

Metric : Uptime, ticks_running, Avg Turnaround time

For measuring this performance of each command we have created a test file name performance.c in xv6. In performance.c file we have used three if statements for performing analysis for below commands which are given in the shell. The commands which are given in shell are called used by exec().

For stressfs: performance stressfs

For find : performance find . -name file_name

Eg: performance find . -name ls

For cat README | uniq -c :

We have created a input.txt for running uniq command.

performance cat input.txt | uniq -c

we have analyzed the performance of ls command for non-trivial command or sequence of commands.

Simple Scheduler

vs

Default scheduler

	Turnaround	,uptime ,	ticks_running	Turnaround ,	uptime	ticks_running
stressfs	33	1435	141	30	686	177
find	5	7964	166	9	3396	202
Cat input.txt uniq	4	1823	12	7	1197	11
ls	20	427	42	24	5125	230

Advanced Scheduler

vs

Default scheduler

	Turnaround ,	uptime ,	ticks_running	Turnaround	, uptime,	ticks_running
stressfs	20	3077	246	30	686	177
find	4	9844	270	9	3396	202
Cat input.txt uniq	3	1777	11	7	1197	11
ls	23	16552	305	24	5125	230

Simple Scheduler

vs

Advanced scheduler

	Turnaround ,	uptime ,	ticks_running	Turnaround ,	uptime,	ticks_running
stressfs	33	1435	141	20	3077	246
find	5	7964	166	4	9844	270
Cat input.txt uniq	4	1823	12	3	1777	11
ls	20	427	42	23	16552	305

Overall Discussion:

Based on the above results we can say:

→The performance data analysis from comparing these scheduling algorithms reveals a crucial trade-off between task execution speed and system load overall. In particular, it appears that expediting task completion was a primary design goal for the Advanced Scheduler. Because of this design decision, it is ideal for environments where completing tasks fast is crucial, even if it means the system will be under more stress. On the other hand, the Simple Scheduler appears to take a more conservative approach to scheduling. It tends to keep the system less occupied, which might result in a slower task completion rate.

→Sitting between these two, the Default Scheduler offers a more balanced performance. It doesn't specifically excel at either reducing system load or hastening task completion. Because of its moderate performance, it's a flexible choice that may be applied to a variety of general-purpose situations where high performance in either direction isn't required.

→The decision on which scheduler to employ should, therefore, be informed by the specific demands and characteristics of the workload faced by the system. For environments where task completion speed is of the essence, the Advanced Scheduler would be the go-to option. Conversely, scenarios that call for a more measured use of system resources might benefit more from the judicious scheduling approach of the Simple Scheduler.