

Performance Analysis Report

COP6611.001S23 - Operating Systems - Project 2 Report

Katarina Capalbo, John Maurer, Azim Ibragimov

Date: 3/2/2023

Overview

The following is a performance analysis comparing the different schedulers we have implemented in Project 2. We will be comparing them based on the following metrics: turnaround time and response time. To capture these metrics we will utilize the `uptime()` system call and the `time_scheduled()` system call. To help analyze and view our results, we have created a user program `ptime`. By running `'ptime [fifo|priority|default]'`, we will create multiple child processes that will execute the following user program commands: `stressfs`, `uniq`, `find`. Additionally, we will perform a second test by simply running the command `"cat README | uniq"` to gather additional data to assist us in our analysis. Each scheduler will be compared with each other against the trials performed in `ptime` and the trials performed by `"cat README | uniq"`. Metrics will be measured in ticks.

Testing System Environment

- Windows WSL/Ubuntu

Schedulers Types Tested

- xv6 Default Scheduler: RR with time quantum based on timer interrupt time.
- Basic Scheduler: FIFO
- Interesting Scheduler: Priority with FIFO Queues

Testing Methodology

We perform the following steps in order to collect our data:

1. Run `make clean`.
2. To run the default scheduler, run `make qemu-nox`.
3. Run the user program: `ptime default`.
4. Screenshot and note metrics displayed.
5. Run the command: `cat README | uniq`.
6. Screenshot and note metrics displayed
7. Press 'Ctrl-A + X' to exit.
8. Run `make clean`.
9. Run `make qemu-nox SCHEDULER=PRIORITY` to run the priority scheduler.
10. Run the user program: `ptime priority`.
11. Screenshot and note metrics displayed.
12. Run the command: `cat README | uniq`.
13. Screenshot and note metrics displayed
14. Press 'Ctrl-A + X' to exit.
15. Run `make clean`.
16. Run `make qemu-nox SCHEDULER=FIFO` to run the FIFO scheduler.
17. Run the user program `ptime fifo`.
18. Screenshot and note metrics displayed.
19. Run the command: `cat README | uniq`.
20. Screenshot and note metrics displayed
21. Press 'Ctrl-A + X' to exit.

Default Scheduler Metrics

Please see submission folders for screenshots and text files of output from performance testing: "Default Scheduler Performance Testing Screenshots"

Average Turnaround Time: $((867-809) + (891 - 819) + (877 - 839))/3 = (58 + 72 + 38)/3 = 56$

Average Response Time: $((810-809)+(839-819)+(870-839))/3 = (1+20+31)/3 = 17.33$

Running 'ptime default' results:

Process Name	Action Process Performs	Arrival Time	First Start Time	Final End Time	Time Scheduled (time it was in a RUNNING state)
PID 4	Stressfs 1	809	810	867	7
PID 9	Uniq README	819	839	891	38
PID 10	Find . -name ls	839	870	877	2

Running 'cat README | uniq' results:

Average Turnaround Time: $((650-622) + (659 - 622))/2 = (28 + 37)/2 = 32.5$

Average Response Time: $((623-622)+(624-622))/2 = (1+2)/2 = 1.5$

Process Name	Action Process Performs	Arrival Time	First Start Time	Final End Time	Time Scheduled (time it was in a RUNNING state)
PID 4	Cat README	622	623	650	1
PID 5	uniq	622	624	659	37

FIFO Scheduler Metrics

Please see submission folders for screenshots and text files of output from performance testing: "FIFO Scheduler Performance Testing Screenshots"

Average Turnaround Time: $((457-374) + (454-384) + (463-402))/3 = (83 + 70 + 61)/3 = 71.3$

Average Response Time: $((375-374)+(401-384)+(458-402))/3 = ((1) + 17 + 56)/3 = 24.67$

Running 'ptime fifo' results:

Process Name	Action Process Performs	Arrival Time	First Start Time	Final End Time	Time Scheduled (time it was in a RUNNING state)
PID 4	Stressfs 1	374	375	457	2
PID 9	Uniq README	384	401	454	39
PID 10	Find . -name ls	402	458	463	3

Running 'cat README | uniq' results:

Average Turnaround Time: $((708-670) + (717-670))/2 = (38 + 47)/2 = 42.5$

Average Response Time: $((671-670)+(671-670))/2 = (1 + 1)/2 = 1$

Process Name	Action Process Performs	Arrival Time	First Start Time	Final End Time	Time Scheduled (time it was in a RUNNING state)
PID	Cat README	670	671	708	1
PID	uniq	670	671	717	37

Priority Scheduler Metrics

Please see submission folders for screenshots and text files of output from performance testing: "Priority Scheduler Performance Testing Screenshots"

Average Turnaround Time: $((623-453) + (547-463) + (584-473))/3 = (170 + 111 + 111)/3 = 130.57$

Average Response Time: $((455-453)+(471-463)+(573-473))/3 = (2 + 8+100)/3 = 36.67$

Running 'ptime priority' results:

Process Name	Action Process Performs	Arrival Time	First Start Time	Final End Time	Time_Scheduled (time spent in RUNNING state)
PID 4 - PRIORITY 3	Stressfs 1	453	455	623	1
PID 9 - PRIORITY 2	Uniq README	463	471	547	57
PID 10 - PRIORITY 7	Find . -name ls	473	573	584	2

Average Turnaround Time: $((965-917) + (977-917))/2 = (48 + 60)/2 = 54$

Average Response Time: $((918-917)+(918-917))/2 = (1 + 1)/2 = 1$

Running 'cat README | uniq' results:

Process Name	Action Process Performs	Arrival Time	First Start Time	Final End Time	Time Scheduled (time it was in a RUNNING state)
PID 4 - PRIORITY	Cat README	917	918	965	1

2					
PID 5 - PRIORITY 3	uniq	917	918	977	47

FIFO vs. Default

‘Ptime’ performance comparison

Scheduler	Turnaround Time	Response Time
FIFO	71.3	24.67
Default	56	17.3

‘cat README | uniq’ comparison

Scheduler	Turnaround Time	Response Time
FIFO	42.5	1
Default	32.5	1.5

Priority vs. Default

‘Ptime’ performance comparison

Scheduler	Turnaround Time	Response Time
Priority	130.57	36.67
Default	56	17.3

‘cat README | uniq’ comaprison

Scheduler	Turnaround Time	Response Time
Priority	54	1
Default	32.5	1.5

Priority vs. FIFO

‘Ptime’ performance comparison

Scheduler	Turnaround Time	Response Time
FIFO	71.3	24.67
Priority	130.57	36.67

‘cat README | uniq’ comparison

Scheduler	Turnaround Time	Response Time
FIFO	42.5	1
Priority	54	1

Analysis & Discussion

From analyzing the results from our three schedulers, we can conclude that the Round Robin default xv6 scheduler performs best when measured against turnaround time and response time. When using those metrics to compare FIFO and Priority scheduler, FIFO takes second place for both its turnaround time and response time. This leaves our implementation of Priority scheduler to have the worst possible metrics in regard to turnaround time and response. Knowing what we do about these schedulers in regards to these specific metrics, our results here were to be expected. When comparing our shorter analysis, 'cat README | uniq', we can see that these results are consistent.

The Round Robin default xv6 scheduler utilizes time slices to enhance response time and turnaround time specifically. These time slices support running both heavier and lighter processes without letting one dominate. Whereas with FIFO, if a longer process gets to run first, (in our case, uniq performed consistently longer) based on the arrival time of the other processes, both the response time and turnaround time will suffer. For our implementation of priority scheduler, it suffers the same fate as FIFO if a higher priority has a heavier running time. Additionally, since the priority scheduler implements FIFO for processes of the same priority, it would be expected to have the results of FIFO with additional overhead causing additional delays.