# Assignment 2.1

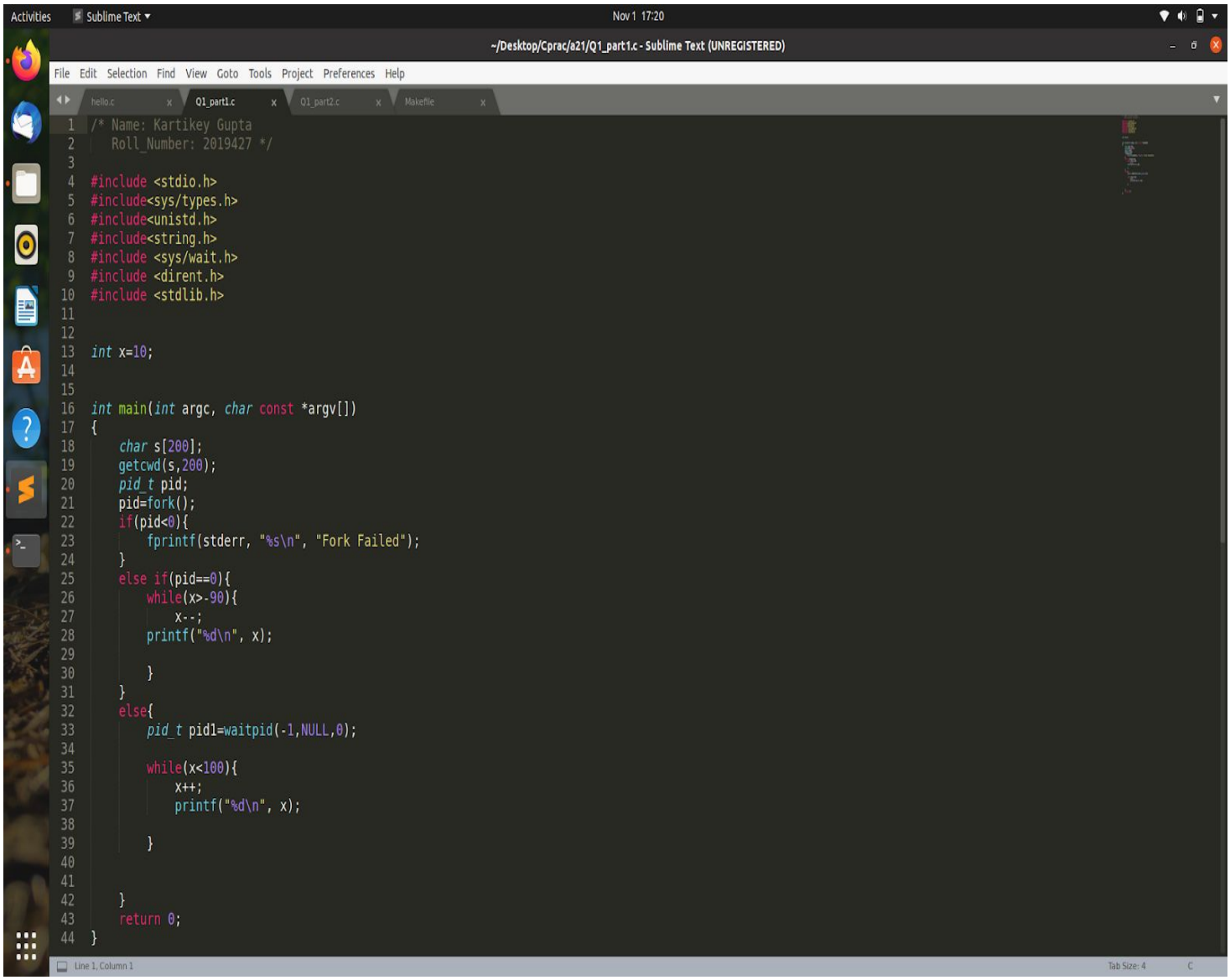
Name: Kartikey Gupta
Roll no.=2019427

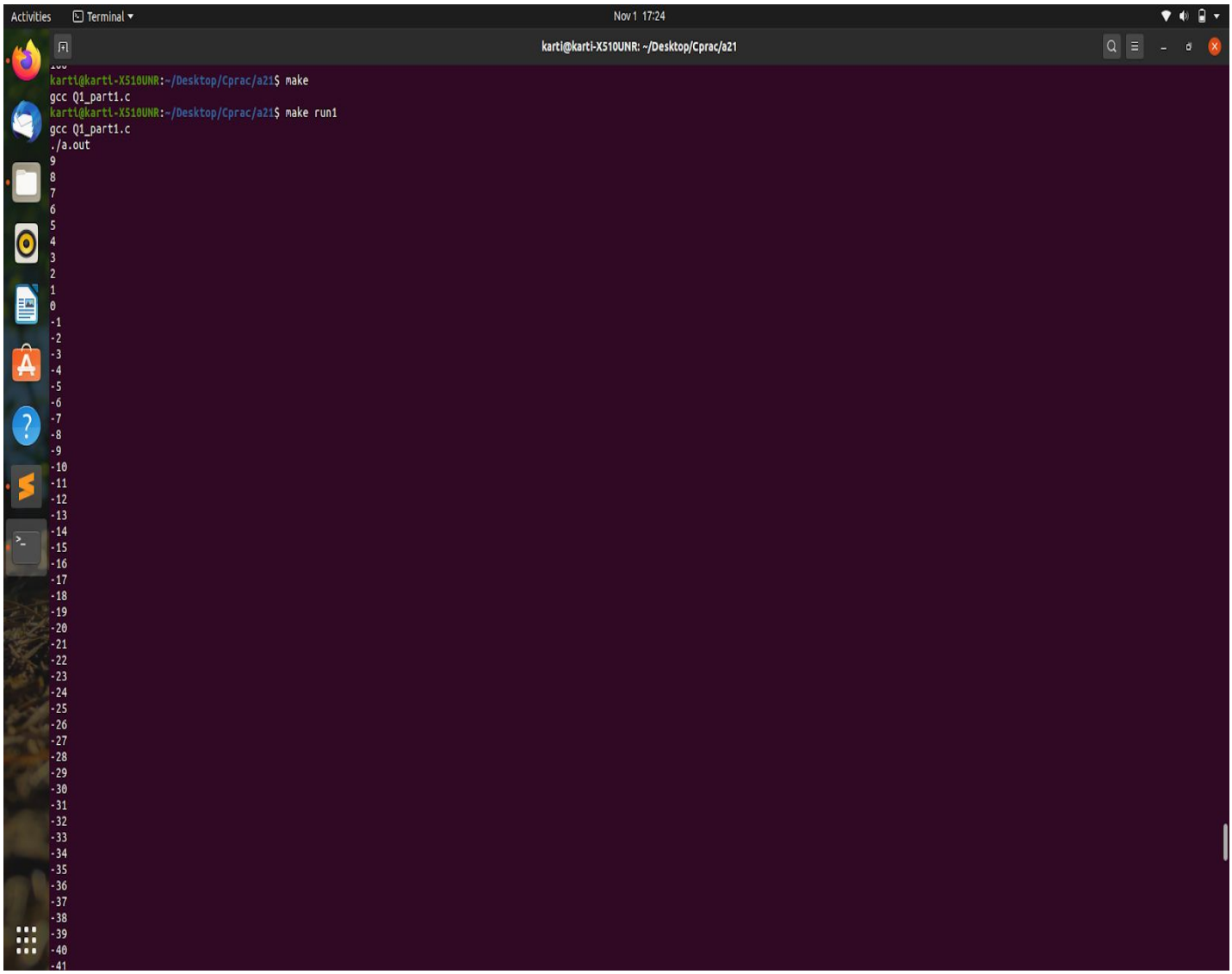
For the first part of this assignment, we implement the instruction through a fork system call.
Fork is implemented via the clone system call. The clone system call coupled with some flags makes implementation of fork possible.
Fork duplicates the memory space of the parent and gives it to the child process.
Now there are two individual processes running concurrently.
This means there are now two global variables one in the child process and one in the parent process.
Now, the child process keeps on decreasing its own global variable until it hits -90.
The parent process after waiting for the child to finish(since we used wait()) start increasing its own global variable from 10 until it hits 100.

For the second part of this assignment, we implement the given instructions through a pthread_create() system call to create threads.
Pthread_create does not duplicate the memory space which causes the global variable which is to be accessed by both the processes to remain the same.
This causes the changes in global variable by one process to be reflected in the other.
Therefore, we run the threads sequentially to first decrease the value of global variable to -90 and then after increasing the value of this global variable to 100 by parent process.

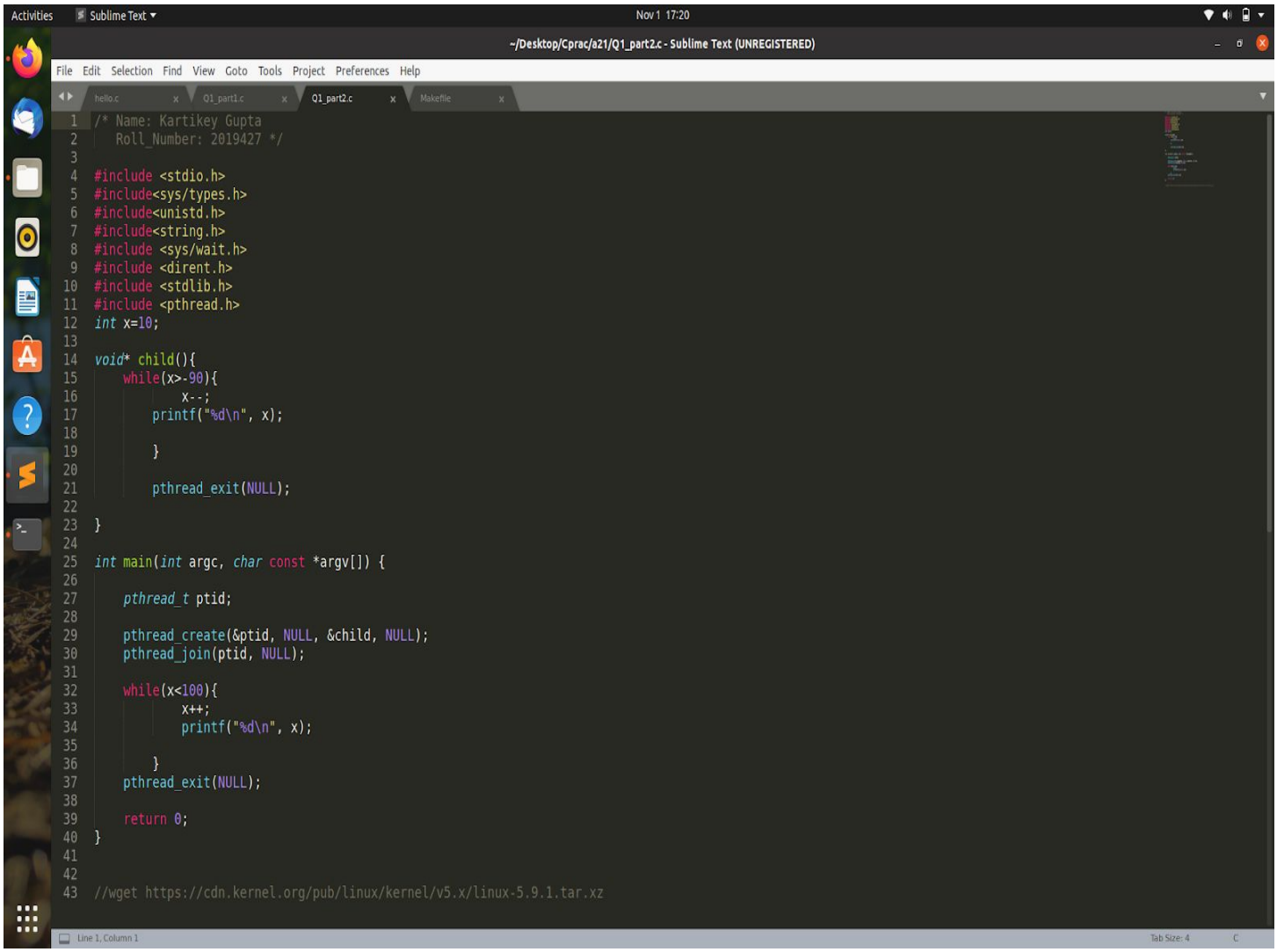
//Differences:
Child decreases the global variable from 10 to -90 in the first case. The parent then increase global variable from 10 to 100.
In second case, child decreases the global variable from 10 to -90. The parent then increase the global variable from -90 to 100.

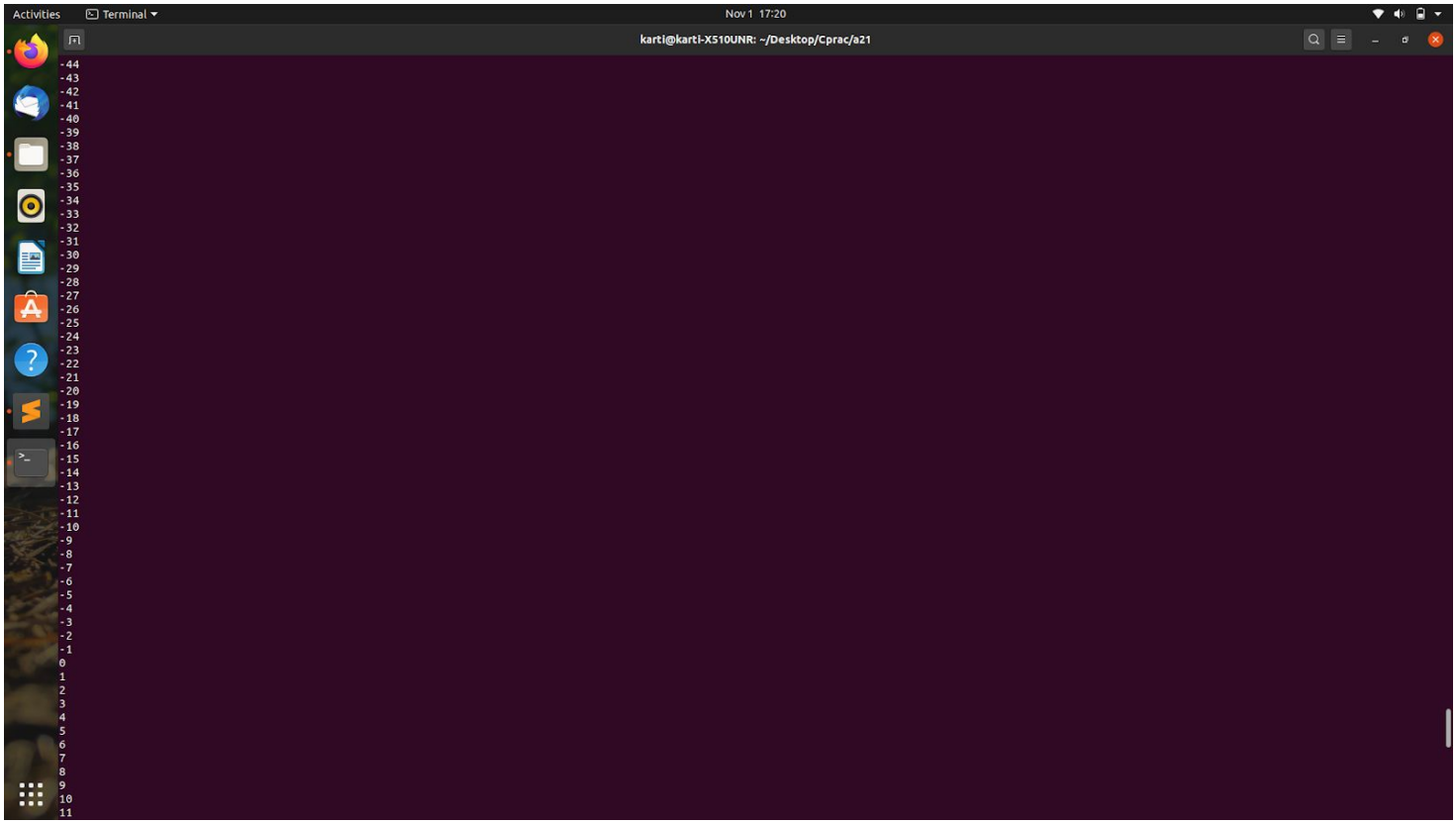
Q1_part1:


Activities
Sublime Text
Nov 1 17:20
~/Desktop/Cprac/a21/Q1_part1.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
hello.c
Q1_part1.c
Q1_part2.c
Makefile
/* Name: Kartikey Gupta
Roll_Number: 2019427 */
#include <stdio.h>
#include<sys/types.h>
#include<unistd.h>
#include<string.h>
#include <sys/wait.h>
#include <dirent.h>
#include <stdlib.h>
int x=10;
int main(int argc, char const *argv[])
{
char s[200];
getcwd(s,200);
pid_t pid;
pid=fork();
if(pid<0){
fprintf(stderr, "%s\n", "Fork Failed");
}
else if(pid==0){
while(x>-90){
x--;
printf("%d\n", x);
}
}
else{
pid_t pid1=waitpid(-1,NULL,0);
while(x<100){
x++;
printf("%d\n", x);
}
}
return 0;
}
Line 1, Column 1
Tab Size: 4
C


Activities
Terminal
Nov 1 17:24
karti@karti-X510UNR: ~/Desktop/Cprac/a21
karti@karti-X510UNR:~/Desktop/Cprac/a21$ make
gcc Q1_part1.c
karti@karti-X510UNR:~/Desktop/Cprac/a21$ make run1
gcc Q1_part1.c
./a.out
9
8
7
6
5
4
3
2
1
0
-1
-2
-3
-4
-5
-6
-7
-8
-9
-10
-11
-12
-13
-14
-15
-16
-17
-18
-19
-20
-21
-22
-23
-24
-25
-26
-27
-28
-29
-30
-31
-32
-33
-34
-35
-36
-37
-38
-39
-40
-41

Q2_part2:


Activities
Sublime Text
Nov 1 17:20
~/Desktop/Cprac/a21/Q1_part2.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
hello.c
Q1_part1.c
Q1_part2.c
Makefile
/* Name: Kartikey Gupta
Roll_Number: 2019427 */
#include <stdio.h>
#include<sys/types.h>
#include<unistd.h>
#include<string.h>
#include <sys/wait.h>
#include <dirent.h>
#include <stdlib.h>
#include <pthread.h>
int x=10;
void* child(){
while(x>-90){
x--;
printf("%d\n", x);
}
pthread_exit(NULL);
}
int main(int argc, char const *argv[]) {
pthread_t ptid;
pthread_create(&ptid, NULL, &child, NULL);
pthread_join(ptid, NULL);
while(x<100){
x++;
printf("%d\n", x);
}
pthread_exit(NULL);
return 0;
}
//wget https://cdn.kernel.org/pub/linux/kernel/v5.x/linux-5.9.1.tar.xz
Line 1, Column 1
Tab Size: 4
C


Activities
Terminal
Nov 1 17:20
karti@karti-X510UNR: ~/Desktop/Cprac/a21
-44
-43
-42
-41
-40
-39
-38
-37
-36
-35
-34
-33
-32
-31
-30
-29
-28
-27
-26
-25
-24
-23
-22
-21
-20
-19
-18
-17
-16
-15
-14
-13
-12
-11
-10
-9
-8
-7
-6
-5
-4
-3
-2
-1
0
1
2
3
4
5
6
7
8
9
10
11