

Assignment 3

Name: Vaibhav Girish

Roll No: 2019121

I used linux kernel v5.9.1 in this assignment. In the extracted folder I created a directory called rt_nice where I have declared the system call in the c file rt_nice.c

I declared the function using SYSCALL_DEFINE2 and passed two arguments to it. The two arguments were pid(integer type variable) of the process whose nice value had to be changed and the tim(integer type variable) the value to be changed to. I then look for my process from the pid and update the nice value in the system call.

The nice value or soft is stored in a variable called rt_nice in the se entity of the task variable. This was declared to be an unsigned 64 bit variable and initialized to 0 in core.c in the kernel files

In fair.c I decremented the rt_nice in the update_curr function with the delta_exec i.e. the amount of time the program has run for. If rt_nice is less than delta_exec then it is set to 0. In fair.c I also modified the entity_before() function so that the process with lower non zero rt_nice value gets picked first and if the two processes have the same rt_nice value then the process with lower vruntime gets picked.

I have a test.c function written in c to test my system call and verify the changes in the CFS scheduler with rt_nice.

The inputs passed are int pid and int tim.

This tim variable denotes the time in 100000000000000 nanoseconds i.e. the value is passed is multiplied with the above value to set the soft real time requirements (Please note that this value was used to show significant change in the time values).

The errors detected by my system call are:

No such process: It returns the value 3 which is equivalent to the error number ESRCH. This error is returned when the entered pid could not be found.

Invalid argument: It returns the value 22 which is equivalent to the error number EINVAL. This error is returned when the time passed is negative.

Sample Inputs and corresponding outputs:

- 1) As we can see in this case, the system call was called in the parent process and it had soft real time guarantees of 2×10^{10} passed to it, therefore the priority had to be for this process over the child process. The kernel log message prints the name of the process with the corresponding changed `rt_nice` value. Therefore the execution time of the parent process is less than that of the child.

```
long ret=syscall(440,(int)getpid(),2);
if(ret!=0){
    perror("Error");
    printf("Error no: %d\n",errno);
    return -1;
}
```

```
vaibhav@vaibhav-VirtualBox:~$ make run
gcc test.c
./a.out
Parent Process Time: 5.106007
Child Process Time: 9.012359
```

```
[ 301.021797] Proces: a.out
Nice value:2000000000000000
```

- 2) The above program calls the system call with time value as a negative number and this therefore returns an error Invalid argument and the corresponding error number is 22.

The kernel log message shows the error encountered.

```
long ret=syscall(440,(int)getpid(),-2);
if(ret!=0){
    perror("Error");
    printf("Error no: %d\n",errno);
    return -1;
}
```

```
vaibhav@vaibhav-VirtualBox:~$ make run
gcc test.c
./a.out
Error: Invalid argument
Error no: 22
make: *** [Makefile:4: run] Error 255
```

```
[ 162.103259] Invalid time
```

- 3) The above program called a system call with a negative pid, and there is no such process in the system and this therefore returns the error No such process with the corresponding error number 3
The kernel log message shows the error encountered.

```
long ret=syscall(440,-1,2);
if(ret!=0){
    perror("Error");
    printf("Error no: %d\n",errno);
    return -1;
}
```

```
vaibhav@vaibhav-VirtualBox:~$ make run
gcc test.c
./a.out
Error: No such process
Error no: 3
make: *** [Makefile:4: run] Error 255
```

```
[ 370.212786] Invalid PID
```