

# Write Up

## Implementation

As a part of this give assignment three, we were required to add a soft realtime requirement to a process – say something like each process that requires soft realtime guarantees must receive at least  $x$  units of time-slice. Every time the scheduler is called, we checked if the realtime guarantees of the process with soft realtime requirements are being met or not. We gave higher priority to a processes soft-realtime requirement compared to the vruntime that is normally considered. To implement this, I followed a few steps, which are as follows :

1. First, you have to add your system call required to implement in the given assignment to the kernel system call table.
2. Then you have to add soft runtime variable `s_runtime` in `sched.h` header file.
3. Then you have to initialize the `s_runtime` variable in `core.c` file under `__sched_fork()` function.
4. After that, you have to make changes in two functions in `fair.c` file i.e.
  1. `entity_before`
  2. `update_curr`

## Usage

User Input

The user needs to enter the soft realtime value.

User Output

The output would be the time taken for a program to execute.

We will observe that the time taken for the process to run without the soft real time guarantee is more than that of when the process is run with soft real time guarantee.

## Error handling

There are two types of error handling in user input -

1. In case the user tries to give a negative soft real time value, we notify the user.

2. In case the user gives an invalid value of soft real time i.e., char, float, we notify it to the user.