Running the program.

You'll have to open 4 terminal windows.
1st
gcc testing.c -o testing
./testing
2n
gcc testing2.c -o testing2
./testing2
3rd
./testing
4th
Htop

User can edit the value of pid according to the given in htop.

Added tym filed in sched_entity in include/linux/sched.h
Intialed it tym = 0 in core.c in kernel/sched/core.c
Modified the update_curr in kernel/sched/fair.c
Added modify_vruntime and print_execution system calls in
/home/kern/build/linux-5.14.8/arch/x86/entry/syscalls/syscall_64.tbl

Then added their macro in
/home/kern/build/linux-5.14.6/kernel/sys.c

I used find_get_pid() to get the pid_struct and pid_task() to get task_struct
And inceremented the tym by 10000000.

I assigned the value of print_execution of the process to variable time and printed its value if it was incremented and finally printed the total execution time of the process.