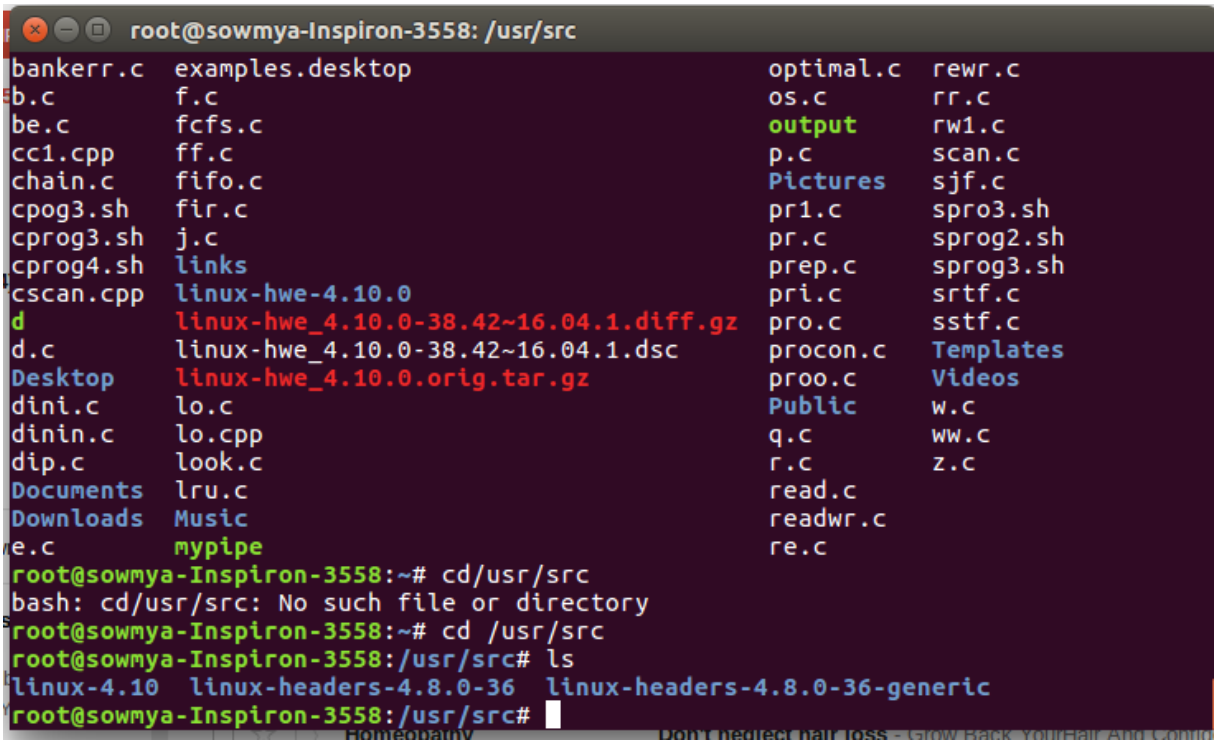
1. uname –r : know your linux version by the command
   Our version is linux-4.10.0
2. sudo –s : directs the control to the root.
3. apt-get source linux-image-4.10.0-36-generic: Download the source code of your kernel version using this command.



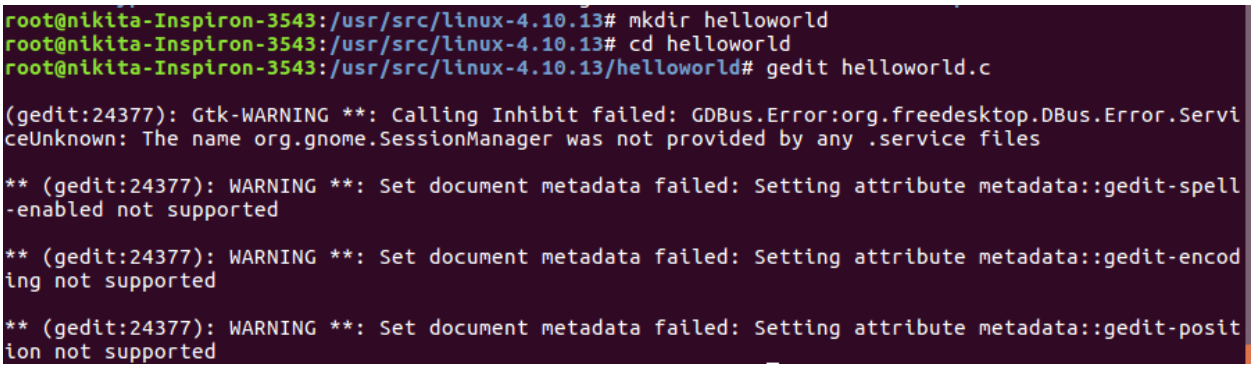




4. ls: using this command we can see all the existing files in the current directory.(Note that linux-hwe_4.10.10.orig.tar.gz is the newly installed file)



5. Cd /usr/src: directs the user to the usr/src directory.(This is where the kernel source code is present).
6. Cd /linux-4.10:direct to the linux-4.10 directory



7. mkdir helloworld: creating a directory for the helloworld file
8. cd helloworld: changes the shell's working directory
9. gedit helloworld.c: it is a text editor which will open the file named 'helloworld.c' for editing.(This is where the C program for our system call is written).

Open — helloworld.c — /usr/src/linux-4.10.13/helloworld — Save

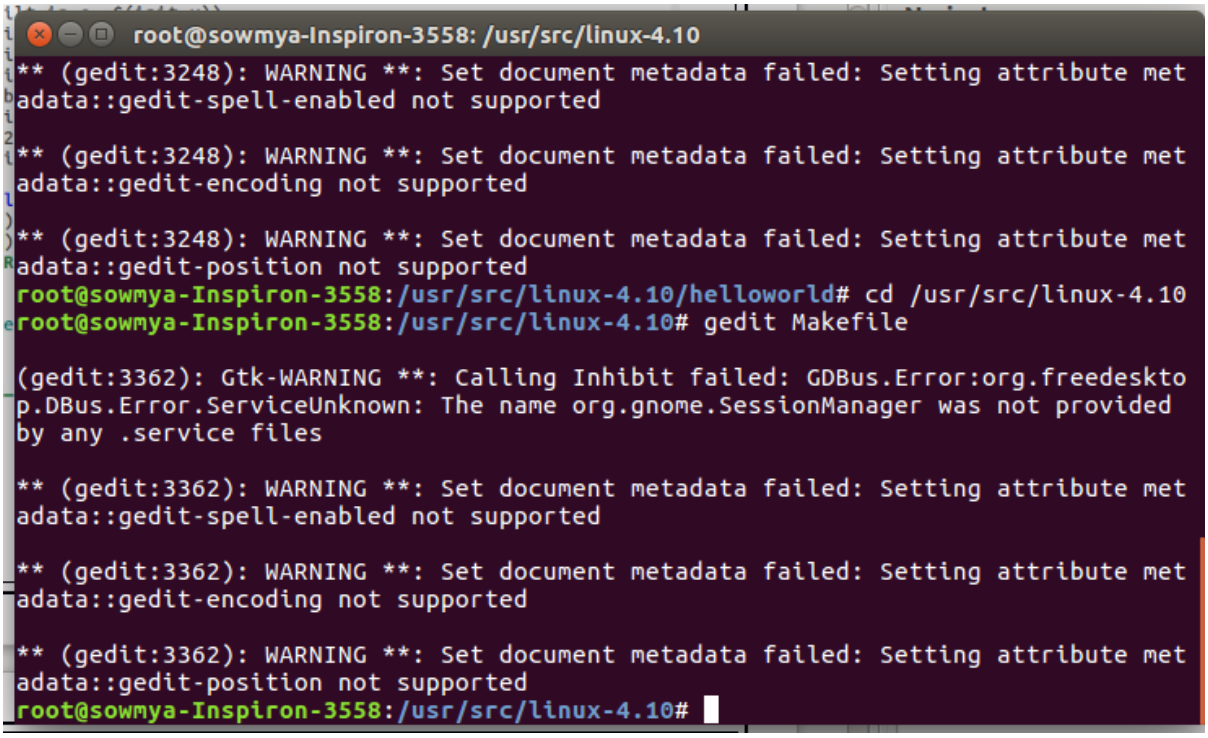
```
#include <linux/kernel.h>

asmlinkage long sys_helloworld(void)
{
        printk("HELLO WORLD\n");
        return 0;
}
```

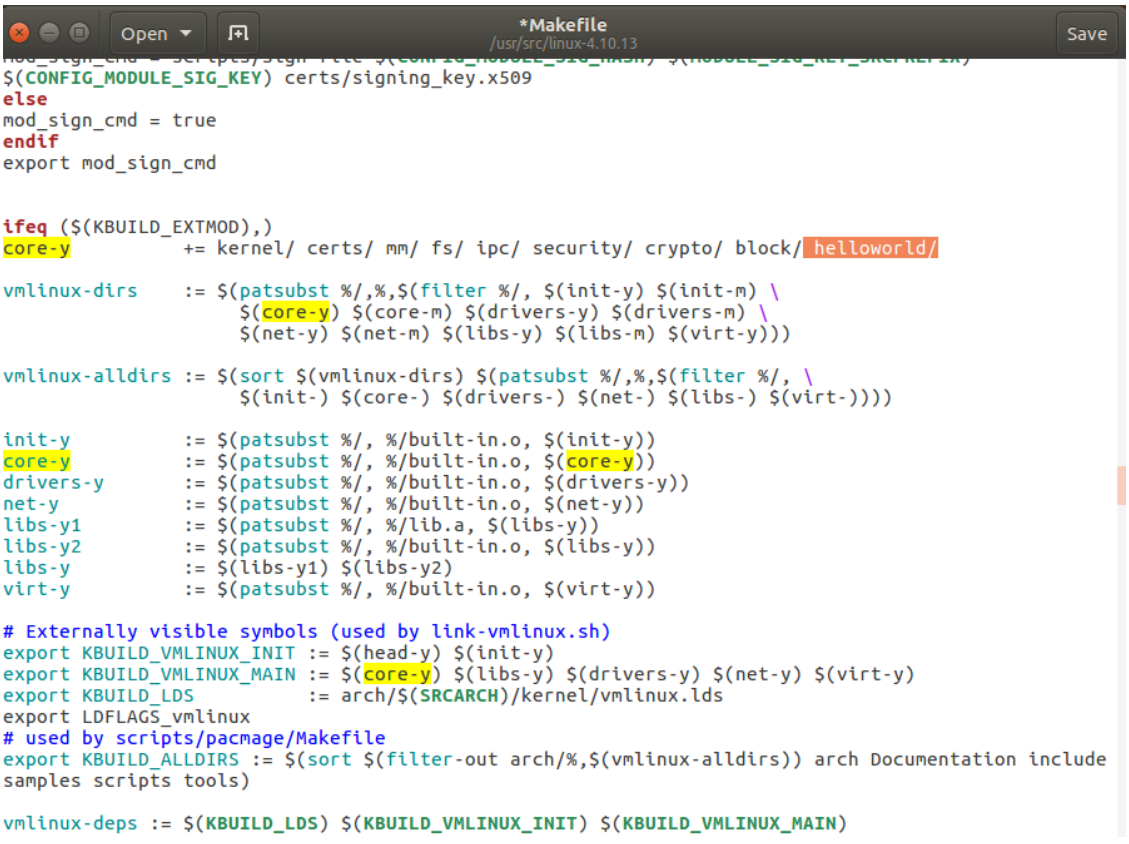
```
root@nikita-Inspiron-3543:/usr/src/linux-4.10.13/helloworld# gedit Makefile

(gedit:24476): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedesktop.DBus.Error.Servi
ceUnknown: The name org.gnome.SessionManager was not provided by any .service files

** (gedit:24476): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-spell
-enabled not supported

** (gedit:24476): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-encod
ing not supported

** (gedit:24476): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-posit
ion not supported
```

Open — Makefile — /usr/src/linux-4.10.13/helloworld — Save

```
obj-y :=helloworld.o
```

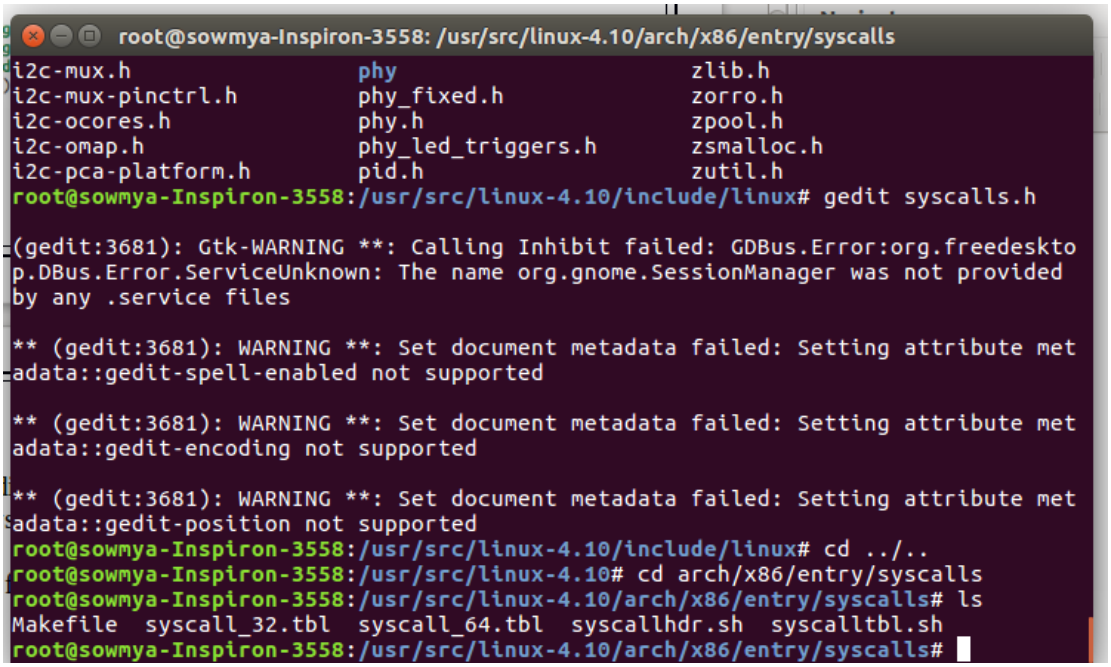
Gedit makefile: it is a text editor which will open the file called makefile for editing, this means that kbuild should go into the directory helloworld. Once it moves to this directory, it looks at the Makefile in it to decide what objects should be built.

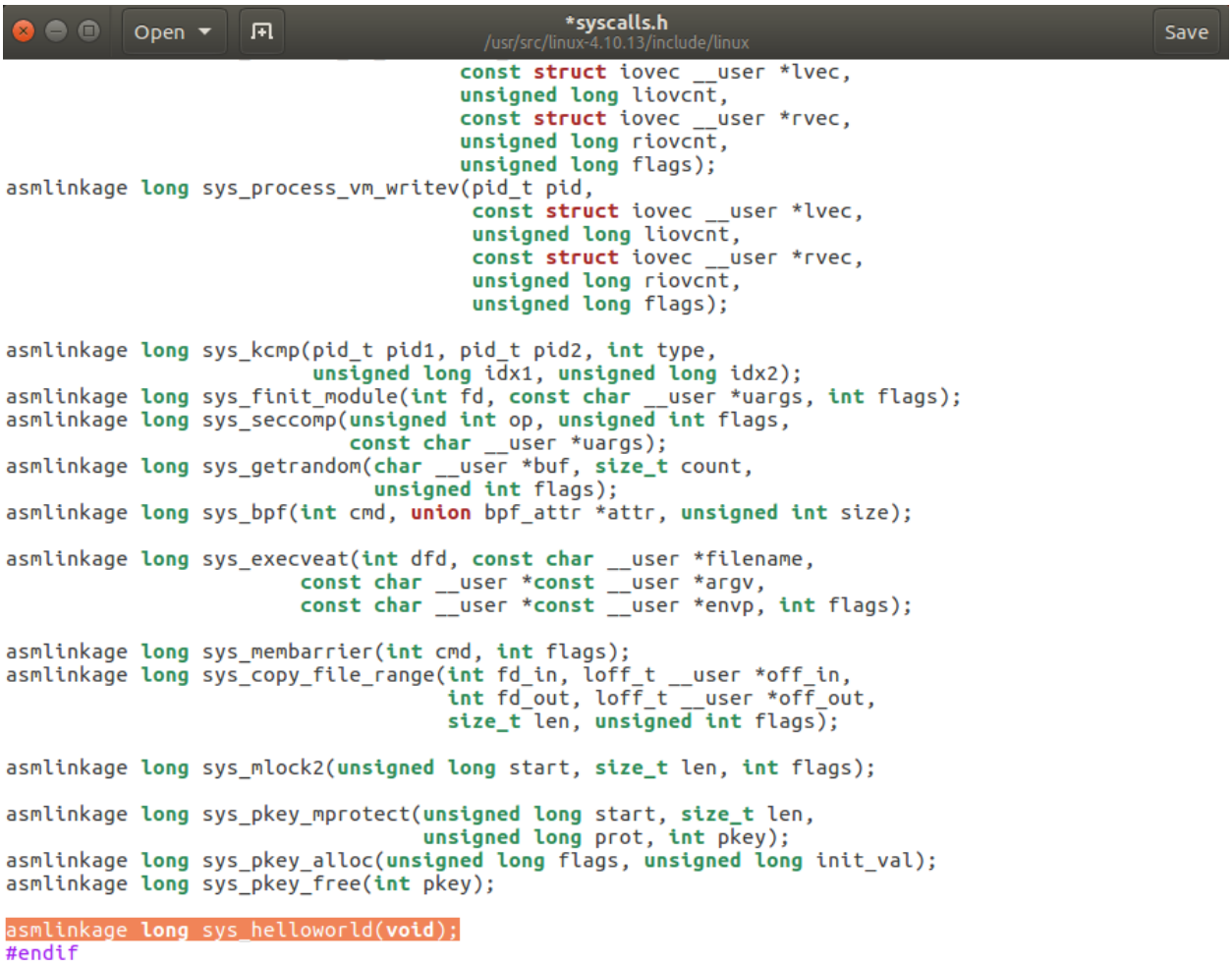




10. Cd /usr/src/linux-4.20: go back to the linux-4.10 directory

11. Gedit makefile: it is a text editor which will open the file called makefile for editing.(Here edit the file by adding the systemcall function name with "/").





12. cd include/linux: we are further going to the directory include/linux

13. gedit syscalls.h: it is a text editor which opens the file called syscalls.h to edit it. It has a log of all system calls. Here we add our system call with asm linkage. Asm linkage tells your compiler to look on the CPU stack for the function parameters, instead of registers. The interesting part is **why** this is necessary. System calls are *services* that user space can call to request the kernel to perform something for them (and therefore execute in kernel space). These functions are quite unorthodox in the sense that you cannot expect them to behave like normal functions, where parameters are typically passed by writing to the program stack, but instead they are written to registers. While still in user space, calling a syscall requires writing certain values to certain registers stacks in the CPU is translated.

So, since all the information about the parameters passed all the way from user land to this point is nicely stored in the stack, the compiler must be instructed about this, hence the **asm linkage**.

```
root@nikita-Inspiron-3543:/usr/src/linux-4.10.13/include/linux# cd ../..
root@nikita-Inspiron-3543:/usr/src/linux-4.10.13# cd arch/x86/entry/syscalls
root@nikita-Inspiron-3543:/usr/src/linux-4.10.13/arch/x86/entry/syscalls# ls
Makefile  syscall_32.tbl  syscall_64.tbl  syscallhdr.sh  syscalltbl.sh
```

14. edit syscall_64.tbl: it is an editor which opens the syscall_64.tbl file. It opens up the syscall table containing all system calls and we add our system call at the end index number, in this case our system call index number is 332.

```
root@nikita-Inspiron-3543:/usr/src/linux-4.10.13/arch/x86/entry/syscalls# gedit syscall_64.tbl

(gedit:24829): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedesktop.DBus.Error.Servi
ceUnknown: The name org.gnome.SessionManager was not provided by any .service files

** (gedit:24829): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-spell
-enabled not supported

** (gedit:24829): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-encod
ing not supported

** (gedit:24829): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-spell
-enabled not supported

** (gedit:24829): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-encod
ing not supported

** (gedit:24829): WARNING **: Set document metadata failed: Setting attribute metadata::gedit-posit
ion not supported
```

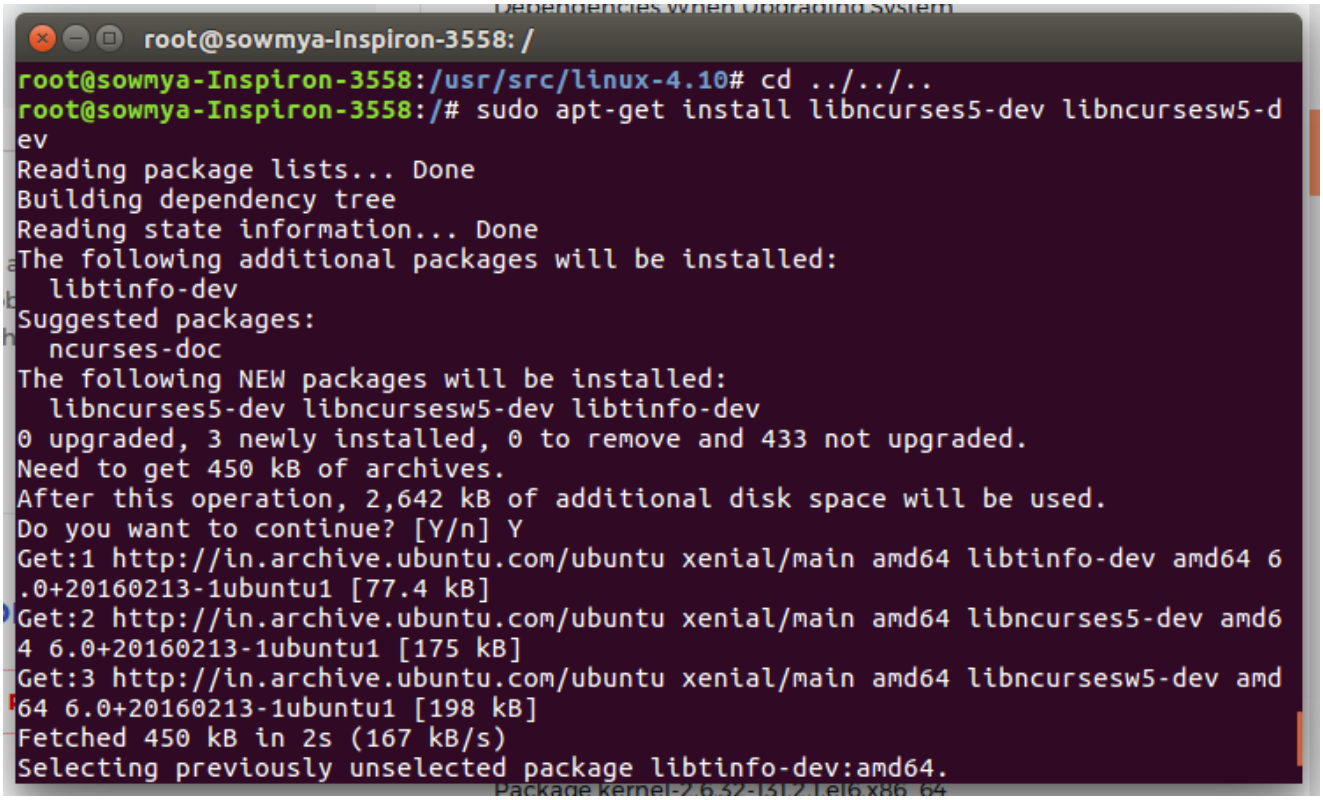
Open — syclall_64.tbl — /usr/src/linux-4.10.13/arch/x86/entry/syscalls — Save

```
315     common  sched_getattr          sys_sched_getattr
316     common  renameat2              sys_renameat2
317     common  seccomp                sys_seccomp
318     common  getrandom              sys_getrandom
319     common  memfd_create           sys_memfd_create
320     common  kexec_file_load        sys_kexec_file_load
321     common  bpf                    sys_bpf
322     64      execveat               sys_execveat/ptregs
323     common  userfaultfd            sys_userfaultfd
324     common  membarrier             sys_membarrier
325     common  mlock2                 sys_mlock2
326     common  copy_file_range        sys_copy_file_range
327     64      preadv2                sys_preadv2
328     64      pwritev2               sys_pwritev2
329     common  pkey_mprotect          sys_pkey_mprotect
330     common  pkey_alloc             sys_pkey_alloc
331     common  pkey_free              sys_pkey_free
332     64      helloworld             sys_helloworld

#
# x32-specific system call numbers start at 512 to avoid cache impact
# for native 64-bit operation.
#
512     x32     rt_sigaction           compat_sys_rt_sigaction
513     x32     rt_sigreturn           sys32_x32_rt_sigreturn
514     x32     ioctl                  compat_sys_ioctl
515     x32     readv                  compat_sys_readv
516     x32     writev                 compat_sys_writev
517     x32     recvfrom               compat_sys_recvfrom
518     x32     sendmsg                compat_sys_sendmsg
519     x32     recvmsg                compat_sys_recvmsg
520     x32     execve                 compat_sys_execve/ptregs
521     x32     ptrace                 compat_sys_ptrace
522     x32     rt_sigpending          compat_sys_rt_sigpending
523     x32     rt_sigtimedwait        compat_sys_rt_sigtimedwait
524     x32     rt_sigqueueinfo        compat_sys_rt_sigqueueinfo
525     x32     sigaltstack            compat_sys_sigaltstack
526     x32     timer_create           compat_sys_timer_create
527     x32     mq_notify              compat_sys_mq_notify
```

Plain Text — Tab Width: 8 — Ln 341, Col 1 — INS

15.cd ../..: changing directory from syscall to the initial directory

```
root@nikita-Inspiron-3543:/usr/src/linux-4.10.13/arch/x86/entry/syscalls# cd ../..
root@nikita-Inspiron-3543:/usr/src/linux-4.10.13/arch/x86# cd ../..
```



Cd ./../..:

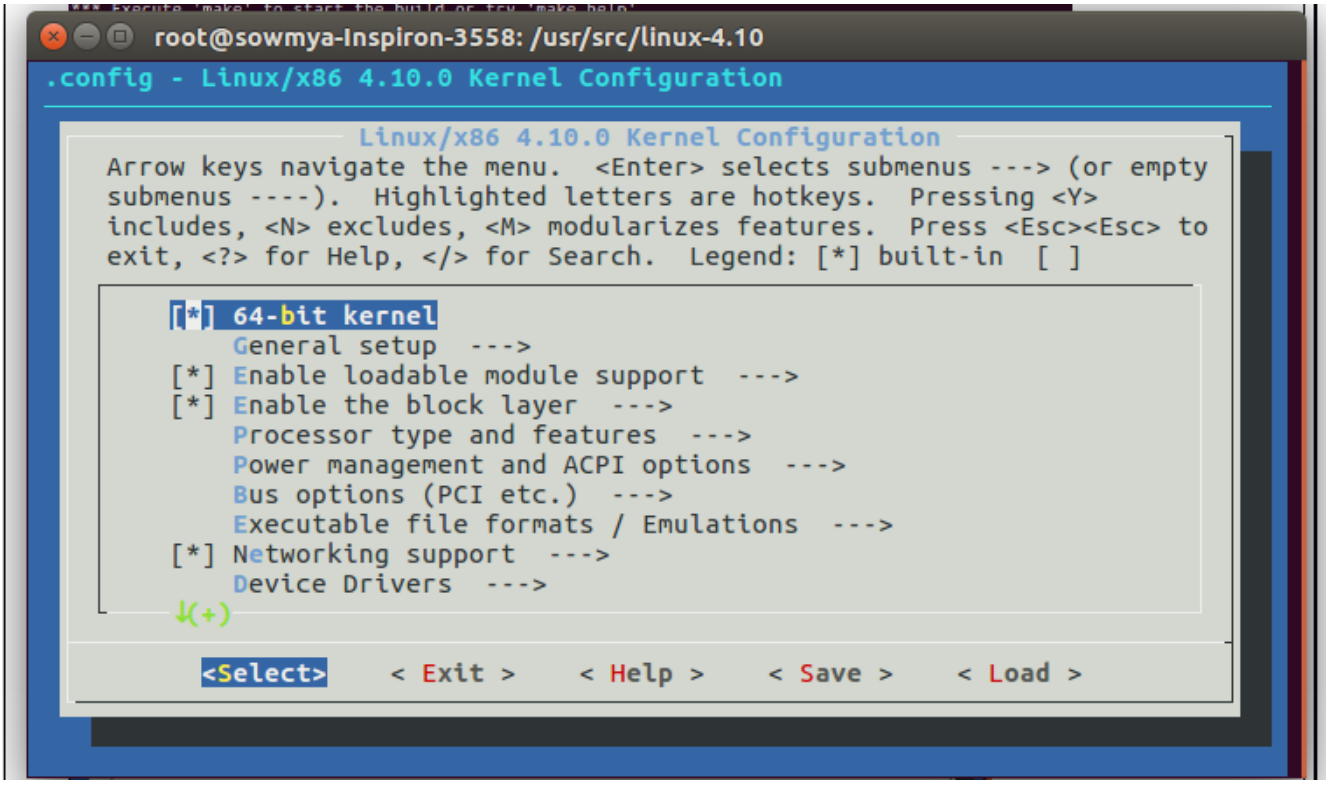
Go back by three directories

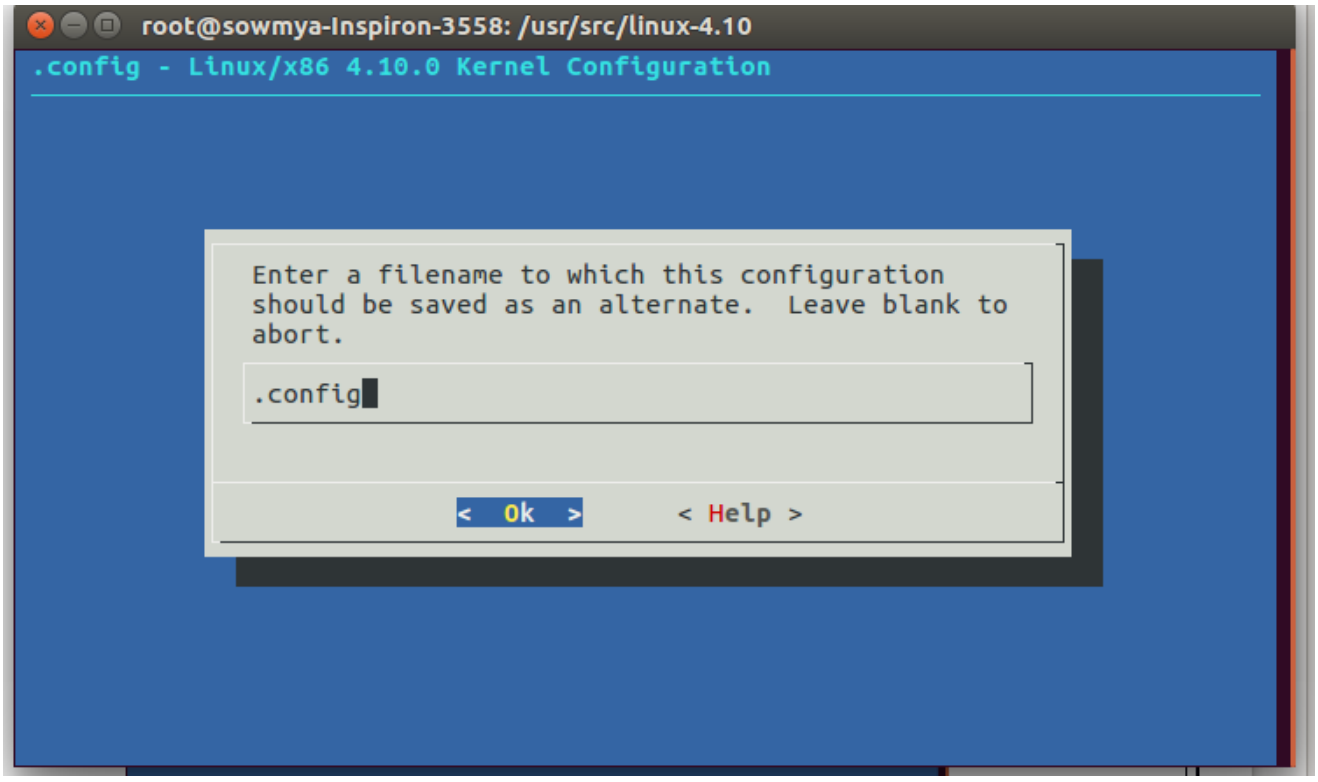
Sudo apt-get install libncurses5-dev libncursesw5-dev:

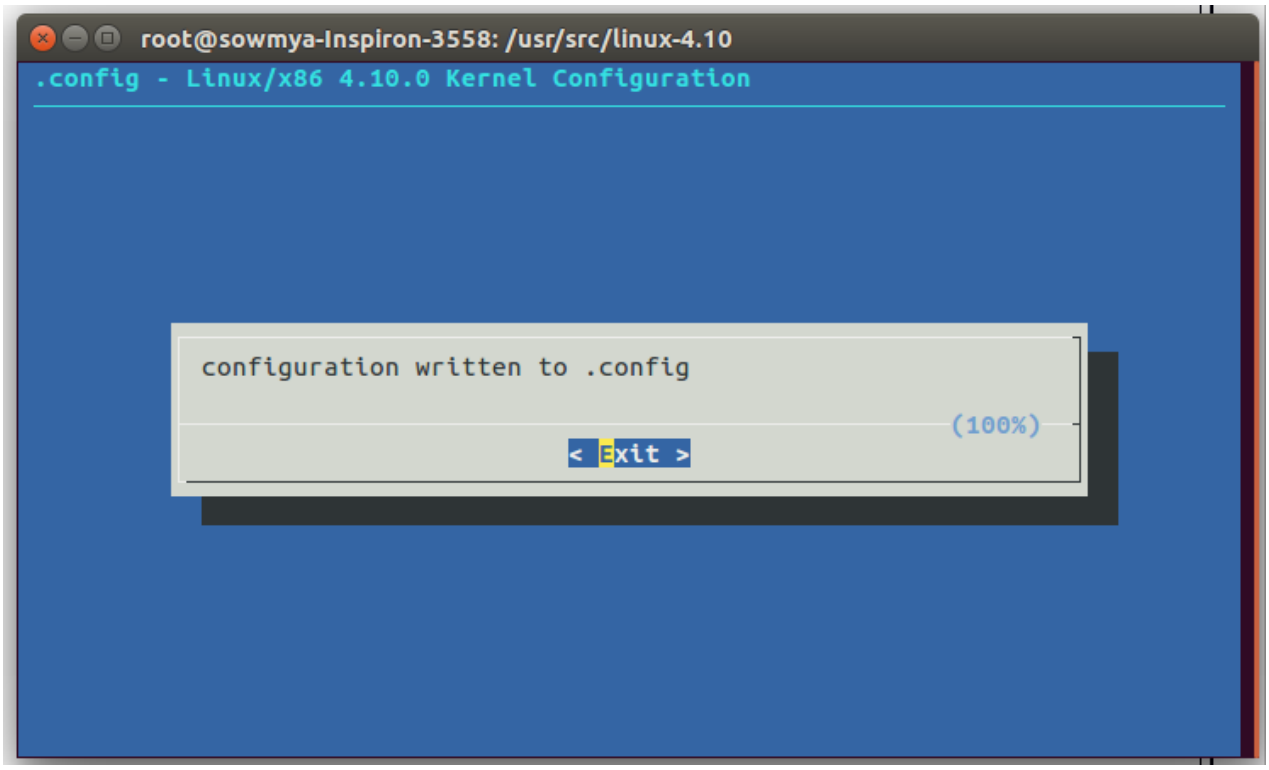
Using this command we install the libncurses5, libncursesw5 packages.

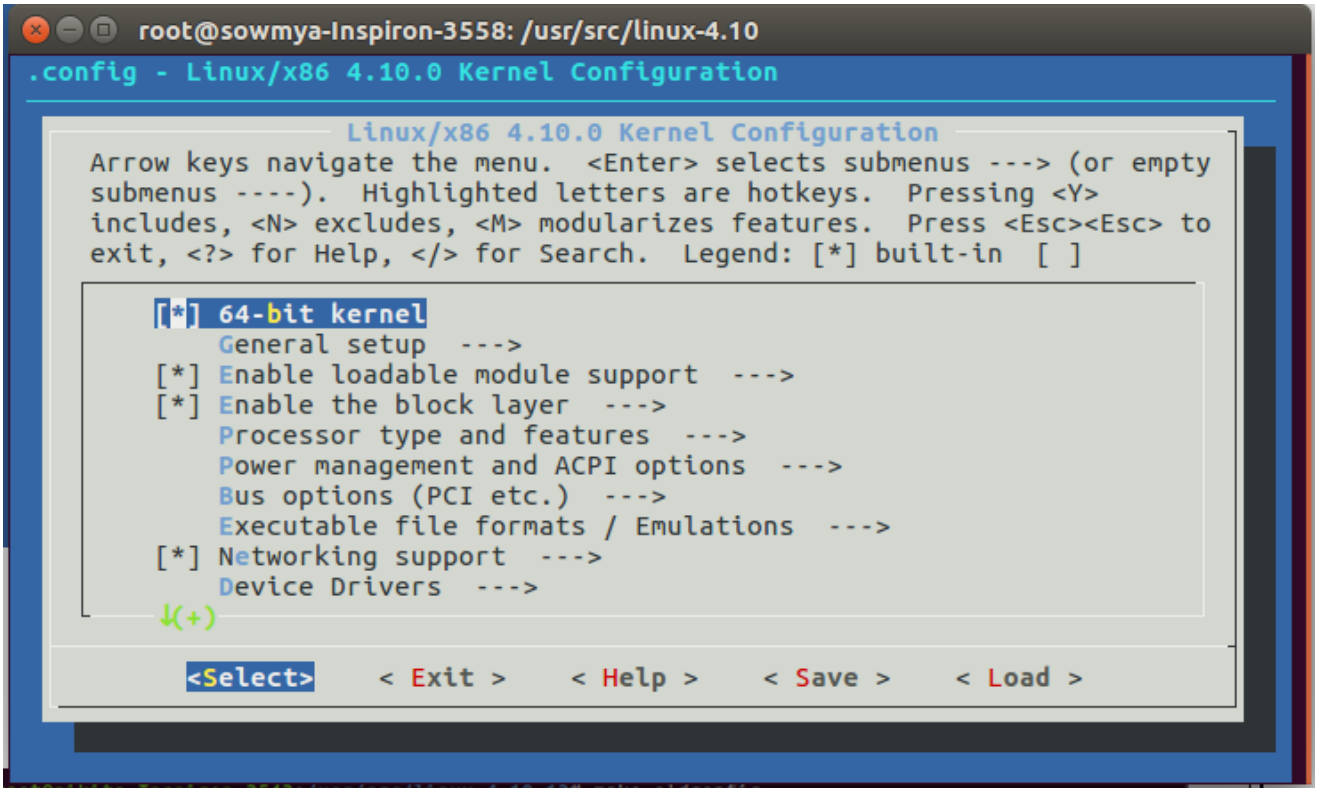
```
root@nikita-Inspiron-3543:/usr/src/linux-4.10.13# make menuconfig
  HOSTCC  scripts/basic/fixdep
  HOSTCC  scripts/kconfig/mconf.o
  SHIPPED scripts/kconfig/zconf.tab.c
  SHIPPED scripts/kconfig/zconf.lex.c
  SHIPPED scripts/kconfig/zconf.hash.c
  HOSTCC  scripts/kconfig/zconf.tab.o
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
  HOSTCC  scripts/kconfig/lxdialog/util.o
  HOSTCC  scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  scripts/kconfig/lxdialog/textbox.o
  HOSTCC  scripts/kconfig/lxdialog/yesno.o
  HOSTCC  scripts/kconfig/lxdialog/menubox.o
  HOSTLD  scripts/kconfig/mconf
scripts/kconfig/mconf  Kconfig
#
# using defaults found in /boot/config-4.4.0-31-generic
#
/boot/config-4.4.0-31-generic:958:warning: symbol value 'm' invalid for NF_CT_PROTO_DCCP
/boot/config-4.4.0-31-generic:960:warning: symbol value 'm' invalid for NF_CT_PROTO_SCTP
/boot/config-4.4.0-31-generic:961:warning: symbol value 'm' invalid for NF_CT_PROTO_UDPLITE
/boot/config-4.4.0-31-generic:979:warning: symbol value 'm' invalid for NF_NAT_PROTO_DCCP
/boot/config-4.4.0-31-generic:980:warning: symbol value 'm' invalid for NF_NAT_PROTO_UDPLITE
/boot/config-4.4.0-31-generic:981:warning: symbol value 'm' invalid for NF_NAT_PROTO_SCTP
/boot/config-4.4.0-31-generic:1631:warning: symbol value 'm' invalid for RXKAD
/boot/config-4.4.0-31-generic:3589:warning: symbol value 'm' invalid for SERIAL_8250_FINTEK
/boot/config-4.4.0-31-generic:7526:warning: symbol value 'm' invalid for EXT4_ENCRYPTION


*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.
```

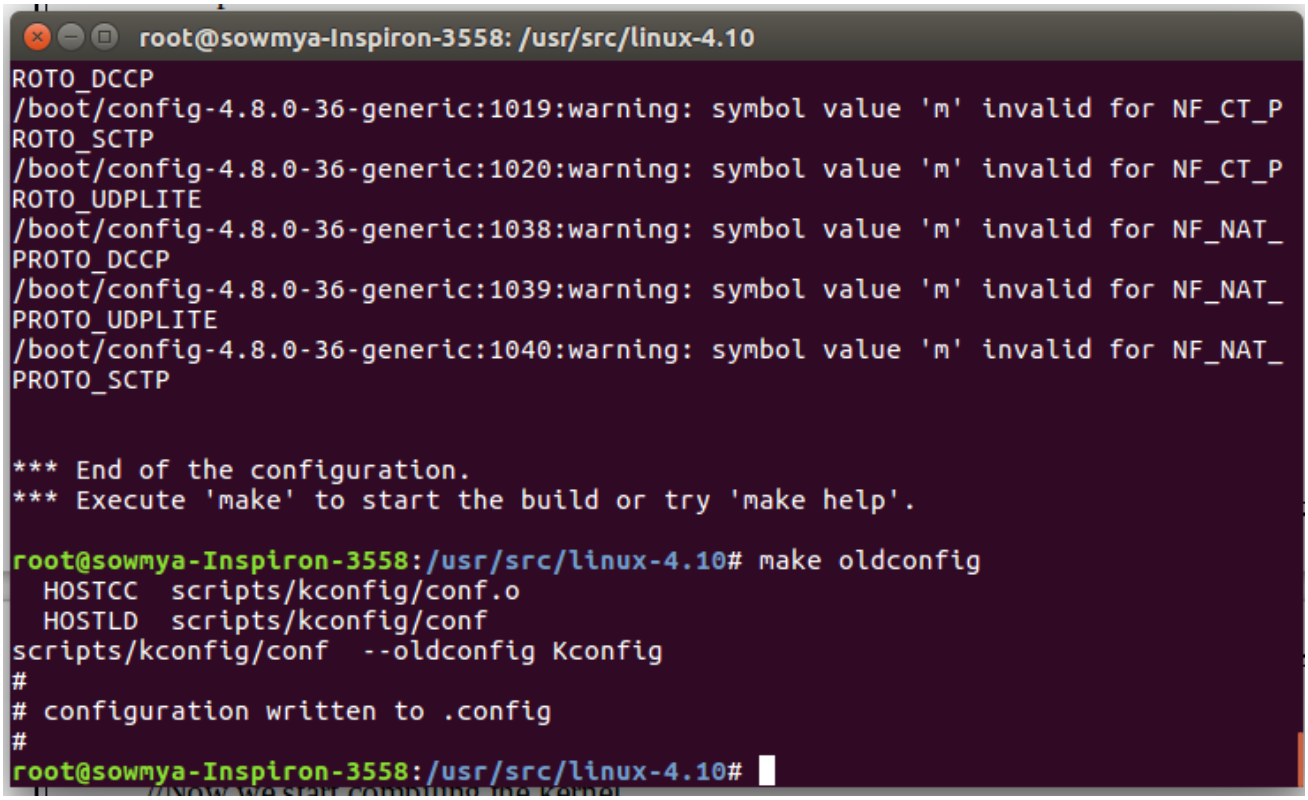
16. make menuconfig: make menuconfig is one of five similar tools that can configure Linux source, a necessary early step needed to compile the source code. make menuconfig, with a menu-driven user interface, allows the user to choose the features of Linux (and other options) that will be compiled. It is normally invoked using the command `make menuconfig`, menuconfig is a target in Linux Makefile.

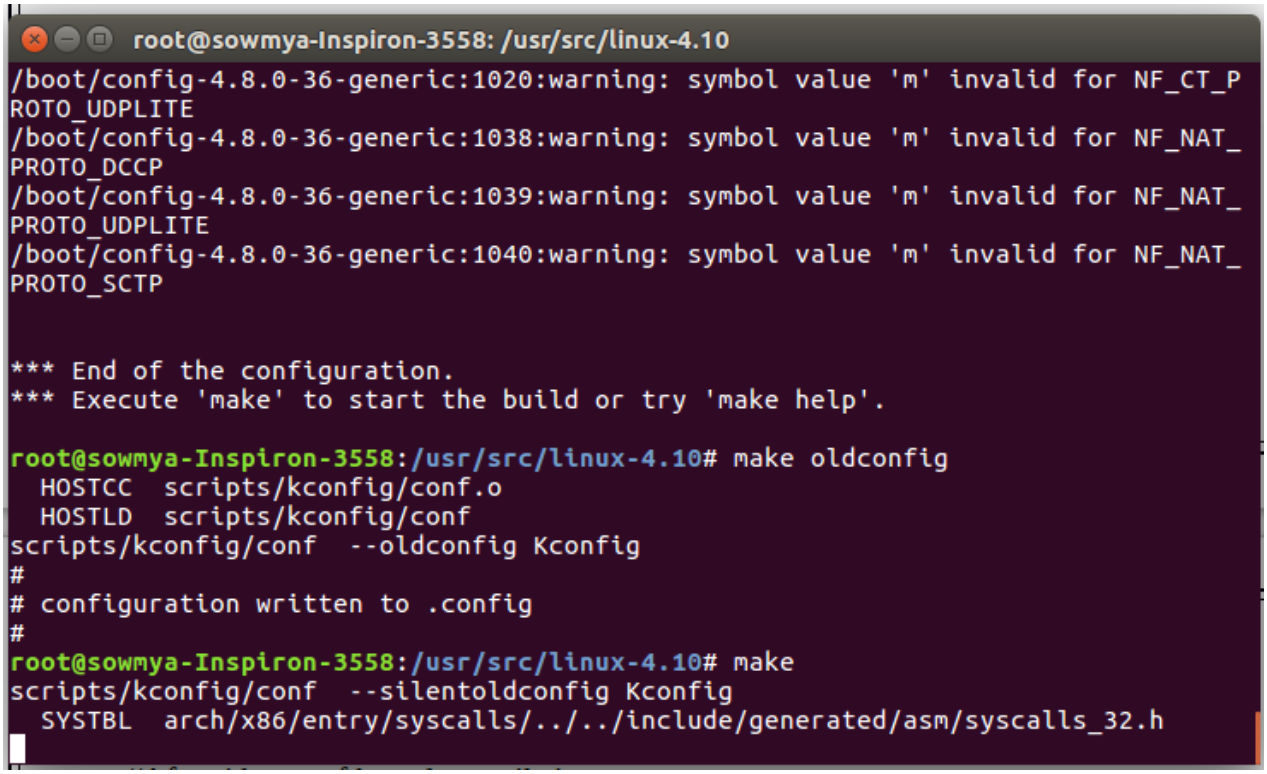

root@sowmya-Inspiron-3558: /usr/src/linux-4.10
.config - Linux/x86 4.10.0 Kernel Configuration
Linux/x86 4.10.0 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [ ]
[*] 64-bit kernel
General setup --->
[*] Enable loadable module support --->
[*] Enable the block layer --->
Processor type and features --->
Power management and ACPI options --->
Bus options (PCI etc.) --->
Executable file formats / Emulations --->
[*] Networking support --->
Device Drivers --->
<Select> < Exit > < Help > < Save > < Load >

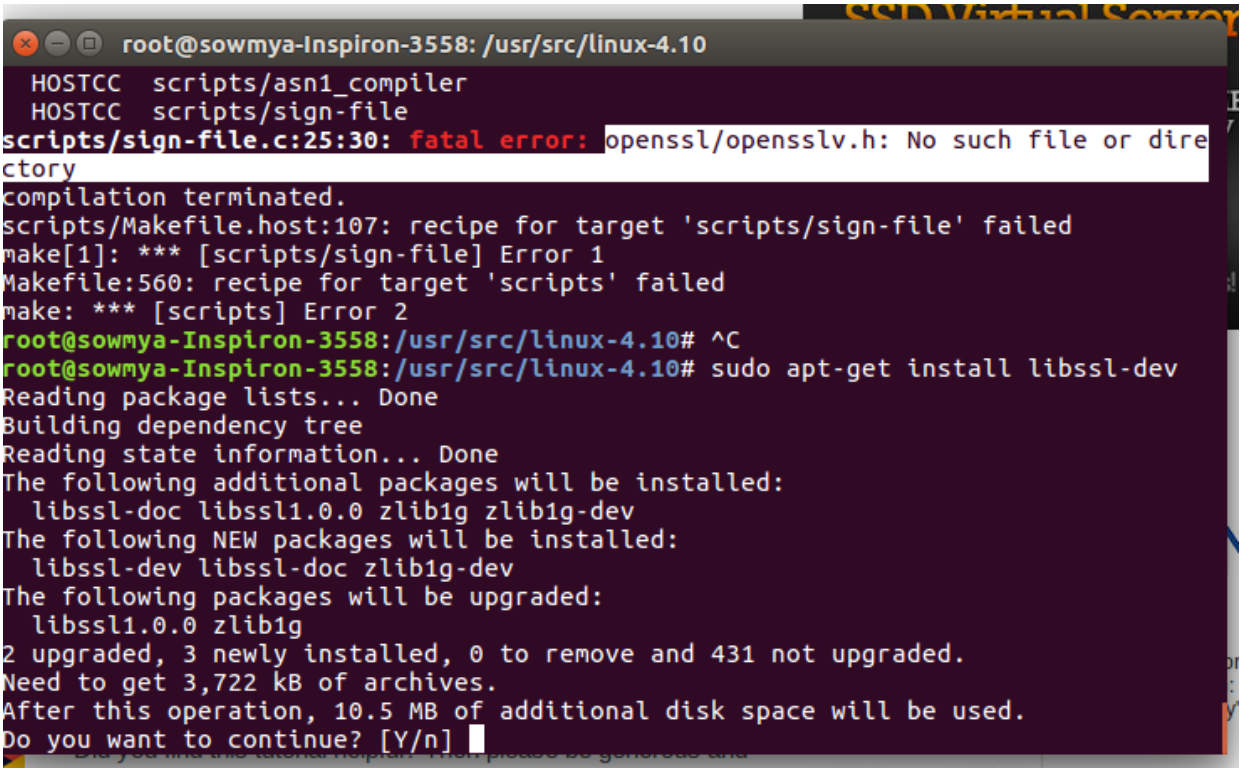

root@sowmya-Inspiron-3558: /usr/src/linux-4.10
.config - Linux/x86 4.10.0 Kernel Configuration
Enter a filename to which this configuration should be saved as an alternate. Leave blank to abort.
.config
< Ok > < Help >


root@sowmya-Inspiron-3558: /usr/src/linux-4.10
.config - Linux/x86 4.10.0 Kernel Configuration
configuration written to .config
(100%)
< Exit >


root@sowmya-Inspiron-3558: /usr/src/linux-4.10
.config - Linux/x86 4.10.0 Kernel Configuration
Linux/x86 4.10.0 Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty 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 [ ]
[*] 64-bit kernel
General setup --->
[*] Enable loadable module support --->
[*] Enable the block layer --->
Processor type and features --->
Power management and ACPI options --->
Bus options (PCI etc.) --->
Executable file formats / Emulations --->
[*] Networking support --->
Device Drivers --->
↓(+)
<Select> < Exit > < Help > < Save > < Load >


root@sowmya-Inspiron-3558: /usr/src/linux-4.10
ROTO_DCCP
/boot/config-4.8.0-36-generic:1019:warning: symbol value 'm' invalid for NF_CT_P
ROTO_SCTP
/boot/config-4.8.0-36-generic:1020:warning: symbol value 'm' invalid for NF_CT_P
ROTO_UDPLITE
/boot/config-4.8.0-36-generic:1038:warning: symbol value 'm' invalid for NF_NAT_
PROTO_DCCP
/boot/config-4.8.0-36-generic:1039:warning: symbol value 'm' invalid for NF_NAT_
PROTO_UDPLITE
/boot/config-4.8.0-36-generic:1040:warning: symbol value 'm' invalid for NF_NAT_
PROTO_SCTP
*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.
root@sowmya-Inspiron-3558:/usr/src/linux-4.10# make oldconfig
HOSTCC scripts/kconfig/conf.o
HOSTLD scripts/kconfig/conf
scripts/kconfig/conf --oldconfig Kconfig
#
configuration written to .config
#
root@sowmya-Inspiron-3558:/usr/src/linux-4.10#

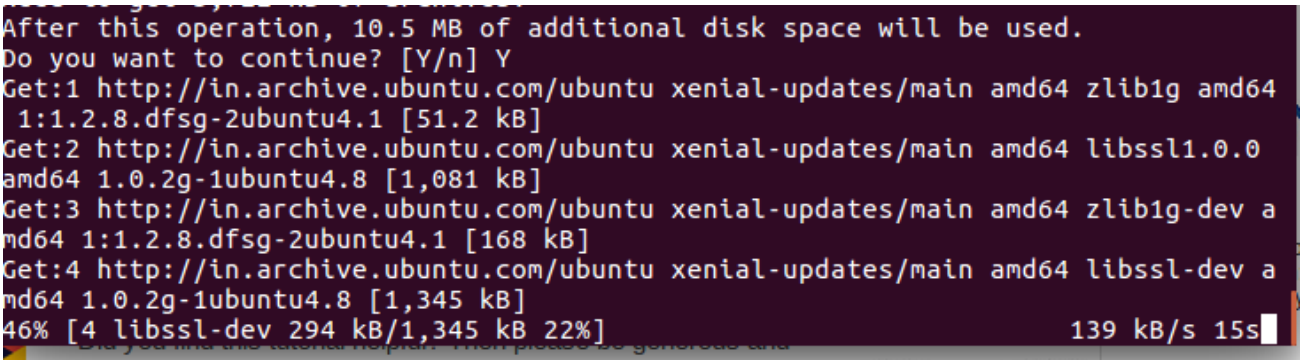

root@sowmya-Inspiron-3558: /usr/src/linux-4.10
/boot/config-4.8.0-36-generic:1020:warning: symbol value 'm' invalid for NF_CT_P
ROTO_UDPLITE
/boot/config-4.8.0-36-generic:1038:warning: symbol value 'm' invalid for NF_NAT_
PROTO_DCCP
/boot/config-4.8.0-36-generic:1039:warning: symbol value 'm' invalid for NF_NAT_
PROTO_UDPLITE
/boot/config-4.8.0-36-generic:1040:warning: symbol value 'm' invalid for NF_NAT_
PROTO_SCTP
*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.
root@sowmya-Inspiron-3558:/usr/src/linux-4.10# make oldconfig
HOSTCC scripts/kconfig/conf.o
HOSTLD scripts/kconfig/conf
scripts/kconfig/conf --oldconfig Kconfig
#
configuration written to .config
#
root@sowmya-Inspiron-3558:/usr/src/linux-4.10# make
scripts/kconfig/conf --silentoldconfig Kconfig
SYSTBL arch/x86/entry/syscalls/../../include/generated/asm/syscalls_32.h

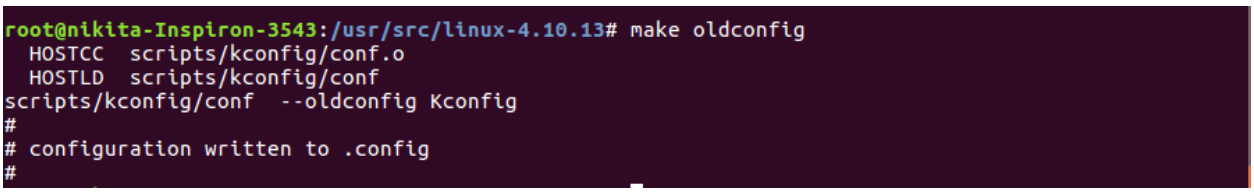


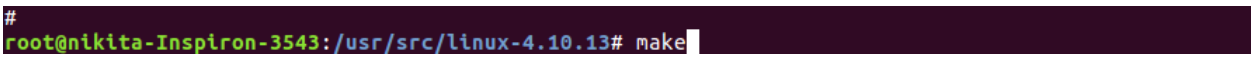
As we can see the error, which has occurred because we don't have the libssl package.

17. Sudo apt-get install libssl-dev:

Using this command we install the libssl package.

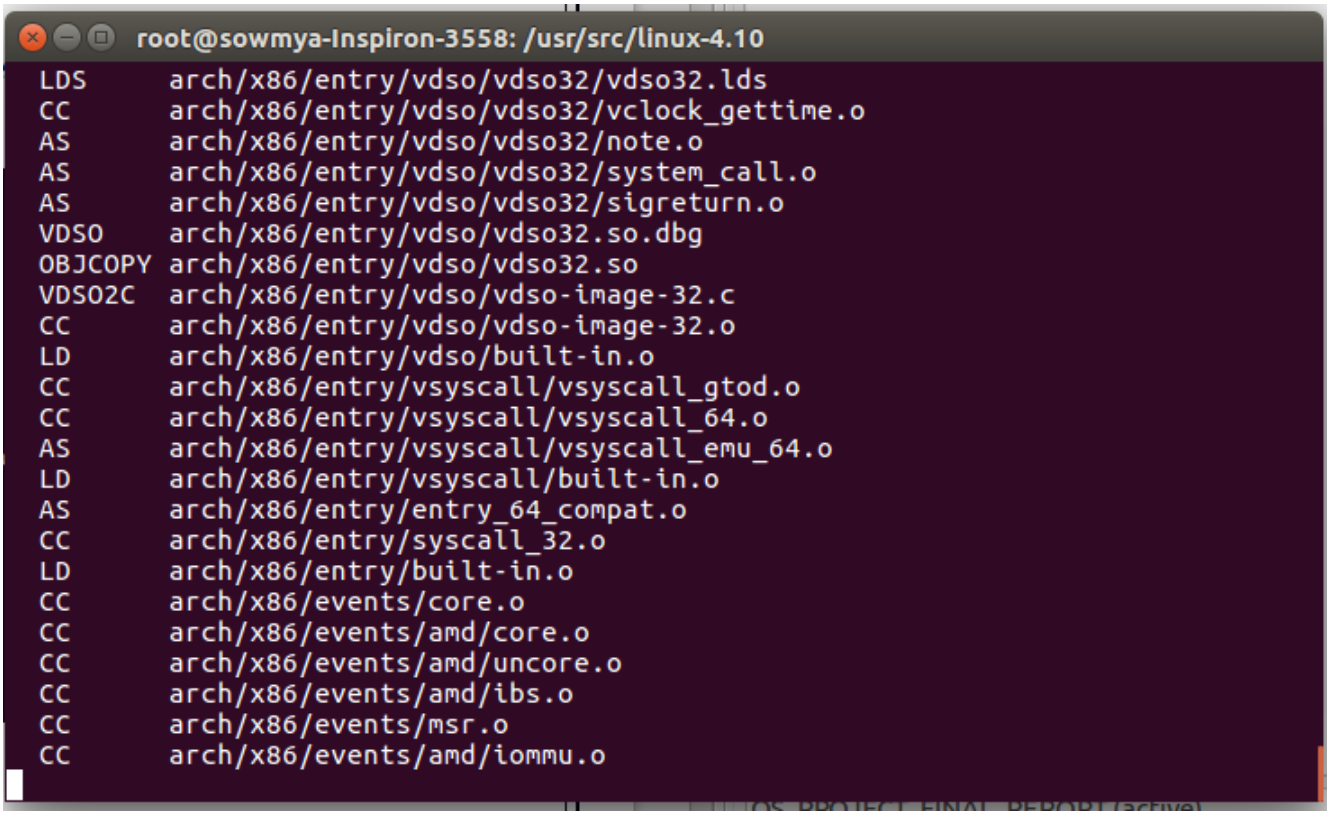




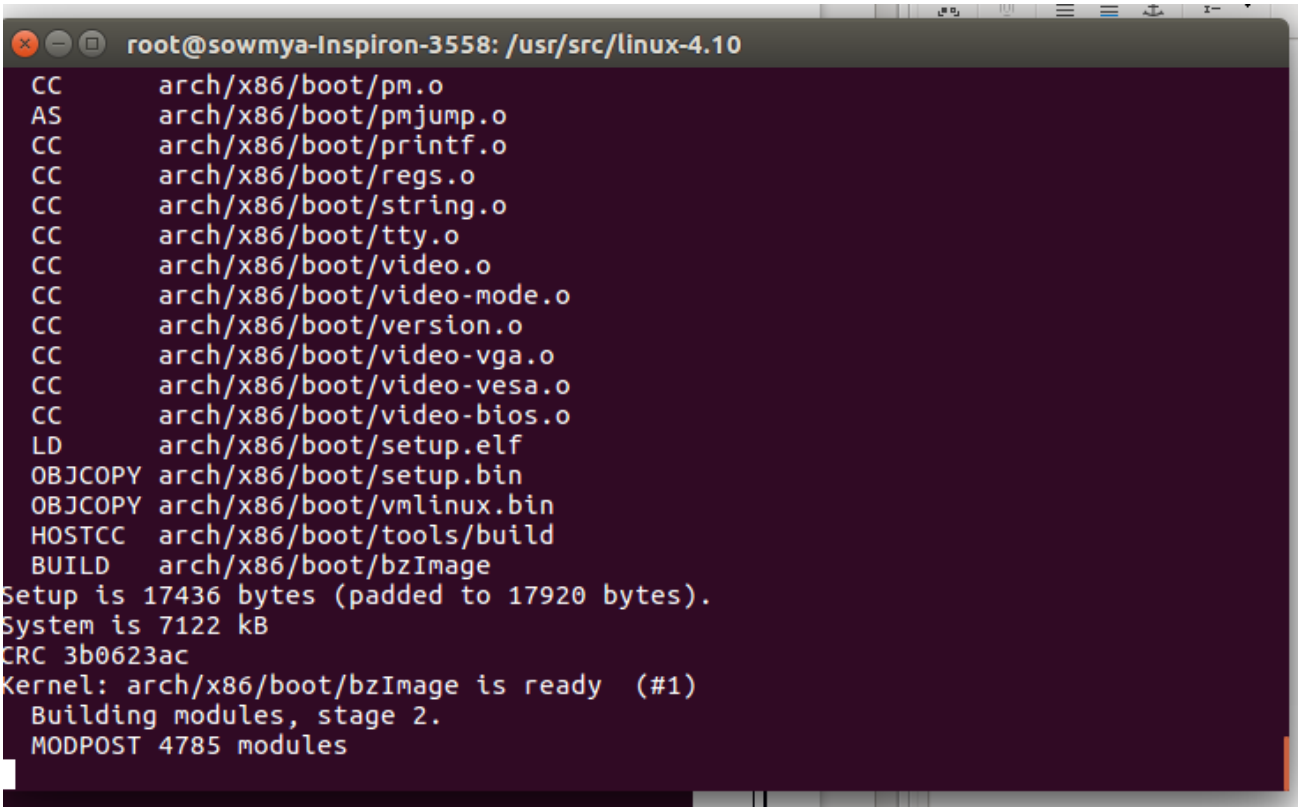


18. make oldconfig: : It reads the existing .config file and prompts the user for options in the current kernel source that are not found in the file. This is useful when taking an existing configuration and moving it to a new kernel.
19. make: utility for building and maintaining groups of programs.

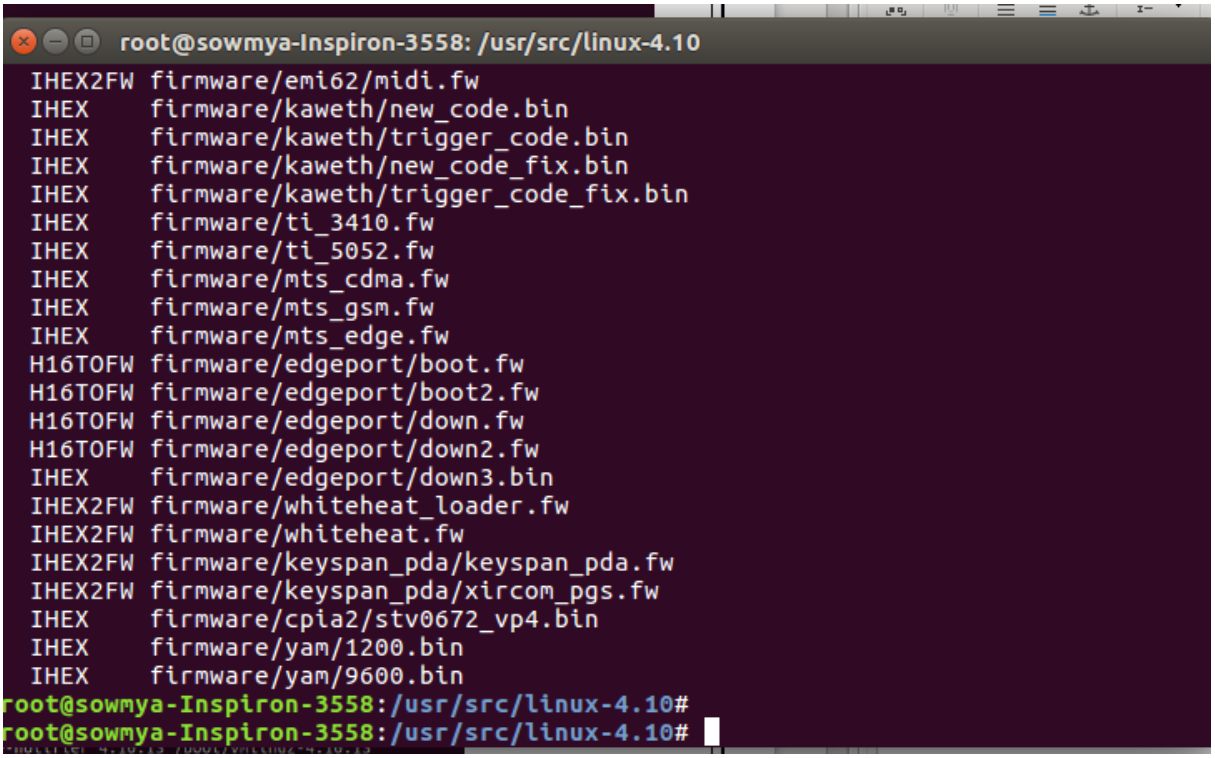
COMPILING STARTS:

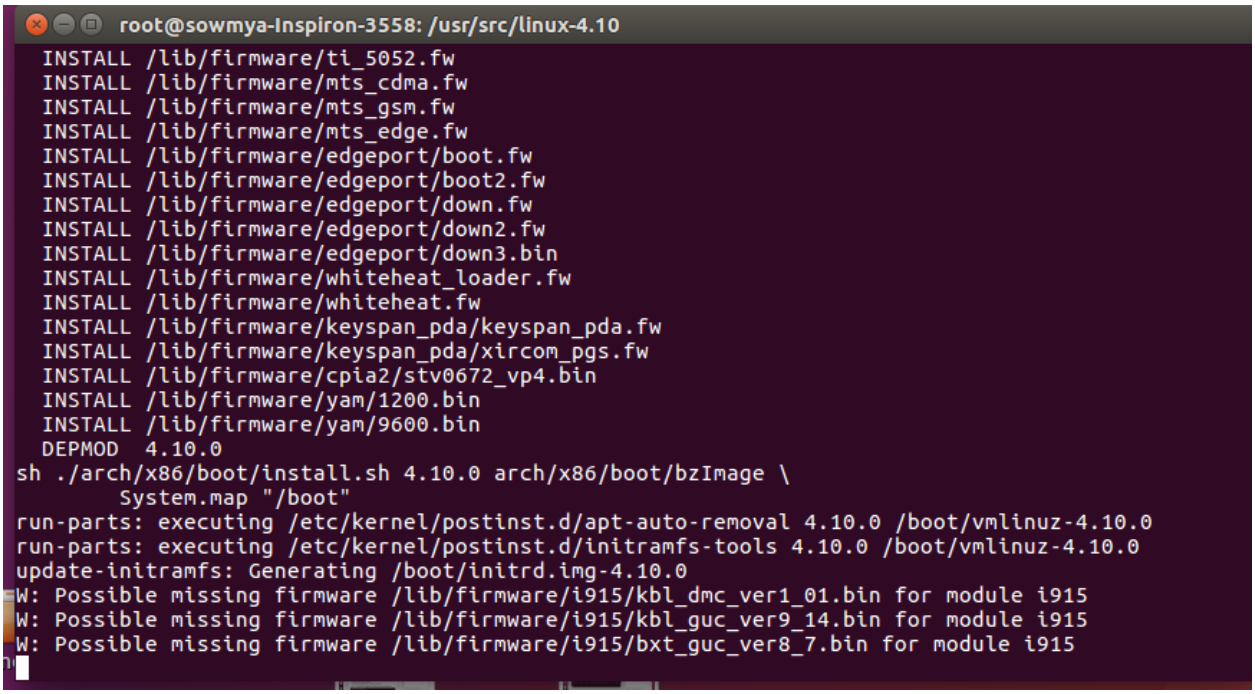


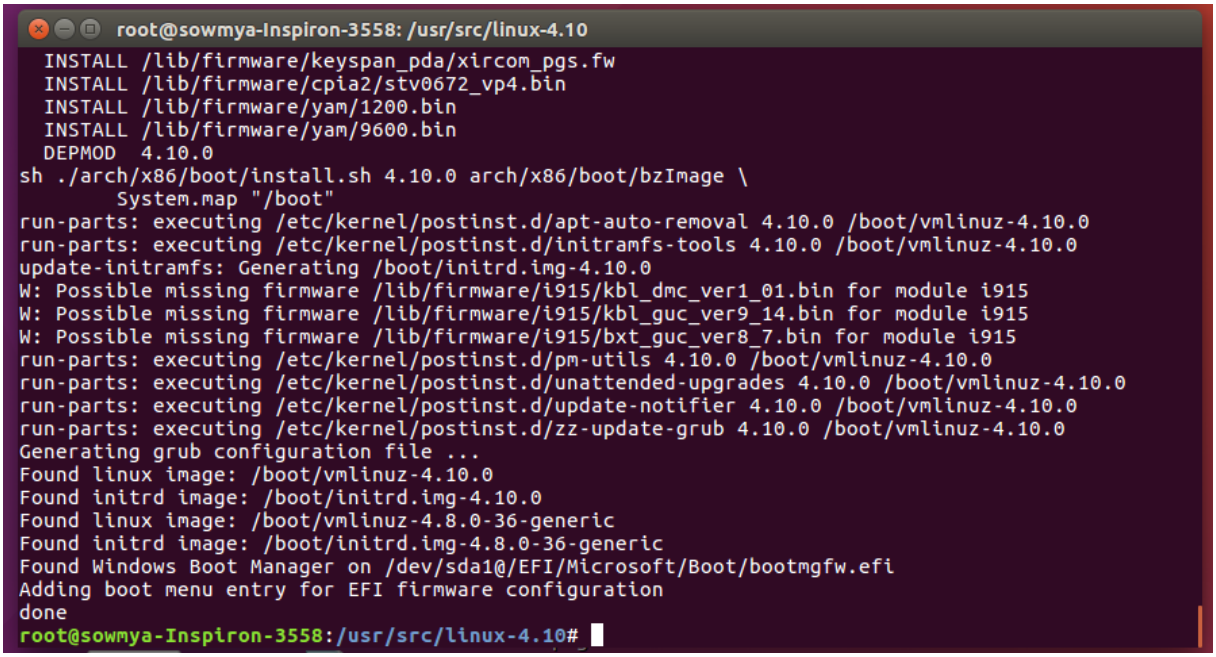
//COMPLINING


root@sowmya-Inspiron-3558: /usr/src/linux-4.10
CC arch/x86/boot/pm.o
AS arch/x86/boot/pmjump.o
CC arch/x86/boot/printf.o
CC arch/x86/boot/regs.o
CC arch/x86/boot/string.o
CC arch/x86/boot/tty.o
CC arch/x86/boot/video.o
CC arch/x86/boot/video-mode.o
CC arch/x86/boot/version.o
CC arch/x86/boot/video-vga.o
CC arch/x86/boot/video-vesa.o
CC arch/x86/boot/video-bios.o
LD arch/x86/boot/setup.elf
OBJCOPY arch/x86/boot/setup.bin
OBJCOPY arch/x86/boot/vmlinux.bin
HOSTCC arch/x86/boot/tools/build
BUILD arch/x86/boot/bzImage
Setup is 17436 bytes (padded to 17920 bytes).
System is 7122 kB
CRC 3b0623ac
Kernel: arch/x86/boot/bzImage is ready (#1)
Building modules, stage 2.
MODPOST 4785 modules

//COMPILING

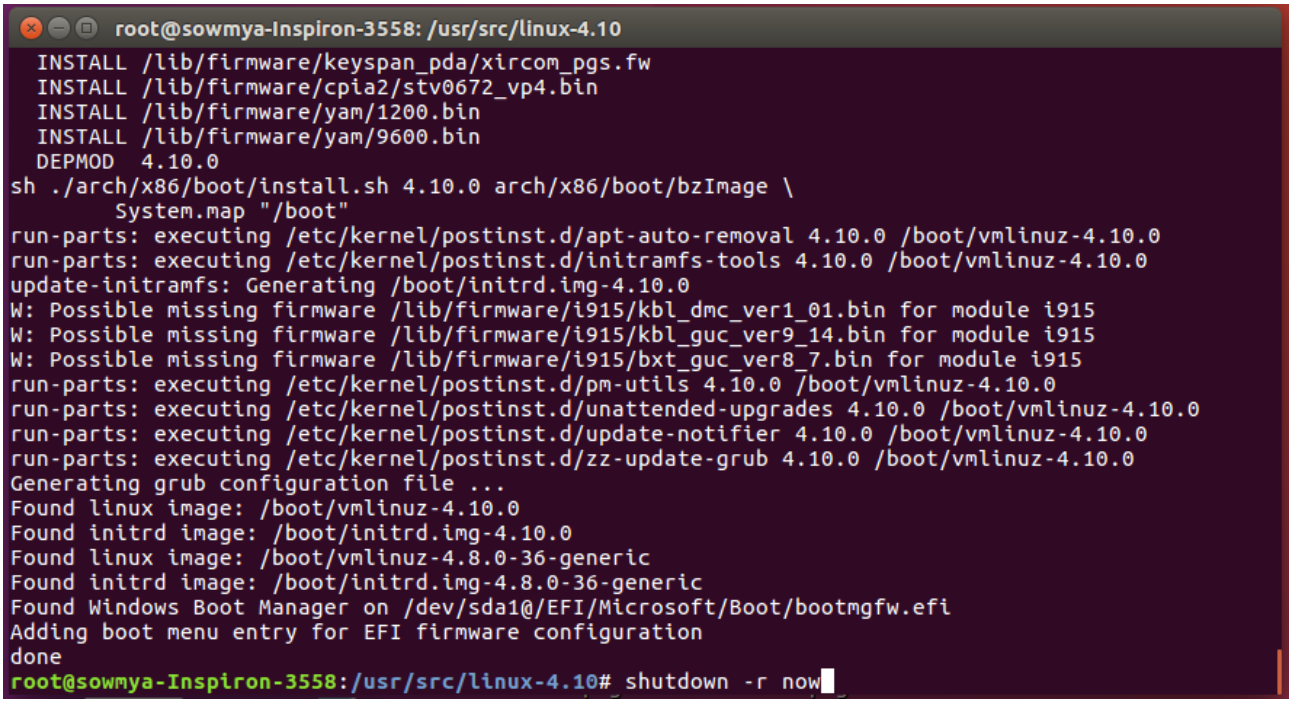

root@sowmya-Inspiron-3558: /usr/src/linux-4.10
IHEX2FW firmware/emi62/midi.fw
IHEX firmware/kaweth/new_code.bin
IHEX firmware/kaweth/trigger_code.bin
IHEX firmware/kaweth/new_code_fix.bin
IHEX firmware/kaweth/trigger_code_fix.bin
IHEX firmware/ti_3410.fw
IHEX firmware/ti_5052.fw
IHEX firmware/mts_cdma.fw
IHEX firmware/mts_gsm.fw
IHEX firmware/mts_edge.fw
H16TOFW firmware/edgeport/boot.fw
H16TOFW firmware/edgeport/boot2.fw
H16TOFW firmware/edgeport/down.fw
H16TOFW firmware/edgeport/down2.fw
IHEX firmware/edgeport/down3.bin
IHEX2FW firmware/whiteheat_loader.fw
IHEX2FW firmware/whiteheat.fw
IHEX2FW firmware/keyspan_pda/keyspan_pda.fw
IHEX2FW firmware/keyspan_pda/xircom_pgs.fw
IHEX firmware/cpia2/stv0672_vp4.bin
IHEX firmware/yam/1200.bin
IHEX firmware/yam/9600.bin
root@sowmya-Inspiron-3558:/usr/src/linux-4.10#
root@sowmya-Inspiron-3558:/usr/src/linux-4.10#





AFTER 10 HOURS OF COMPILATION

20. make modules_install install : to install the recently compiled kernel.

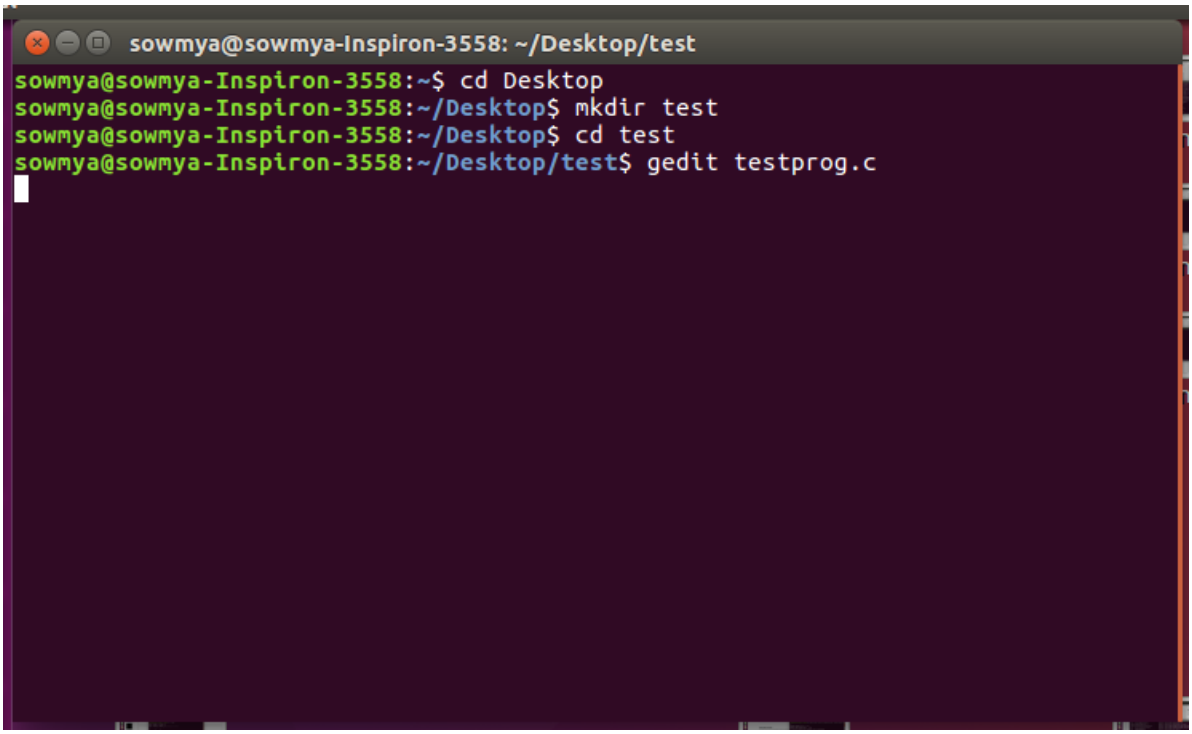


21.shutdown –r now: it will shut down the system.

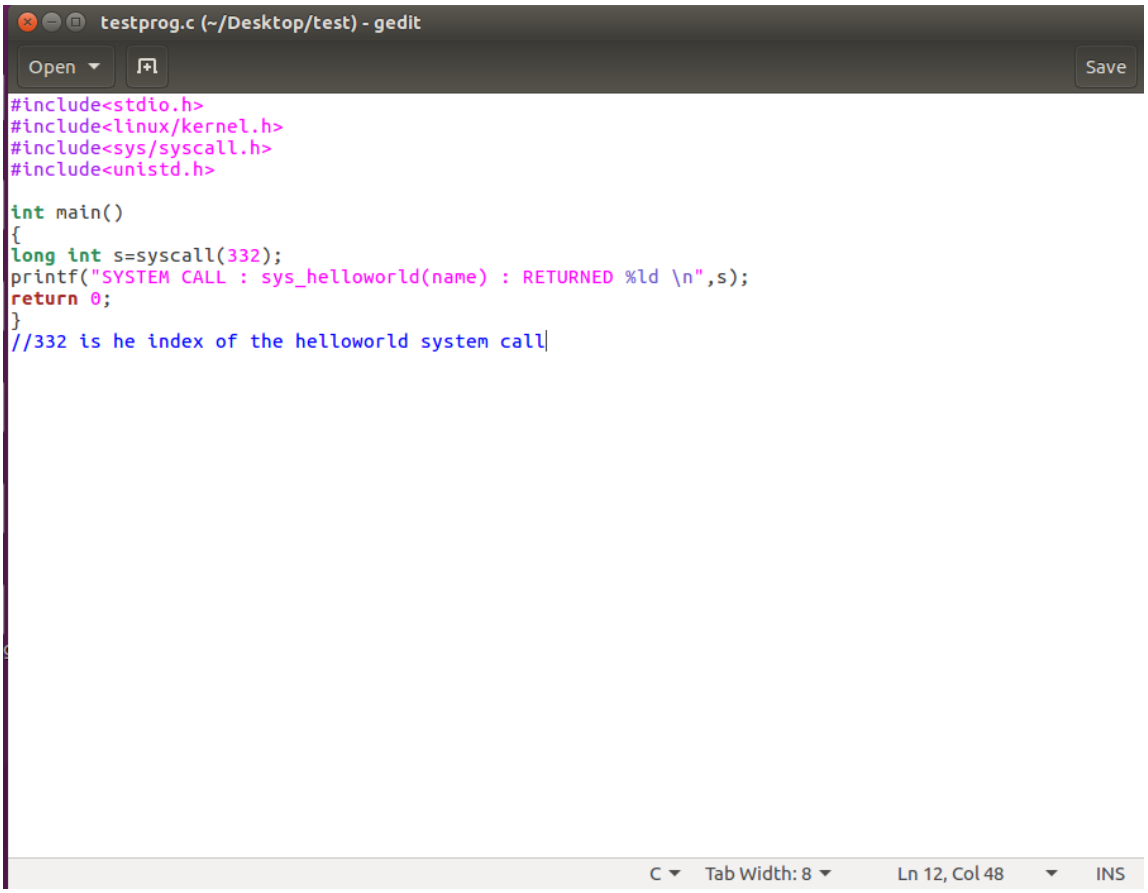
RESULT

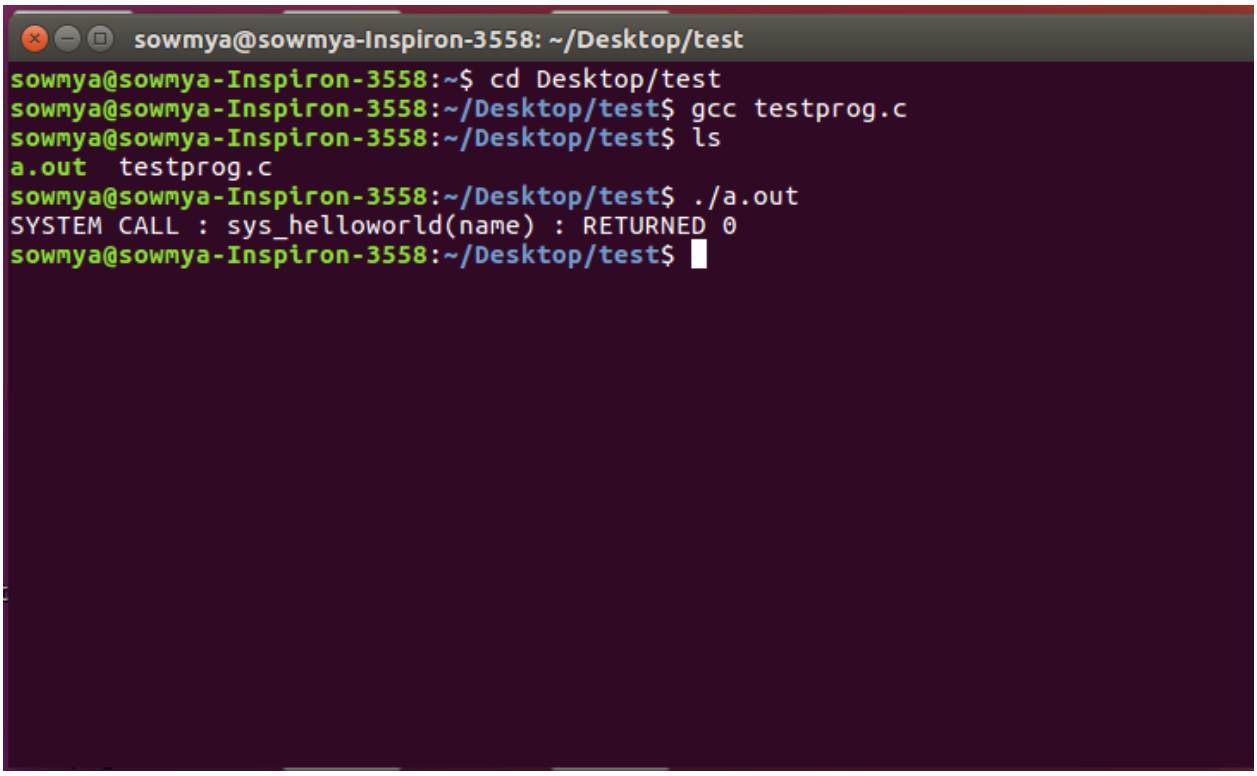
We successfully added the system calls to the UNIX Kernel.

//now that we have restarted our computer we check if the new kernel is installed

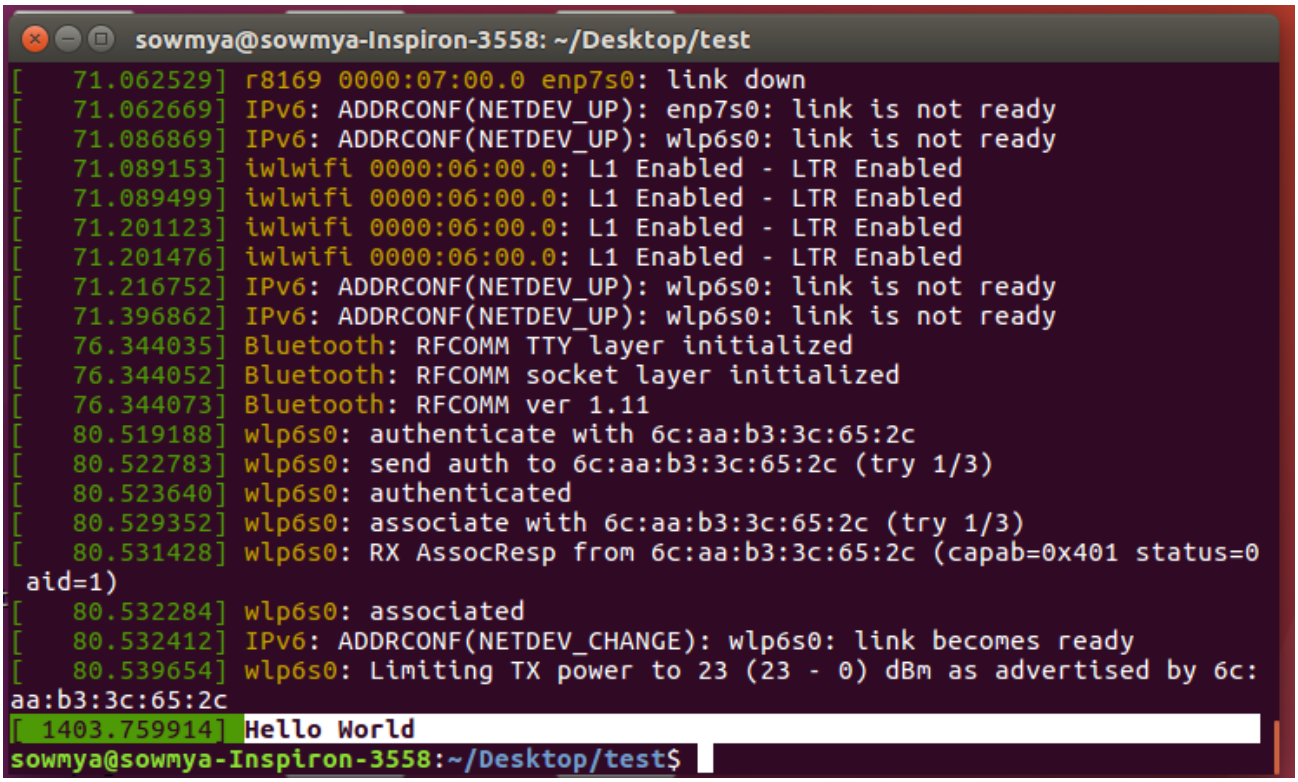


1. cd Desktop: going to the desktop.
2. mkdir test: creating a new directory named ‘test’
3. cd test: moving to this newly created directory
4. gedit testprog.c: it is a text editor opening the file’testprog’. Here we use our newly added system call.


testprog.c (~/Desktop/test) - gedit
Open
Save
#include<stdio.h>
#include<linux/kernel.h>
#include<sys/syscall.h>
#include<unistd.h>
int main()
{
long int s=syscall(332);
printf("SYSTEM CALL : sys_helloworld(name) : RETURNED %ld \n",s);
return 0;
}
//332 is he index of the helloworld system call
C
Tab Width: 8
Ln 12, Col 48
INS



5. gcc testprog.c: used for compiling the file ‘testprog’
6. ls: shows file sunder the current directory
7. ./a.out: displays output after compilation
8. dmesg: writes the kernel message to standard output.



//this is kernel log, helloworld is the message

CONCLUSION

Our new Hello World system call has been successfully added to our kernel.Here we have added a simple print command(which prints hello world, when the system call is used) which if printed returns the value 0. But we can add more complex programs and later incorporate them in our programs as system calls, hence building a better operating system.

REFERENCES

https://tssurya.wordpress.com/.../adding-a-hello-world-system-call-to-linux-kernel-3-1...

https://www.youtube.com/watch?v=AP-tBd84vbM

https://www.kernel.org/

https://en.wikipedia.org/wiki/System_call

www.studytonight.com/operating-system/system-calls