ASSIGNMENT -3

UPDATE THE KERNEL

1>Firstly we compiled linux- 5.9.1.

2>This was done by first downloading the tar file from the linux kernel website.

3>Then untar the file and do make

4>Then make modules

5>Then make modules_install

6>Then update grub

7>And lastly doing the reboot

We now have linux -5.9.1 64 bits installed in our system.

In the assignment we have implemented a system to add a soft real-time guarantee so that the process which is being processed by the system. Each process will receive atleast t units of time slice that is being provided by the function rtnice that is implemented by us.

This ensures that the priority is high for the process that has a small rtnice value will be taken into consideration first by the operating system than the rest of the processes.

In general the system prioritizes on the value of vruntime but here it will prioritize according to the rtnice values .

SYSCALL LOGICAL AND TECHNICAL IMPLEMENTATION

We have to update the linux-5.9.1/kernel/sched/fair.c to make changes in entity_before() function a code is added to compare the s_runtimes of the two input sched_entities. The comparisons are done according to vruntime in case both of them are 0 and hence are allotted no s_runtime values from us.

In update_curr() function if we have any sched_entity has a s_runtime value is more than 0 , we have to update the s_runtime of the process rather than the vruntime.

We also made changes to the linux-5.9.1/Makefile and added rtnice in the core -y .

We also have to update the linux-5.9.1/include/linux/sched.h to add the s_runtime to it.

We also have to update the linux-5.9.1/include/linux/syscalls.h to add the asmlinkage long sys_rtnice(int pid, long s_runtime) in it.

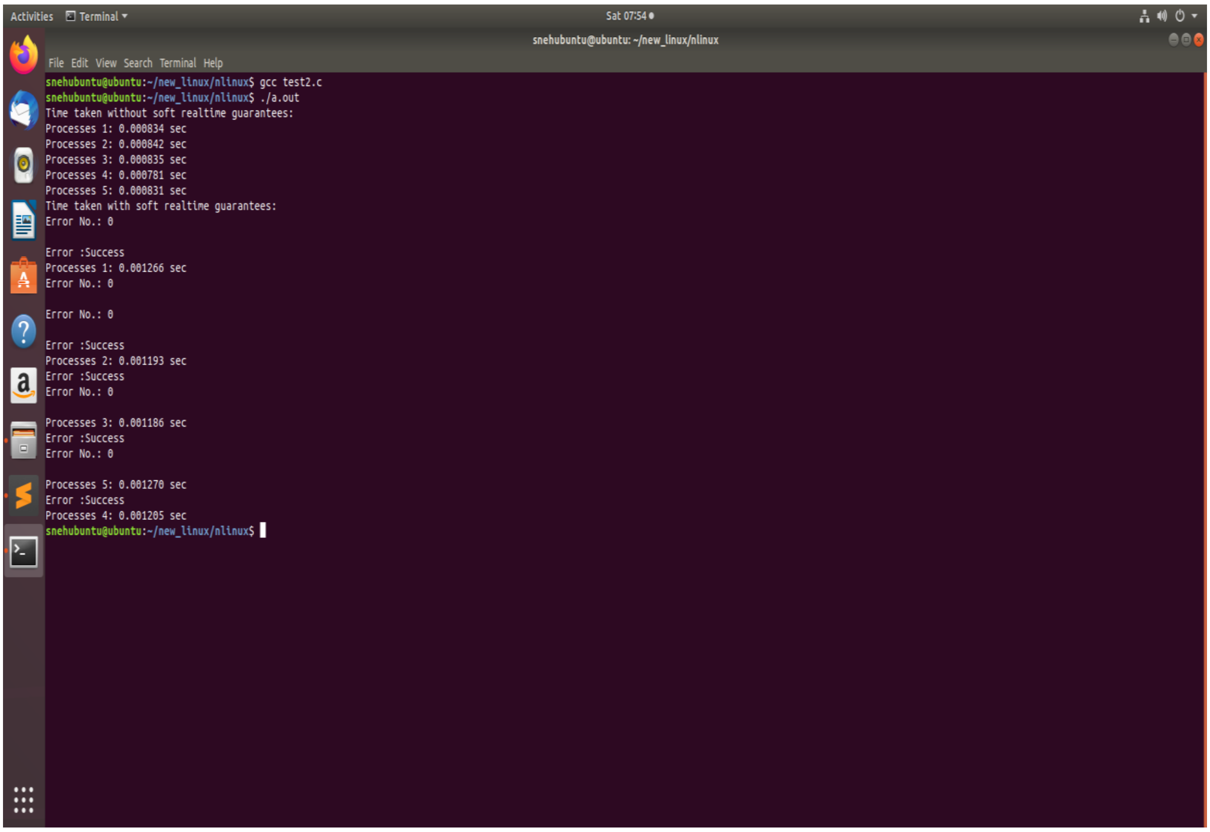
We add a new directory in the kernel directory to add our makefile and c code to add rtnice syscall.

RUN

The system will take the pid of five processes and allot a time slice to the syscall we have implemented.

The output will show two things-

The time of the process completion for the five processes without the soft real-time guarantee and the time of the process completion for the five processes with soft real-time guarantees.



Activities
Terminal
Sat 07:54
snehubuntu@ubuntu: ~/new_linux/nlinux
File Edit View Search Terminal Help

```
snehubuntu@ubuntu:~/new_linux/nlinux$ gcc test2.c
snehubuntu@ubuntu:~/new_linux/nlinux$ ./a.out
Time taken without soft realtime guarantees:
Processes 1: 0.000834 sec
Processes 2: 0.000842 sec
Processes 3: 0.000835 sec
Processes 4: 0.000781 sec
Processes 5: 0.000831 sec
Time taken with soft realtime guarantees:
Error No.: 0

Error :Success
Processes 1: 0.001266 sec
Error No.: 0

Error No.: 0

Error :Success
Processes 2: 0.001193 sec
Error :Success
Error No.: 0

Processes 3: 0.001186 sec
Error :Success
Error No.: 0

Processes 5: 0.001270 sec
Error :Success
Processes 4: 0.001205 sec
snehubuntu@ubuntu:~/new_linux/nlinux$ 
```

ERRORS

1>If the syscall is unable to find the process with the given pid then it will show an error .

2>If the syscall gets a timeslice <1 then also an error will be occurred.