We built a test file called testgetpinfo.c. In this file, we generated 2 children processes by calling fork and let both of them spin by themselves. We also call the getpinfo to check the status of each process, returning us the process pid, its current priority and its running ticks. The spin function is restricted by spinning to 1000 times. Plus init and sh system calls, we have total 6 system calls in this program. One of the processes calls a sleep function in its codes thus when it is running and displaying itself process state on the screen. Another test is almost same without the sleep call. We can tell from the graph1.pdf and graph2.pdf that process 6 (pid = 6) is sleeping while other processes have finished running. Pretty much the graph shows a nice mlfq properties, having 4 different priority levels and each level has a different running time ticks. We are aware this scheduler is quite good, when a new time process comes in, it will be arranged to top level and running for a few time ticks. But this scheduler is not so perfect, sometimes it will still let lower level process running while there are still some top level processes waiting. We think there is place to improve this scheduler, checking time tick mechanism and queue process. But generally, this satisfies the requirement for this project and it works fine.