l**inux-5.8.1/hello/hello.c:**

#include <linux/kernel.h>

#include <linux/syscalls.h>

SYSCALL\_DEFINE0(hello)

{

printk("Hello world\n");

return 0;

}

l**inux-5.8.1/Makefile:**

ifeq ($(KBUILD\_EXTMOD),)

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

l**inux-5.8.1/include/linux$ gedit syscalls.h**

asmlinkage long sys\_hello(void);

**Location: linux-5.8.1/arch/x86/entry/syscalls$ gedit syscall\_64.tbl**

440 common hello sys\_hello

**home/report.c**

#include <linux/kernel.h>

#include <sys/syscall.h>

#include <stdio.h>

#include <unistd.h>

#include <string.h>

#include <errno.h>

#define \_\_NR\_identity 548

long identity\_syscall(void)

{

return syscall(\_\_NR\_identity);

}

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

{

long activity;

activity = identity\_syscall();

if(activity < 0)

{

perror("Sorry, IMRAN. Your system call appears to have failed.");

}

else

{

printf("Congratulations, IMRAN! Your system call is functional. Run the command dmesg in the terminal and find out!\n");

}

return 0;

}

**Updated System Call:**

#include <linux/kernel.h>

#include <linux/syscalls.h>

#include <linux/sched.h>

SYSCALL\_DEFINE0(hello)

{

printk(KERN\_INFO "The process is \"%s\" (pid %i)\n",current->comm, current->pid);

struct task\_struct \*task = current; // getting global current pointer

printk(KERN\_NOTICE "ptr:%p", task);

printk(KERN\_NOTICE"pcount: %lu",task->sched\_info.pcount);

printk(KERN\_NOTICE"csw1: %lu",task->nvcsw);

printk(KERN\_NOTICE"csw2: %lu",task->nivcsw);

printk(KERN\_NOTICE"delay\_waiting: %llu",task->sched\_info.run\_delay);

return 0;

}

**Updated home/report.c:**

#include <linux/kernel.h>

#include <sys/syscall.h>

#include <stdio.h>

#include <unistd.h>

#include <string.h>

#include <errno.h>

#define \_\_NR\_identity 548

long identity\_syscall(void)

{

return syscall(\_\_NR\_identity);

}

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

{

int a;

for(a=0;a<1000000;a++)

{

printf("kernel mode");

}

long activity;

activity = identity\_syscall();

if(activity < 0)

{

perror("Sorry, IMRAN. Your system call appears to have failed.");

}

else

{

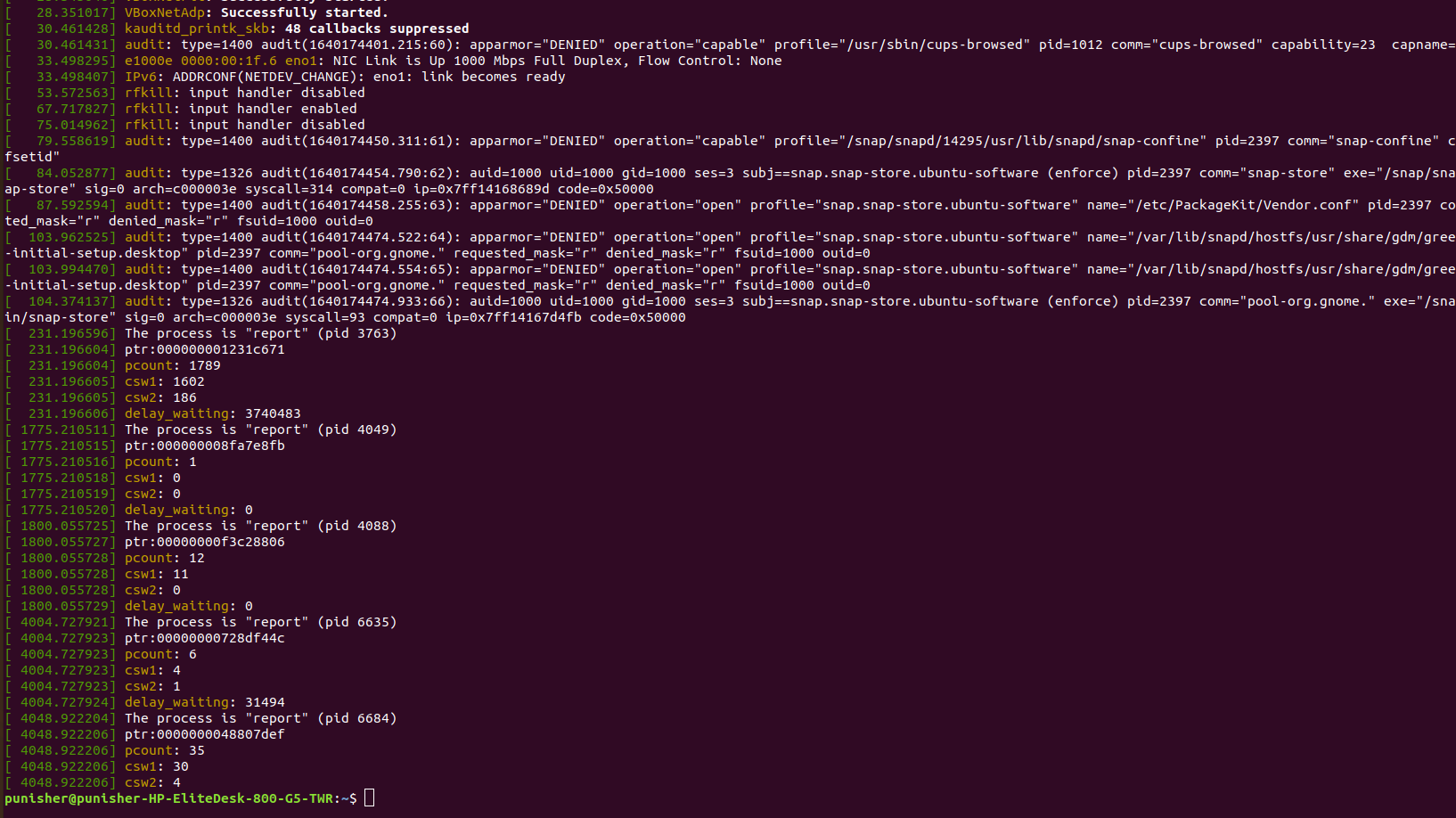
printf("%ld\n",activity);

printf("Congratulations,! Your system call is functional. Run the command dmesg in the terminal and find out!\n");

}

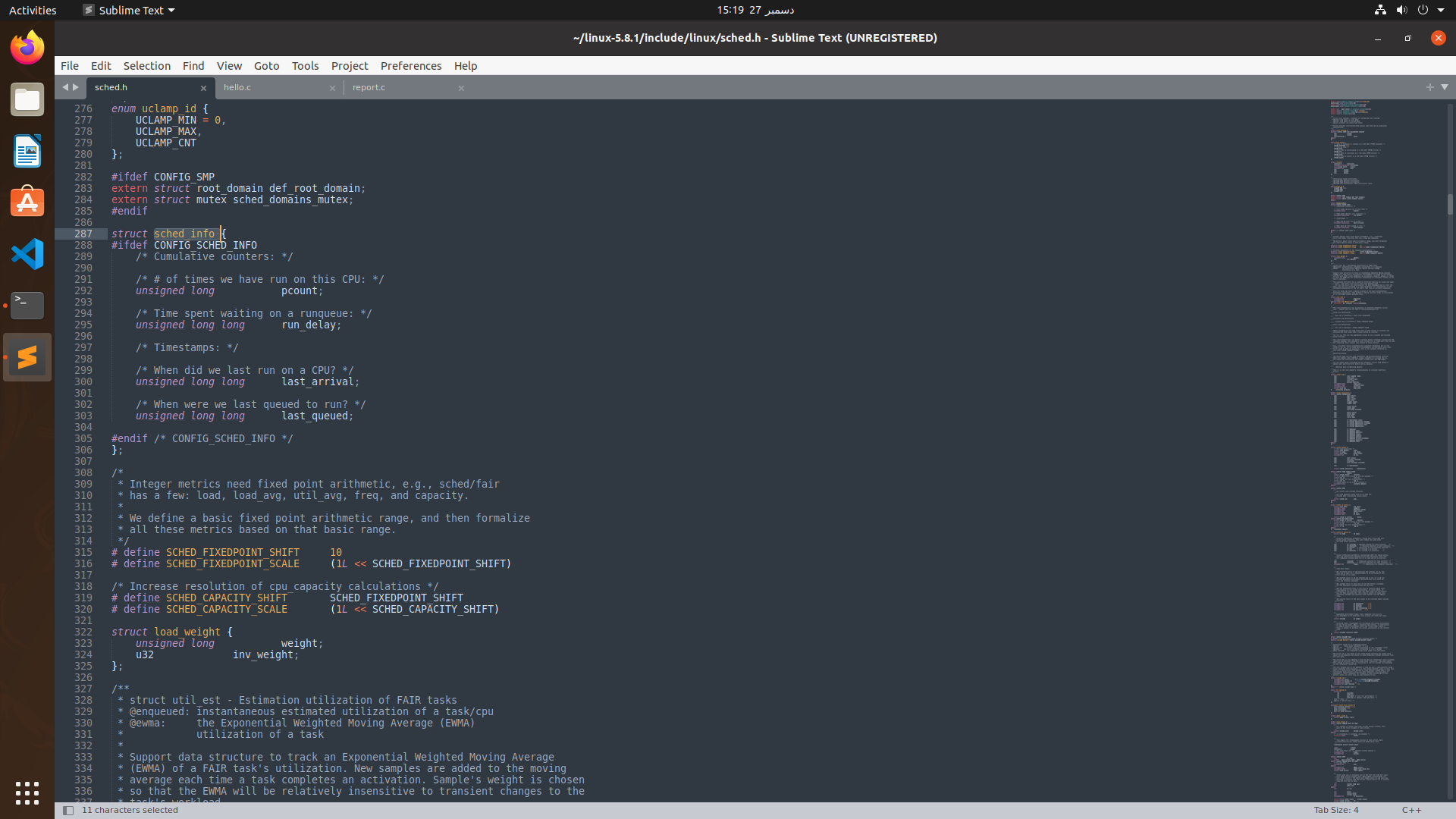
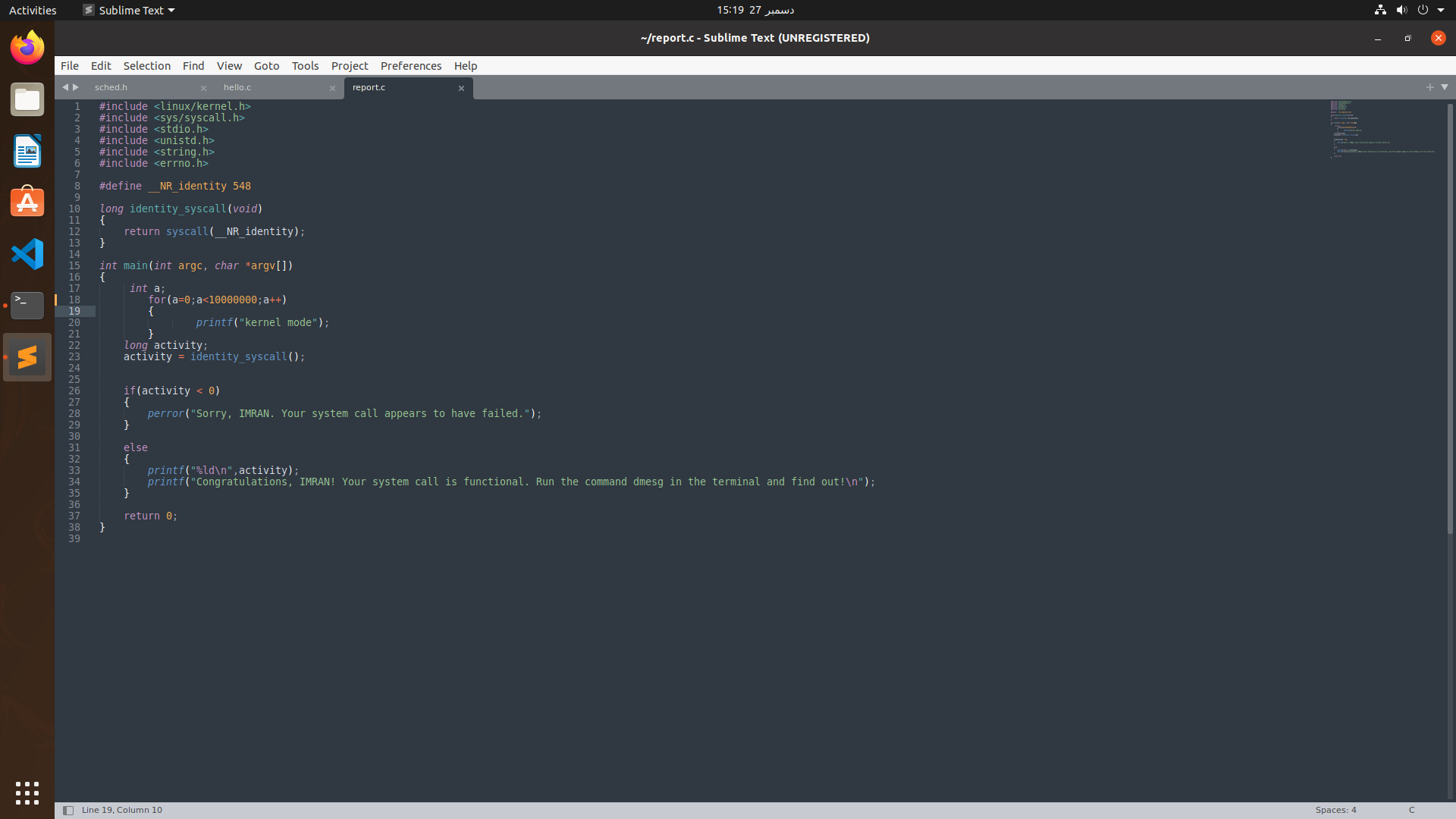
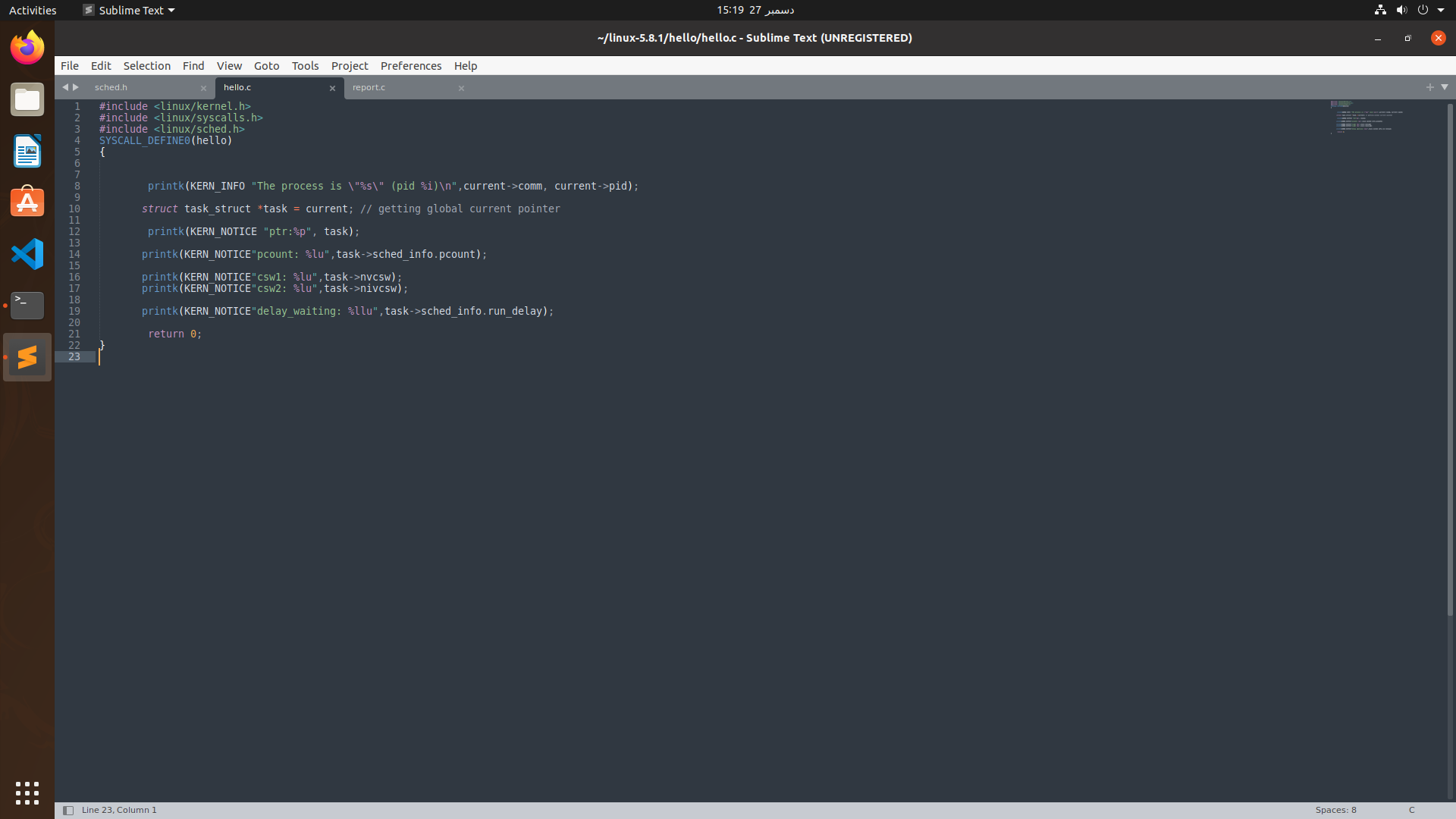
return 0;

}



**Reference:**

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

****