I initially initialized the three process IDs pid1, pid2, and pid3 once starting with the main function. Then I used fork() to create three processes, each of which does the same counting operation count from 1 to 2^32. I use the clock_gettime() function in these routines to determine how long it takes the function to complete its duty by using the attribute CLOCK_REALTIME.

I have created three functions to do the task. Then I am using sched_priority to set the priorities of the policies. I am using waitpid() to wait for the other process to get into the ready state. I have created another file counter.c in which I have a for loop which counts 1 to 2^32. I am calling that file in my main file using the execl() function. Then after counting the code is printing the times taken by the policies to count when they have different priorities. In observation the other always takes more time then RR and FIFO.