

## Project Idea

Adding System Call the Linux OS Kernel

## Project Members

( 1 )	Ahmed Ramadan Abdallah Farag
( 2 )	Ali Mohamed Ahmed
( 3 )	Ahmed Nasser Nabil

# Adding System Call the Linux OS Kernel

## System Details

<b>CPU Cores</b>	<b>2</b>
<b>RAM Capacity</b>	<b>5</b>
<b>Kernel version</b>	<b>5.12.9</b>

## Step ( 1 ) : Preparation

**( 1.1 ) In This Step , I Fully update My operating system**

```
research@research-VB:~$ sudo apt update && sudo apt upgrade -y  
[sudo] password for research:
```

**( 1.2 ) In This Step , I Download and install the essential packages to compile kernels.**

```
research@research-VB:~$ sudo apt install build-essential libncurses-dev libssl-dev libelf-dev bison flex -y
```

## Step ( 1 ) : Preparation

**( 1.3 ) In This Step , I Clean up your installed packages.**

```
research@research-VB:~$ sudo apt clean && sudo apt autoremove -y
```

**( 1.4 ) In This Step , I Download the source code of the latest stable version of the Linux.**

```
research@research-VB:~$ wget -P ~/ https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.12.9.tar.xz
```

**( 1.5 ) In This Step , I decompress the kernel. just downloaded to your home folder.**

```
research@research-VB:~$ tar -xvf ~/linux-5.12.9.tar.xz -C ~/
```

## Step ( 2 ) : Creation

**( 2.1 ) In This Step , I Check the version of your current kernel.**

```
research@research-VB:~$ uname -r  
5.12.9
```

**( 2.2 ) I Change your working directory to the root directory of the recently unpacked source code..**

```
research@research-VB:~$ cd ~/linux-5.12.9/  
research@research-VB:~/linux-5.12.9$ mkdir research/
```

**( 2.3 ) In This Step , I Create a C file for your system call.**

```
research@research-VB:~/linux-5.12.9$ nano arch/x86/entry/syscalls/syscall_64.tbl
```

## Step ( 2 ) : Creation

( 2.4 ) In This Step , I Create a C file for my system call.

```
research@research-VB:~/linux-5.12.9$ nano research/research.c
```

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

SYSCALL_DEFINE0(research)
{
    printk("Hello Research Team.\n");
    return 0;
}
```

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

( 2.5 ) In This Step , I Create a Makefile for my system call.

```
research@research-VB:~/linux-5.12.9$ nano research/Makefile
```

## Step ( 2 ) : Creation

( 2.6 ) In This Step , I Add the home directory of your system call at the end.

```
core-y += kernel/ certs/ mm/ fs/ ipc/ security/ crypto/ block/ research/
```

( 2.7 ) In This Step, I Open the header file, and navigate to the bottom of it and write  
**asmlinkage long sys\_research (void );**

```
research@research-VB:~/linux-5.12.9$ nano include/linux/syscalls.h
```

### Step ( 3 ) : Installation

**( 3.1) In This Step, I Configure the kernel.**

```
research@research-VB: ~/linux-5.12.9$ make menuconfig
```

**( 3.2 ) In This Step, I Find out how many logical cores have.**

```
research@research-VB: ~/linux-5.12.9$ nproc  
2
```

**( 3.3 ) In This Step, I Compile the kernel's source code.**

```
research@research-VB: ~/linux-5.12.9$ make -j2
```



**( 3.4) In This Step, I Prepare the installer of the kernel.**

```
research@research-VB: ~/linux-5.12.9$ sudo make modules_install -j2
```

**( 3.5) In This Step, I Update the bootloader of the operating system with the new kernel.**

```
research@research-VB: ~/linux-5.12.9$ sudo update-grub
```

**( 3.6) In This Step, I Reboot my computer.**

```
research@research-VB: ~/linux-5.12.9$ reboot
```

## Step ( 4 ) : Result

**( 4.1) In This Step, I Check the version of your current kernel..**

```
research@research-VB:~$ uname -r  
5.12.9
```

**( 4.2) In This Step, I Change your working directory to your home directory. and I Create a C file to generate a report of the success or failure of your system call.**

```
research@research-VB:~$ cd ~  
research@research-VB:~$ nano report.c
```

## C File That contains Report

```
#include <linux/kernel.h>
#include <sys/syscall.h>
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <errno.h>

#define __NR_research 443

long research_syscall(void)
{
    return syscall(__NR_research);
}

int main(int argc, char *argv[])
{
    long activity;
    activity = research_syscall();

    if(activity < 0)
    {
        perror("Sorry, Team of research. Your system call appears to have failed.");
    }

    else
    {
        printf("Congratulations, Team of Research! Your system call is functional. Run the command dmesg in the terminal and find out!\n");
    }
}
```

**( 4.3) In This Step, I Compile the C file you just created, and Run the C file you just compiled.**

```
research@research-VB:~$ gcc -o report report.c
research@research-VB:~$ ./report
```

```
[ 570.805603] Hello Research Team.
research@research-VB:~$
```

## References

<https://dev.to/jasper/adding-a-system-call-to-the-linux-kernel-5-8-1-in-ubuntu-20-04-lts-2ga8>