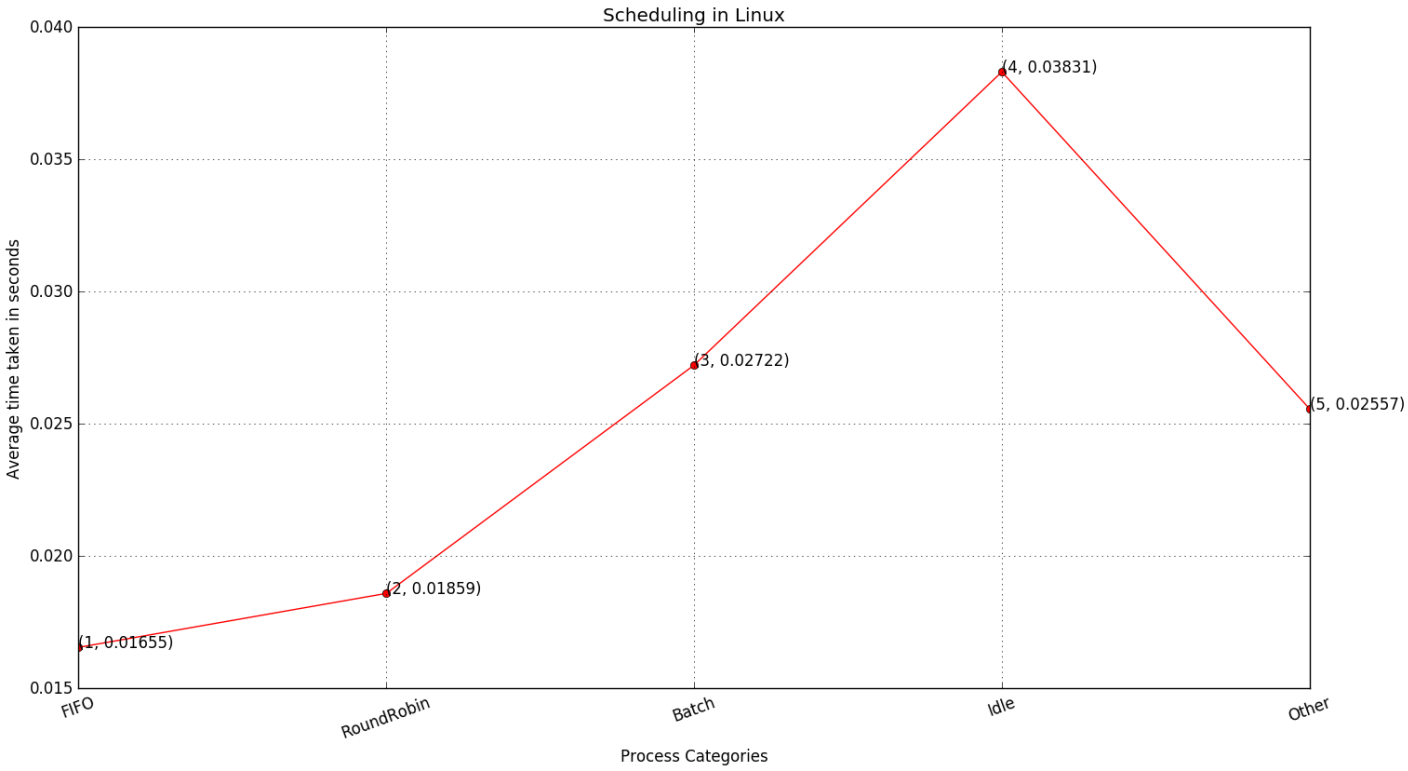
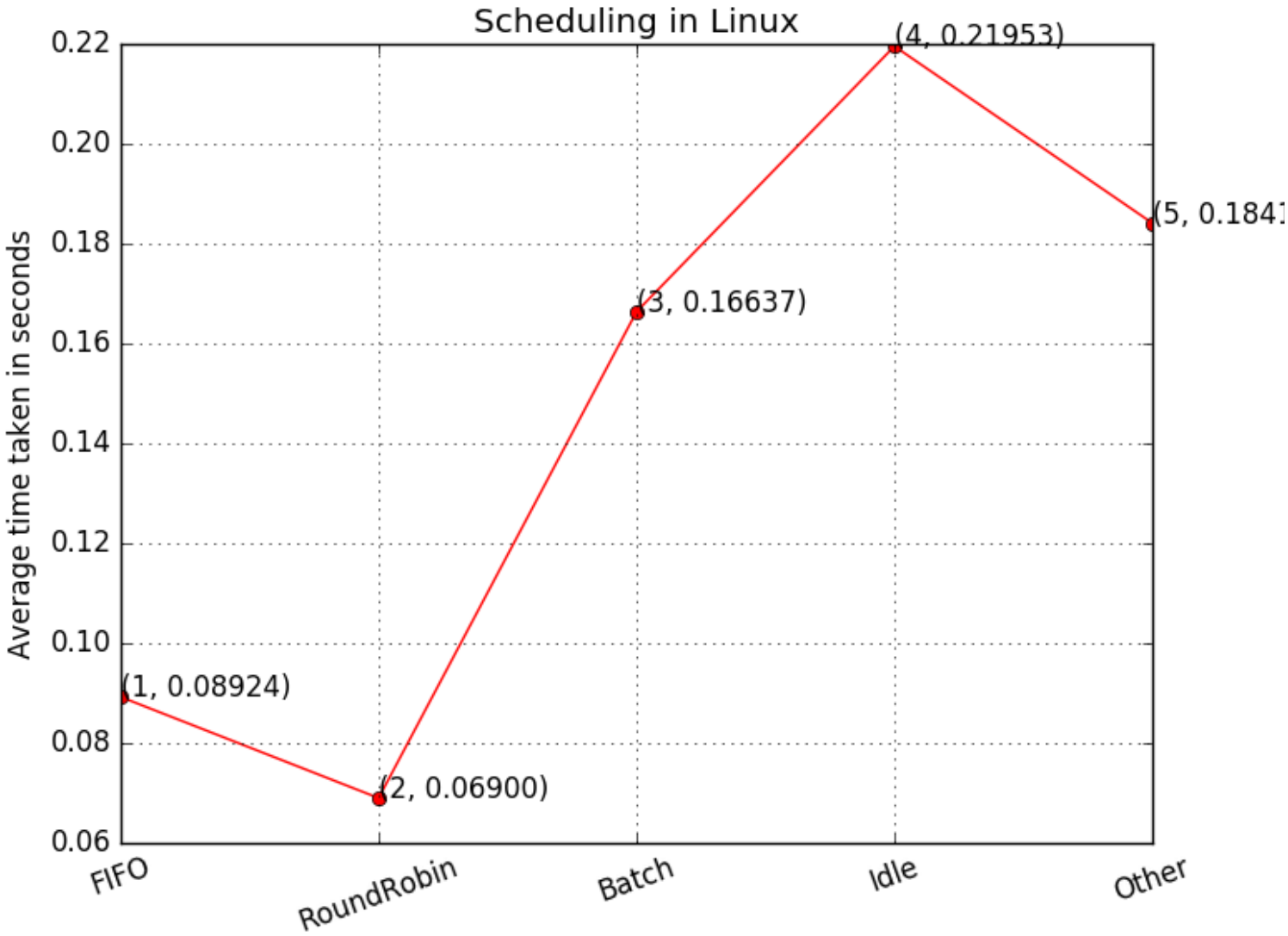
Scheduling in Linux

Graph representing various process/scheduler categories v/s average time of each category :



The above graph represents the average time taken by each of the scheduler_policies. The maximum time taken is by the scheduler – **SCHED_IDLE**, which is used for running very low priority background jobs, whereas, the Real-Time scheduler_policies generally require less time, with **SCHED_FIFO**, which is a first-in, first-out policy, taking the least average turnaround time.

It’s not the case that SCHED_FIFO always takes least time, for example as shown in the graph below :



Its quite notable that **SCHED_RR** may also take somewhat less time than **SCHED_FIFO**, but in most cases, **SCHED_FIFO** performs better. These both implement the fixed-priority real-time scheduling specified by the POSIX standard. Tasks with these policies **preempt** every other task, which can thus easily go into **starvation** (if they don't release the CPU).

The difference between SCHED_FIFO and SCHED_RR is that among tasks with the same priority, SCHED_RR performs a round-robin with a certain timeslice; SCHED_FIFO, instead, needs the task to explicitly yield the processor, thus, requiring somewhat less time than the former. Also, SCHED_FIFO processes do not preempt SCHED_RR processes of the same priority.

Its also quite clear that SCHED_OTHER and SCHED_BATCH, nearly take the same time, with SCHED_OTHER taking a less amount as compared to SCHED_BATCH. These comprise the category of the “normal” schedulers used by the Linux system.

Threads that are scheduled with **SCHED_BATCH** are assumed to be non-interactive, but CPU bound and optimized for throughput. Thus, this policy is more cache-friendly. Also, in case of Symmetric MultiProcessing, **SCHED_BATCH** will migrate to a core with high idleness (with respect to non-batch threads). That’s why, SCHED_BATCH performs better than SCHED_IDLE, and sometimes, SCHED_OTHER also.

Thus, in all, we can say that Real-time scheduler policies are better than the Normal scheduler policies.