Assignment 2.2

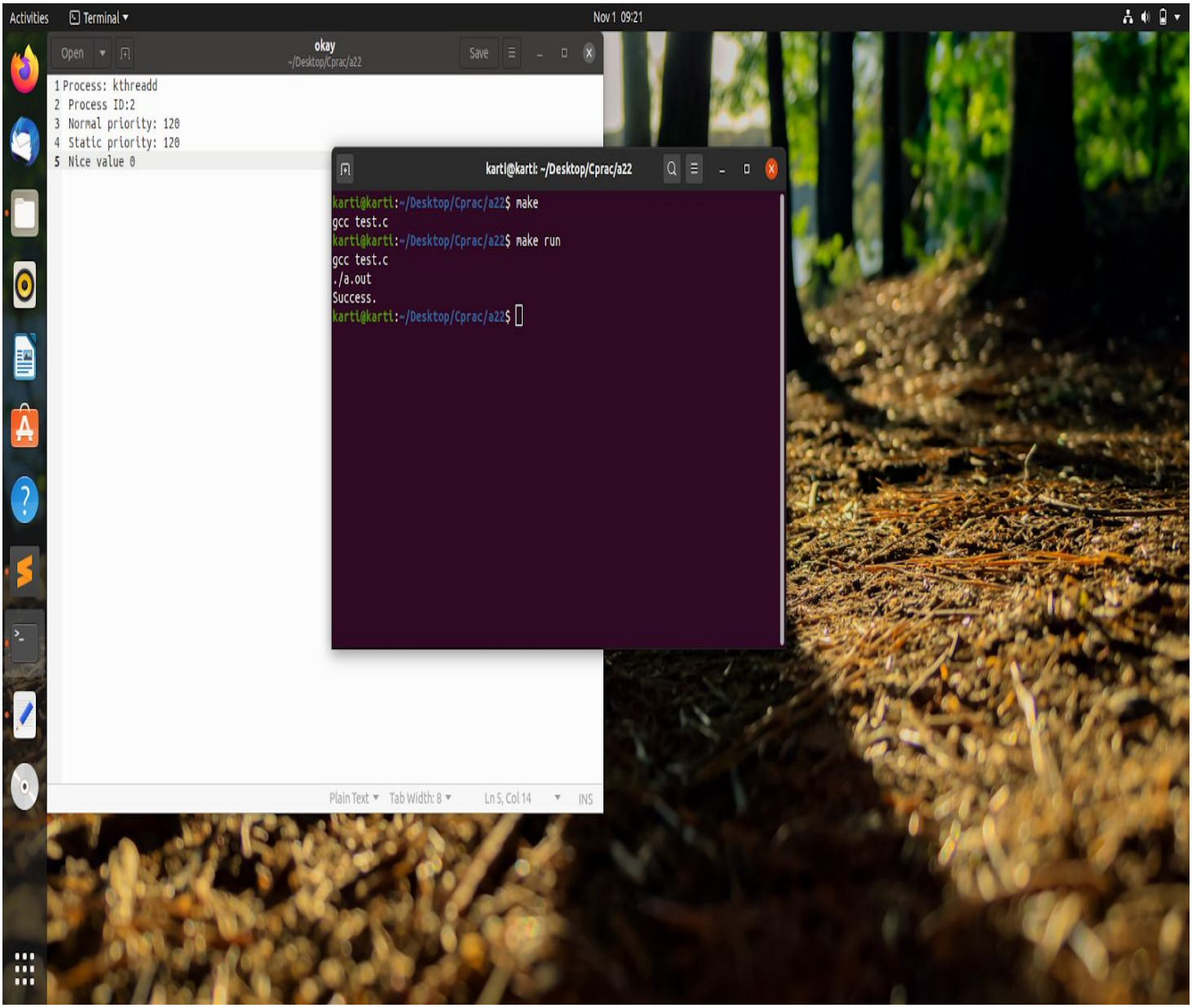
Name: Kartikey Gupta
Roll no.=2019427

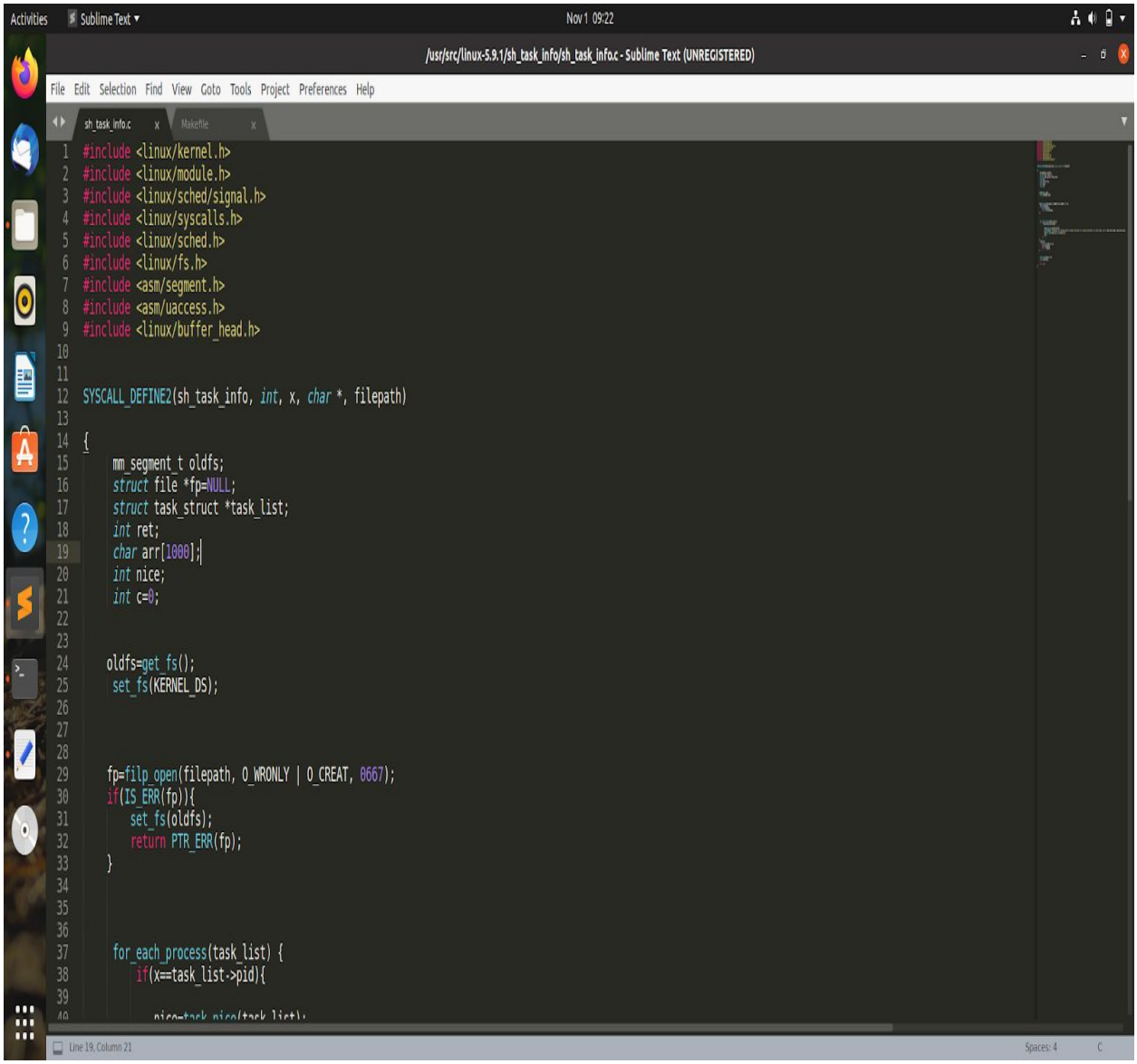
//Implementation
Two arguments are taken from user-pid and the filename.
Pid is the process ID while filename is the name of the file where we want the output to be written.
A task_struct is defined in our code to find the necessary values related to a process.
A char array is also defined to contain the output which is to be written.
To write onto a file, we used various functions related to opening and closing a file and writing onto the file.
We put the details of the process onto the char array that we defined and then we write this char array onto the file.
Errors are handled in the system call implementation through return values.
A different return value(other than 0) will be returned to the user if the system call fails.
Now adding this system call to the linux kernel requires us to change the makefile of linux-5.9.1.
We also make a makefile in the sh_task_info folder of linux that we just created and add “obj-y := identity.o”.
Now we add an entry in include/linux/syscalls.h.
Next we also add an entry in the syscall entry table, that is, nano arch/x86/entry/syscalls/syscall_64.tbl.
Next we save the default settings in the menuconfig and compile the kernel using sudo make.
Last step, we install the modules using make modules_install install.
Lastly, we reboot to install our kernel.

//Testing
System call syntax: syscall(440, pid, filename);
The first argument is the system call number here. We declared the syscall at 440th number therefore we passed that.
Second argument is the pid number and third is the filename.
Now, if the syscall is a success, the console shows an appropriate message and if it is a failure the console shows that too.

When the system call succeeds the output will be written to a file of our choice.
Sample input: syscall(440, 2, "file.txt");
We can use dmesg to see the printed output on kernel log.

//Error handling:
File related errors such as the filename is actually a directory.
Invalid PID provided as input error is also handled.




Activities
Sublime Text
Nov 1 09:22
/usr/src/linux-5.9.1/sh_task_info/sh_task_info.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
sh_task_info.c
Makefile
#include <linux/kernel.h>
#include <linux/module.h>
#include <linux/sched/signal.h>
#include <linux/syscalls.h>
#include <linux/sched.h>
#include <linux/fs.h>
#include <asm/segment.h>
#include <asm/uaccess.h>
#include <linux/buffer_head.h>
SYSCALL_DEFINE2(sh_task_info, int, x, char *, filepath)
{
mm_segment_t oldfs;
struct file *fp=NULL;
struct task_struct *task_list;
int ret;
char arr[1000];
int nice;
int c=0;
oldfs=get_fs();
set_fs(KERNEL_DS);
fp=filp_open(filepath, O_WRONLY | O_CREAT, 0667);
if(IS_ERR(fp)){
set_fs(oldfs);
return PTR_ERR(fp);
}
for_each_process(task_list) {
if(x==task_list->pid){
Line 19, Column 21
Spaces: 4
C