# Final-Project-OS

#Add System Call Called RABIX TO Kernel 5.8.1

First I will show the settings of my Virtual Machine :

Number of cores: 1

The capacity of memory is 1G

The kernal virsion is 5.8.1

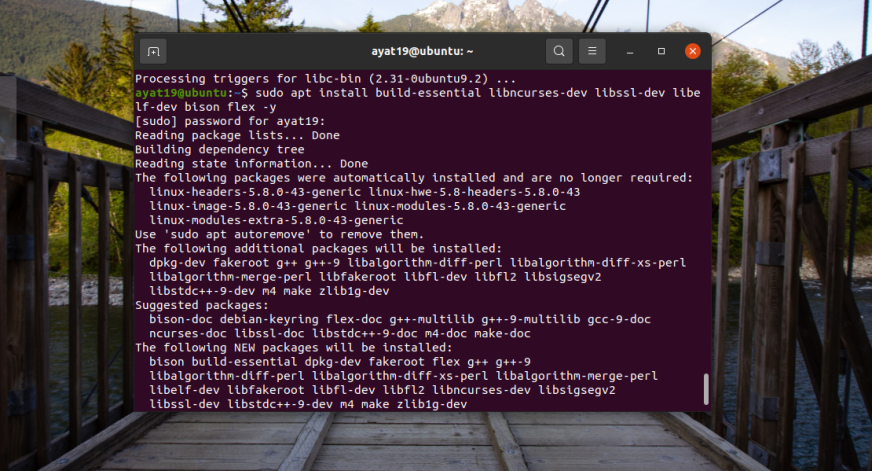
Second Steps to how to add a system call :

1- Make my Linux Ubuntu update:

sudo apt update && sudo apt upgrade –y

2- Install all packeges that i will use to compile Kernal by :

sudo apt install build-essential libncurses-dev libssl-dev libelf-dev bison flex –y



3- Clean installed packages:

sudo apt clean && sudo apt autoremove –y

4- Download the source code of the Linux kernel 5.8.1:

wget -P ~/ <https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.8.1.tar.xz>

And unpack it by using tar -xvf ~/linux-5.8.1.tar.xz -C ~/

5- Reboot My Computer

6- Change my working directory to the root directory of the recently unpacked source code

cd ~/linux-5.8.1/

7- Make a directory called RABIX and create file called RABIX.c in this file write a program

mkdir RABIX

nano RABIX/RABIX.c

#include <linux/kernel.h>

#include <linux/syscalls.h>

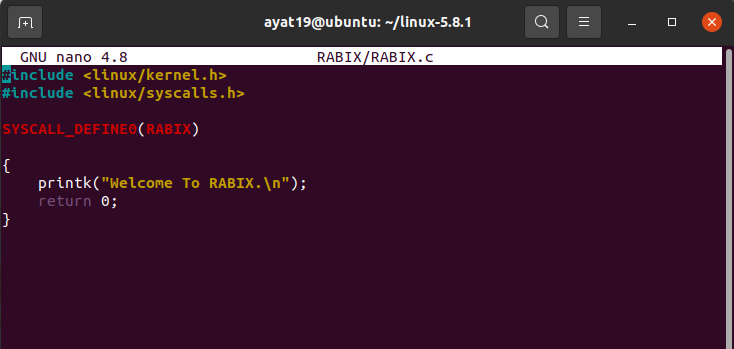
SYSCALL\_DEFINE0(RABIX)

{

printk("Welcome to RABIX.\n");

return 0;

}



8- Now i will create a makefile

nano RABIX/Makefile

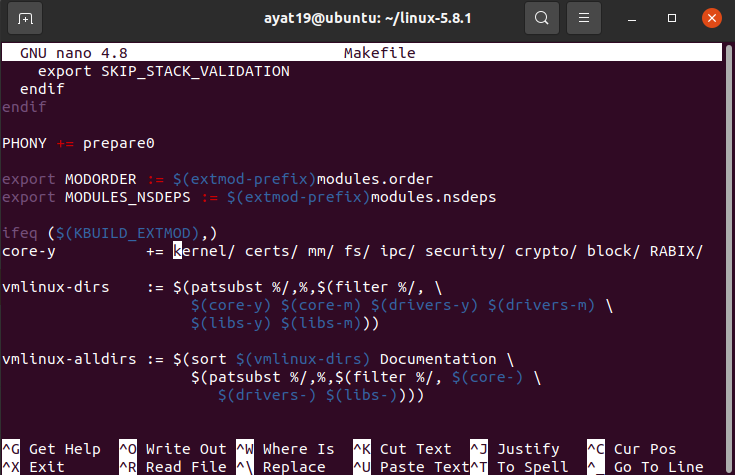
And write obj-y := RABIX.c

9- And i will open the Makefile to add the home directory to my system call to the main Makefile of the kernel.

Open the Makefile with the following command.

nano Makefile

and i will search for core-y it will apper in the second time of searching . We did the search to see this kernel/ certs/ mm/ fs/ ipc/ security/ crypto/ block/ I will add my home directory called RABIX .

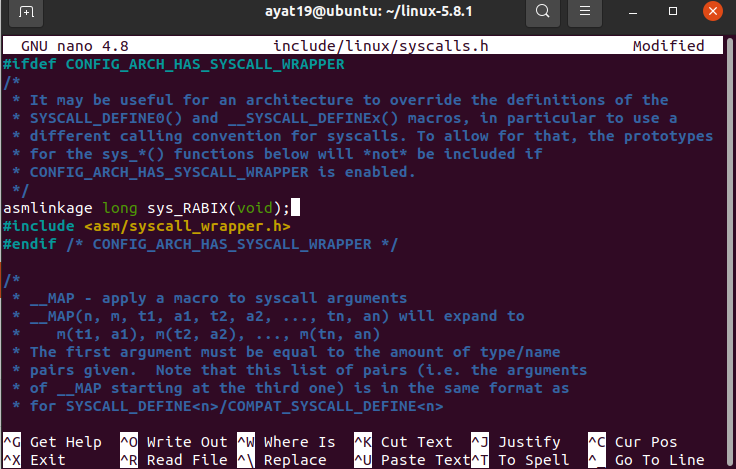


10- And I will open the header file with the following command.

nano include/linux/syscalls.h

to add a corresponding function prototype for my system call to the header file of system calls.

Search for endif and put asmlinkage long sys\_RABIX(void); above it .

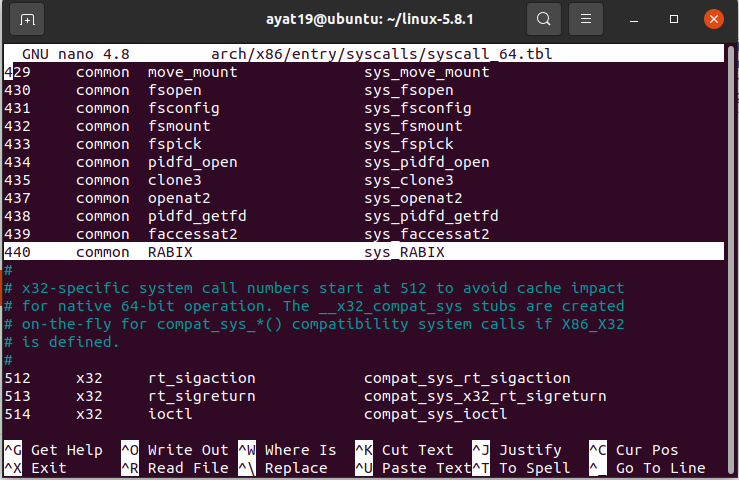


11- Add my system call to the kernel's system call table.

nano arch/x86/entry/syscalls/syscall\_64.tbl

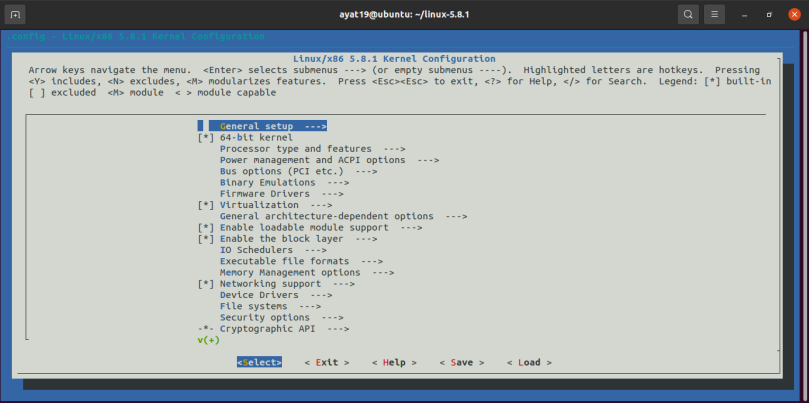
And I will navigate to the bottom of it even find a series of x32 system calls. I will put

440 common RABIX sys\_RABIX



12- Configure the kernel.

make menuconfig

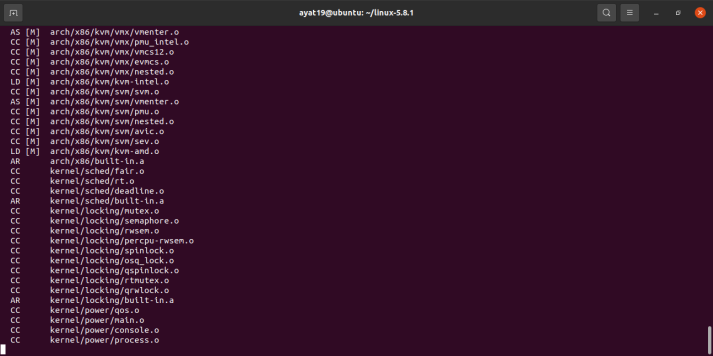


13- Find out how many logical cores you have.

nproc

14- Compile the kernel's source code.

make -j1

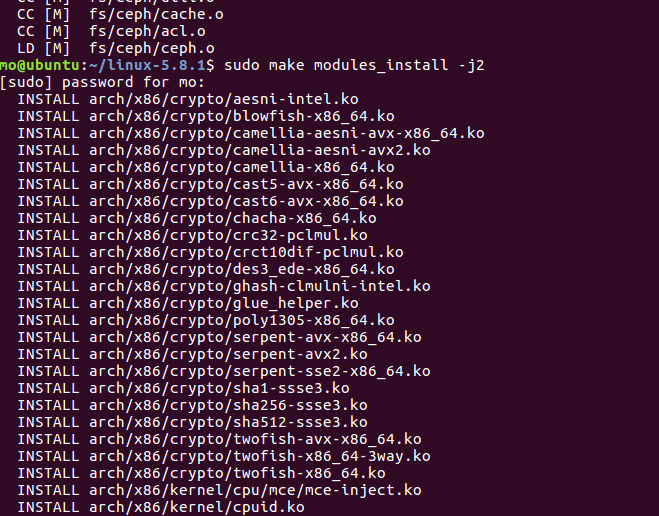


15- Prepare the installer of the kernel.

sudo make modules\_install -j1

16- Install the kernel.

sudo make install -j1



17- Update the bootloader of the operating system with the new kernel.

sudo update-grub

18- Reboot my computer.

19- I will change my working directory to my home directory.

Now i will Create a C file to generate a report of the success or failure of your system call.

using nano rabix.c and put this program :

#include <linux/kernel.h>

#include <sys/syscall.h>

#include <stdio.h>

#include <unistd.h>

#include <string.h>

#include <errno.h>

#define \_\_NR\_identity 440

long identity\_syscall(void)

{

return syscall(\_\_NR\_identity);

}

int main(int argc, char \*argv[])

{

long activity;

activity = identity\_syscall();

if(activity < 0)

{

perror("Sorry Try again .");

}

else

{

printf("Congrats, And Weclome to RABIX\n");

}

return 0;

}

20- Compile the C file just created, and run C file

**gcc -o rabix rabix.c**

**./rabix**

**Will display** Congrats, And Weclome to RABIX

21- Check the last line of the dmesg output

The print function:

Welcome to RABIX