The results of sample program (with n=10 programs) running on different schedulers are as follows:

<u>DEFAULT:</u>

CPU bound:
Average running time: 22
Average sleeping time: 0
Average turnaround time: 23

Short Tasks:
Average running time: 0
Average sleeping time: 0
Average turnaround time: 1

I/O bound:
Average running time: 0
Average sleeping time: 30
Average turnaround time: 31

<u>FCFS:</u>

CPU bound:
Average running time: 83

Average sleeping time: 0
Average turnaround time: 86

Short Tasks:
Average running time: 3
Average sleeping time: 0
Average turnaround time: 5

I/O bound:
Average running time: 3
Average sleeping time: 30
Average turnaround time: 37

PBS:

CPU bound:
Average running time: 194
Average sleeping time: 0
Average turnaround time: 197

Short Tasks:
Average running time: 0
Average sleeping time: 0

Average turnaround time: 2

I/O bound:
Average running time: 0
Average sleeping time: 30
Average turnaround time: 33

MLFQ:

CPU bound:
Average running time: 14
Average sleeping time: 0
Average turnaround time: 14

Short Tasks:
Average running time: 0
Average sleeping time: 0
Average turnaround time: 0

I/O bound:
Average running time: 7
Average sleeping time: 30
Average turnaround time: 37

<u>My Analysis:</u>

Although MLFQ might look like the fastest among all the schedulers for CPU bound process, it is actually the slowest since the ticks are themselves very slow for MLFQ.
This is because incrementing ticks and running scheduler takes much more time in MLFQ due to overhead of finding a process to execute using bestProcess function, and overhead of increasing/decreasing priority & incrementing tickcounter.