

1 Abstract

2 Introduction

A scheduler is a vital part of any modern operating system. It decides how to efficiently utilize the CPU. This is important in the resource manager view of an operating system, that the OS is only supposed to manage resources. There are three schedulers that I will test, the Completely Fair Scheduler(CFS), also known as SCHED_OTHER, the First in First Out(FIFO) scheduler, also known as SCHED_FIFO, and the Round Robin(RR) Scheduler, also known as SCHED_RR.

Every test was preformed on a Macbook Pro running Ubuntu 12.04 through VMWare Fusion. The VM has 2Gb of RAM, 2 Cores of the Processor, and has hardware virtualization turned on (Intel VT-x). This provided very uniform data that rarely deviated more than 1-2%.

3 Method

The testing code will test three different kinds of programs, CPU bound processes, I/O bound processes, and a mixed combination of these two. The CPU bound process repeatedly calculates the product of two prime numbers, then preforms a factoring algorithm to find the two prime numbers once again. I did this since this is a common process that a computer might do in real life to, for instance, crack a RSA key. It is also entirely CPU bound. Each individual process has no I/O when measuring total execution time. This code was made using algorithms I found online [1][2]. The I/O bound process will read a line at a time from /dev/urandom and write that line to /dev/null. This is going to have a comparable effect to reading and writing to any file, a common application in everyday code. The mixed processes combine these two tasks, preforming both in an interweaved fashion. This is more representative of your everyday tasks that you run on a computer. Each of these three types of processes are designed so as to use comperable amonuts of time on the CPU. Each uses

about 0.58 seconds of time where that process is running on the CPU. This way comparing data will be much easier.

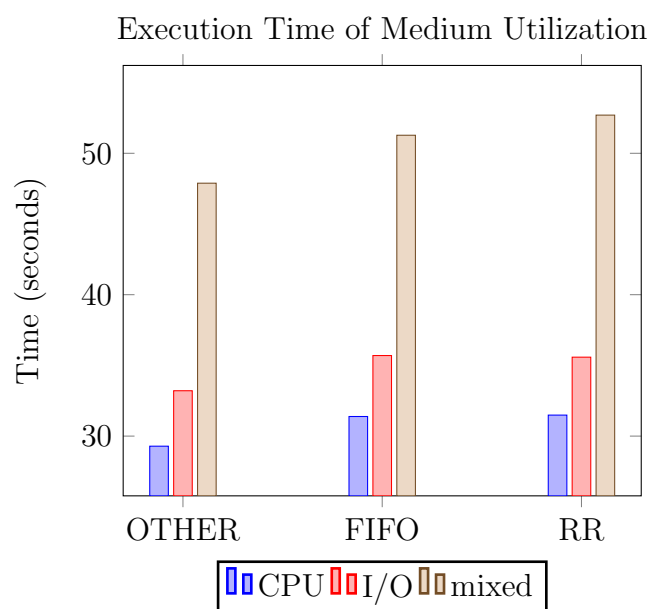
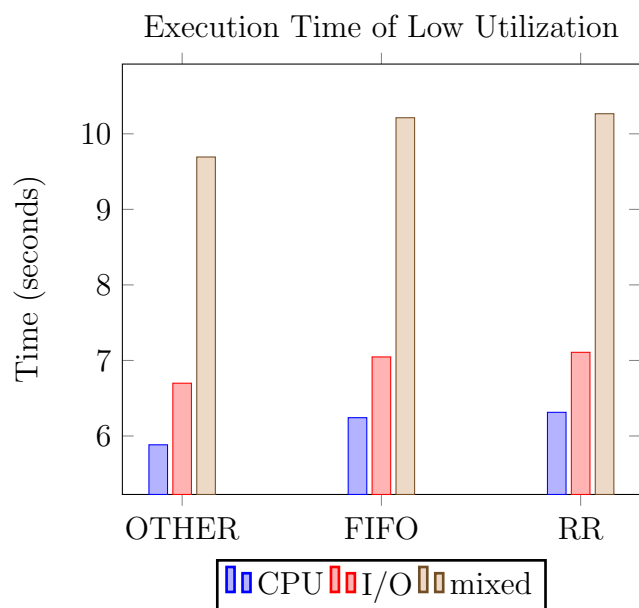
The main code is pa3.c. This must be run as root like so: `sudo .`

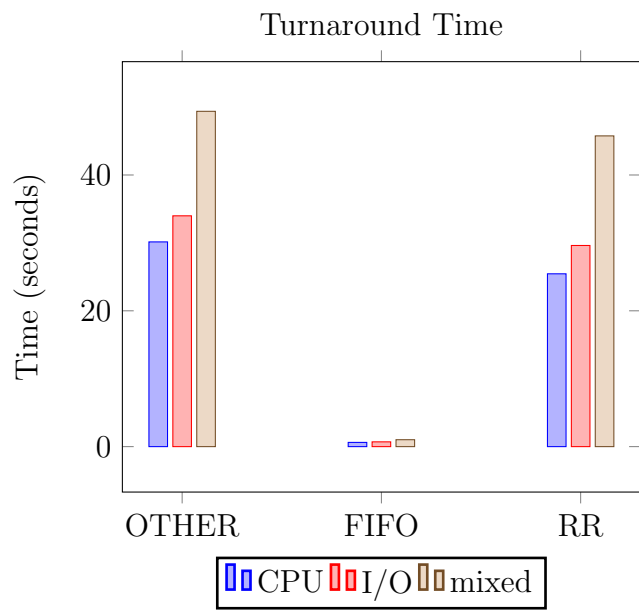
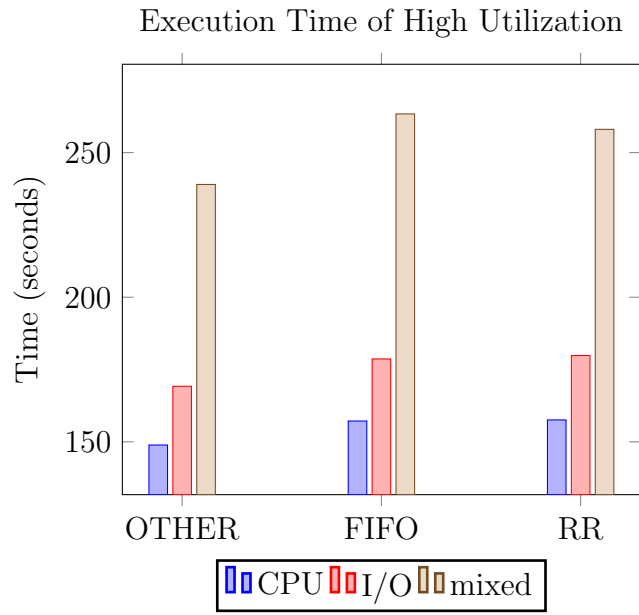
`pa3 <number of processes> <scheduling scheme> <process type> <test type>`. The number of processes can be any integer. For my tests I consider 10 to be low, 50 to be medium, and 250 to be high. The scheduling scheme can be either `SCHED_OTHER`, `SCHED_FIFO`, or `SCHED_RR`. The process type can be either `cpu`, `io`, or `mixed`. The test type is the desired outcome of the test. In order to get more accurate data, I decided to split up the tests. The test type can be either `extime`, `restime`, or `turntime`. `extime` is the time it takes to execute every process. This is the total time from before the parent process forks each child, to after the parent process has waited for every child. `restime` is the response time of the scheduler. This is the time from when the parent begins to fork each child, until after each individual child has had a chance to begin. `turntime` is the Turnaround time of each process. This is the time from when the process starts to the time when the process ends. All of these parameters are things that a good scheduler would want to minimize.

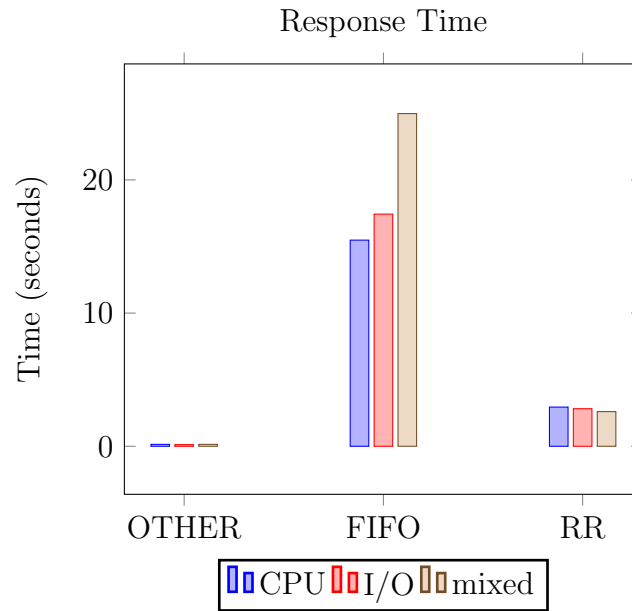
Running the code as listed above will give you a value, or a collection of values, but it is not designed to be run as a standalone test. The shell scripts included will be a much easier and more efficient test. The three shell scripts each run the three tests that can be run, and each returns the results in an easily formatted way. Each shell script will spit out to stdout a table in LaTeX formatting that can be easily copied and pasted anywhere it is needed. LaTeX is the standard for most academic articles, and is very easy to generate from a shell script. These scripts will test each level of utilization, each scheduling scheme, and each process type, 10 times, then average out these values. All of this data can be seen in Appendix A. The only alteration to the code that was spit out by the scripts was a few line breaks so as to fit everything on multiple pages.

4 Results

The results from the scripted tests can be fully examined in Appendix A. The averaged values will be explored in this section. This is the Execution time of the processes using Low Utilizations(10 processes):







5 Analysis

6 Conclusion

7 References

[1] - http://en.wikipedia.org/wiki/Fast_inverse_square_root

[2] - <http://www.geeksforgeeks.org/print-all-prime-factors-of-a-given-number/>

8 Appendix A - Raw Data

Schedule	Utilization	Type	Trial	Execution Time (seconds)
SCHED_OTHER	Low	CPU	1	5.8735
			2	5.9062
			3	5.8699
			avg	5.8832
		I/O	1	6.6987
			2	6.6874
			3	6.7100
			avg	6.6987
		mixed	1	9.6881
			2	9.8189
			3	9.5725
			avg	9.6932
	Medium	CPU	1	29.3128
			2	29.2543
			3	29.2929
			avg	29.2867
		I/O	1	33.1420
			2	33.2971
			3	33.1849
			avg	33.2080
		mixed	1	48.3485
			2	47.5061
			3	47.8008
			avg	47.8851
	High	CPU	1	147.1985
			2	150.6363
			3	148.8392
			avg	148.8913

Schedule	Utilization	Type	Trial	Execution Time (seconds)
SCHED_OTHER	High	I/O	1	168.8146
			2	169.8647
			3	168.9283
			avg	169.2025
		mixed	1	238.4620
			2	238.4029
			3	240.1449
			avg	239.0033
SCHED_FIFO	Low	CPU	1	6.2712
			2	6.2239
			3	6.2316
			avg	6.2422
		I/O	1	7.0592
			2	7.0326
			3	7.0475
			avg	7.0464
		mixed	1	10.1941
			2	10.2298
			3	10.2144
			avg	10.2128
	Medium	CPU	1	31.4025
			2	31.3293
			3	31.4259
			avg	31.3859
		I/O	1	35.5476
			2	35.4535
			3	36.0892
			avg	35.6968
		mixed	1	51.0813
			2	51.2152
			3	51.5310
			avg	51.2758

Schedule	Utilization	Type	Trial	Execution Time (seconds)
SCHED_FIFO	High	CPU	1	157.4083
			2	157.3154
			3	156.8339
			avg	157.1859
		I/O	1	178.4461
			2	178.0876
			3	179.4295
			avg	178.6544
		mixed	1	256.0873
			2	269.7028
			3	264.3920
			avg	263.3940
SCHED_RR	Low	CPU	1	6.2610
			2	6.3182
			3	6.3604
			avg	6.3132
		I/O	1	7.1231
			2	7.1374
			3	7.0624
			avg	7.1076
		mixed	1	10.2902
			2	10.2280
			3	10.2797
			avg	10.2660
	Medium	CPU	1	31.4602
			2	31.3600
			3	31.6350
			avg	31.4851
		I/O	1	35.6692
			2	35.5747
			3	35.5019
			avg	35.5819

Schedule	Utilization	Type	Trial	Execution Time (seconds)
SCHED_RR	Medium	mixed	1	51.4243
			2	54.2640
			3	52.4101
			avg	52.6994
	High	CPU	1	157.5337
			2	159.9028
			3	157.5616
		I/O	1	182.9195
			2	178.2804
			3	178.3100
			avg	179.8366
		mixed	1	259.3416
			2	257.3794
			3	257.4659
			avg	258.0623

Schedule	Type	Trial	Turnaround time	Response time	CPU Time
SCHED_OTHER	CPU	1	30.2187	0.1408	0.61
		2	30.0752	0.1332	0.61
		3	30.1449	0.1585	0.61
		avg	30.1463	0.1442	0.61
	I/O	1	33.7790	0.1119	0.67
		2	34.5134	0.1336	0.67
		3	33.6690	0.1209	0.67
		avg	33.9871	0.1221	0.67
	mixed	1	49.8178	0.1476	0.98
		2	49.0224	0.1710	0.98
		3	49.2882	0.1094	0.98
		avg	49.3761	0.1427	0.98

Schedule	Type	Trial	Turnaround time	Response time	CPU Time
SCHED_FIFO	CPU	1	0.6263	15.5352	0.61
		2	0.6200	15.4867	0.61
		3	0.6230	15.4114	0.61
		avg	0.6231	15.4778	0.61
	I/O	1	0.7032	17.5551	0.67
		2	0.7080	17.5184	0.67
		3	0.6995	17.2163	0.67
		avg	0.7036	17.4299	0.67
	mixed	1	1.0200	25.0276	0.98
		2	1.0177	24.8969	0.98
		3	1.0181	25.0149	0.98
		avg	1.0186	24.9798	0.98
SCHED_RR	CPU	1	23.6338	3.1789	0.61
		2	28.1985	3.1232	0.61
		3	24.5107	2.5229	0.61
		avg	25.4477	2.9417	0.61
	I/O	1	28.8423	3.1798	0.67
		2	30.6580	2.7746	0.67
		3	29.3590	2.5001	0.67
		avg	29.6198	2.8182	0.67
	mixed	1	44.7533	2.5243	0.98
		2	45.9716	2.7949	0.98
		3	46.5461	2.4853	0.98
		avg	45.7570	2.6015	0.98

9 Appendix B - Description of Files