

1 Abstract

The Linux Scheduler has been the Completely Fair Scheduler (CFS) for a while now, but is it the most efficient scheduler? That is up for debate. This report will address the pros and cons of each scheduling scheme as well as comparing it to a completely different scheduler. We will show that the Brain Fuck Scheduler (BFS) is far more efficient for everyday personal computing tasks, and much more suited to what Linux home distrobutions should build default with. We will also show that the default scheduling scheme of the CFS is far better than the other real time schemes.

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 a few percentage points.

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 repedely

calculates the product of two prime numbers, then performs 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, performing 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 comparable amounts 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 .`

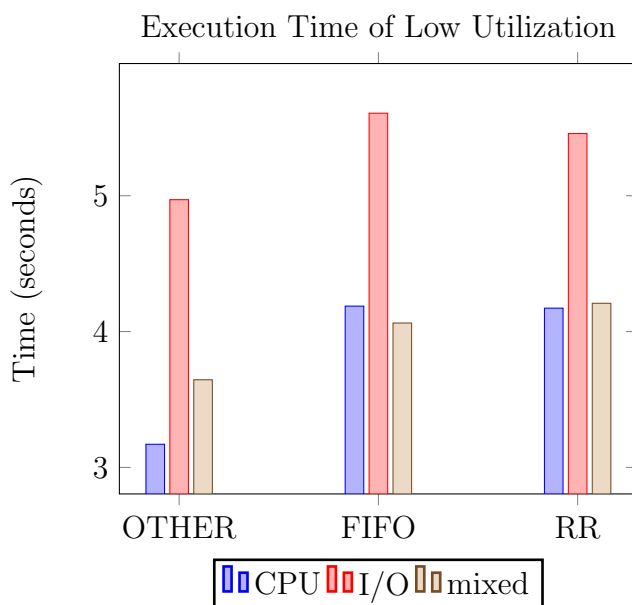
`pa3 <number of processes> <scheduling scheme> <process type> <test type>`. The number of processes can be any integer. For my tests I consider 10 to be low, 50 to be medium, and 250 to be high. The scheduling scheme can be either `SCHED_OTHER`, `SCHED_FIFO`, or `SCHED_RR`. The process type can be either `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.

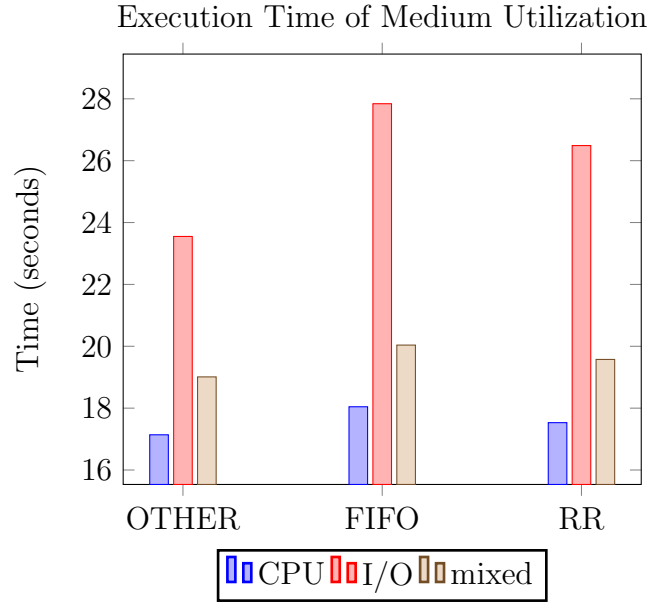
I chose the testing parameters I did because I feel that they most reflect the everyday use of common home computers. Other parameters may matter to a supercomputer or a server, but for a home computer, at the end of the day, you want to know how fast is my computer going to go? Turnaround time, response time, and total execution time, are the main factors involved in this.

4 Results

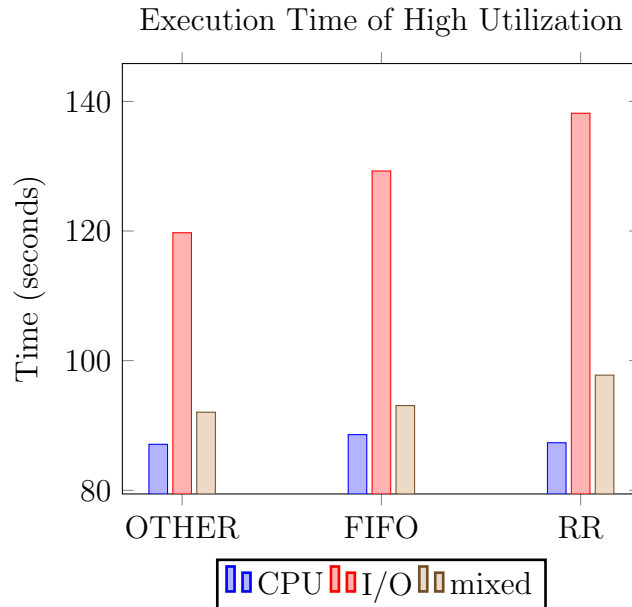
The results from the scripted tests can be fully examined in Appendix A. The averaged values will be explored in this section. This is the Execution time of the processes using Low Utilizations (10 processes):



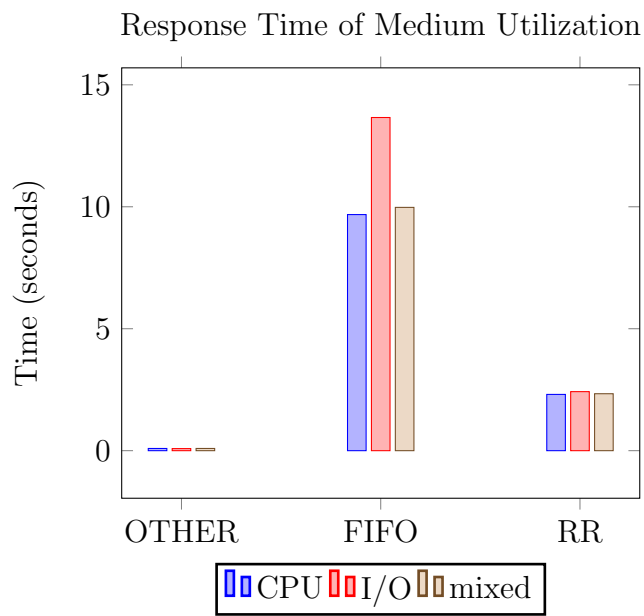
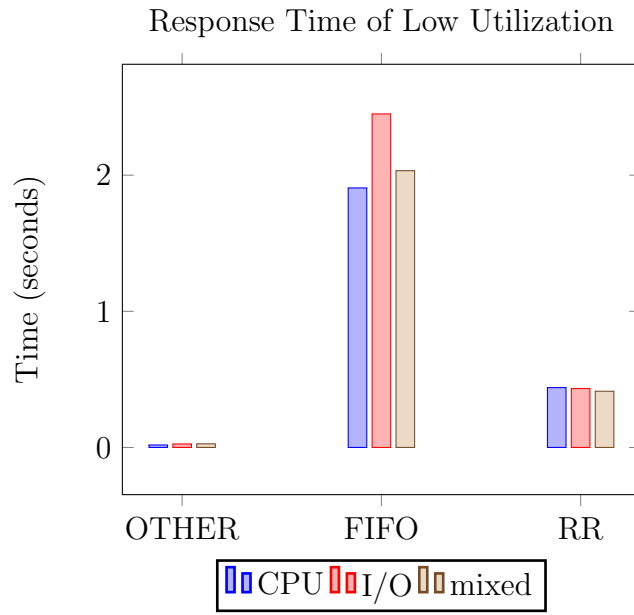
As you can see, the CFS scheduler's default behavior is far more efficient than the other real-time policies in the long run. For a CPU based process, SCHED_OTHER is by far the fastest. FIFO and RR are relatively equivalent. When we run with medium utilization (50 processes), we see that:

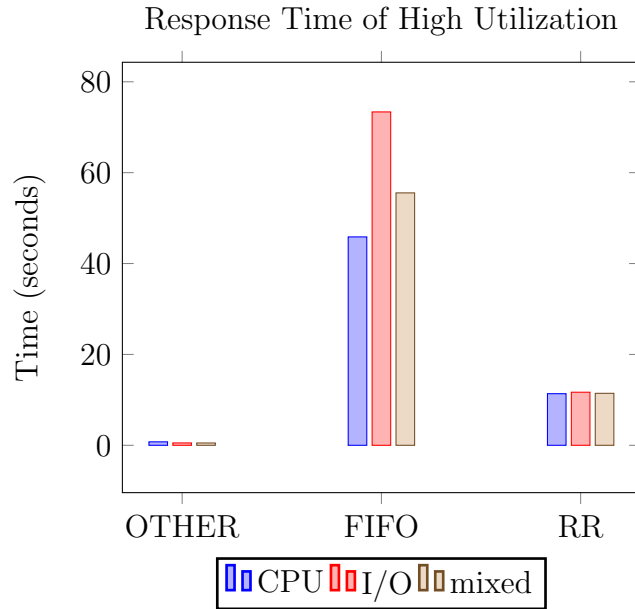


At medium utilization, the benefit of SCHED_OTHER begins to taper off, and is not so apparent. For mixed processes, the performance is comparable. When we use high utilization (250 processes) we find:

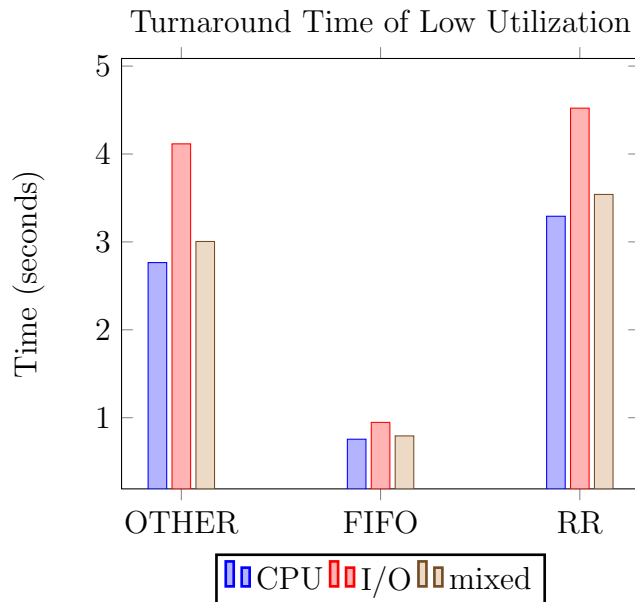


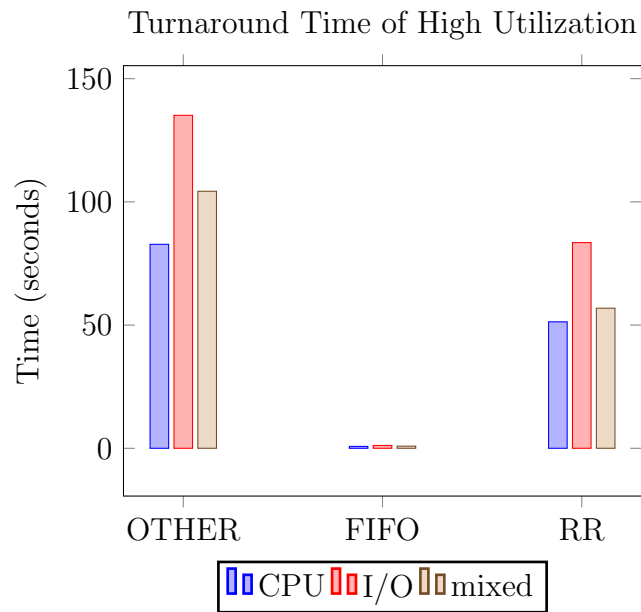
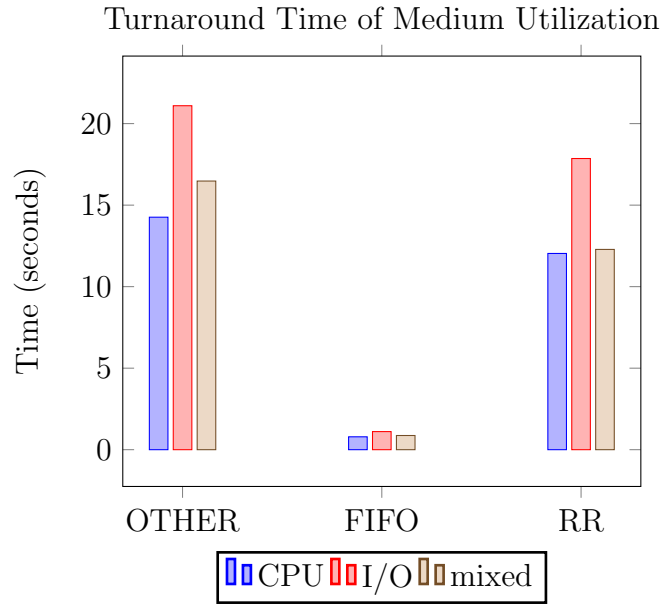
Once again the performance for mixed and cpu bound processes are comparable. An I/O bound program, however, is far faster on SCHED_OTHER. This is because the process of blocking it, and then moving to the next process becomes much more apparent at high amounts of processes. We can see the response time in the following charts:





The Response time is relatively the same regardless of utilization. SCHED_OTHER is by far the fastest, but SCHED_FIFO is by far the slowest. This is what makes sense intuitively, since FIFO will wait until a process is done before beginning on the next one. We can see the turnaround time in the following charts:

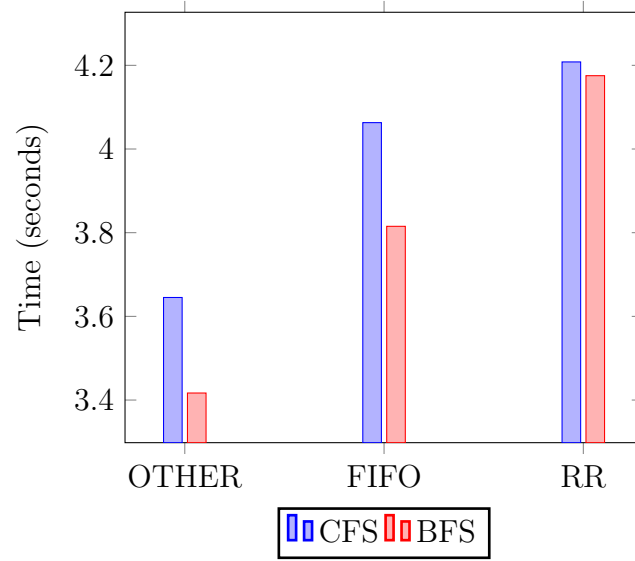




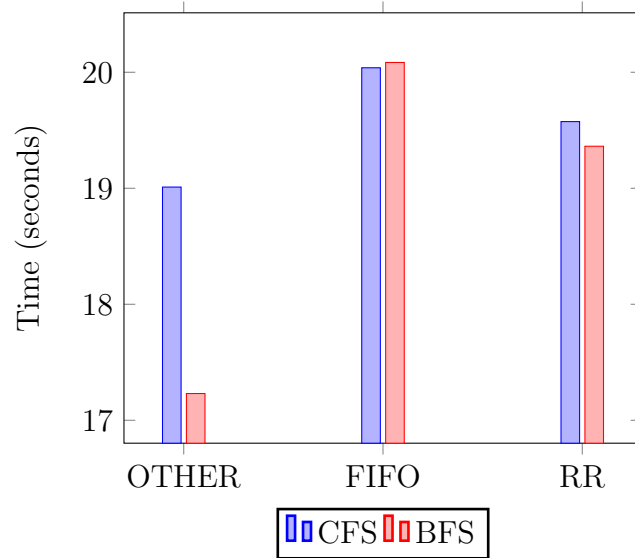
As the amount of processes increases, the turnaround time of FIFO and RR become relatively better to SCHED_OTHER, which maintains the worst turnaround time.

We can also compare this data to the BFS:

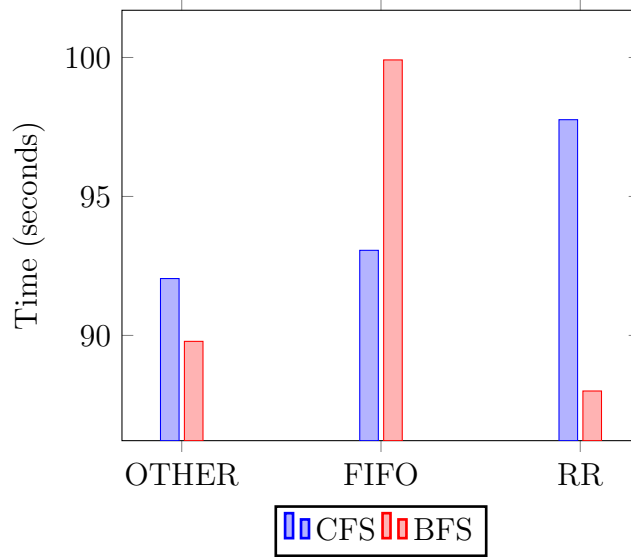
Execution Time of mixed process at Low Utilization



Execution Time of mixed process at Medium Utilization



Execution Time of mixed process at High Utilization



These results are slightly misleading, since I skewed the plots so that they don't start at 0. This makes the differences much more pronounced. Nevertheless, for almost every category, the BFS is much better. This is because it is far simpler, and less overhead is spent in the code of the scheduler itself. For typical programs (mixed), and the default scheduler (OTHER), we can see that the BFS is much more efficient. One thing I noticed, however, is that when I ran the simulations on my CFS scheduler, I could move my mouse around, and maybe open a window. When the BFS scheduler was used, the computer froze to a standstill until the script had finished. I have no idea why this is, but it was very noticeable.

5 Analysis

The results begin to paint a picture of what we should use and when. If we are looking to get many things done very quickly, we would be best served by using SCHED_OTHER. If, however, we want to get one thing done, very quickly, the best scheduler would seem to be FIFO. It all depends on our priorities.

If you are planning on using this for a home computer, the BFS is the clear winner. It is faster in every category, and the source is easy enough to read that you can understand and alter it to your needs.

6 Conclusion

The results clearly show that the BFS is by far the better scheduler, and the default scheduling scheme of both schedulers is better for everyday needs.

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

8.1 Execution Time of CFS

Schedule	Utilization	Type	Trial	Execution Time
SCHED_OTHER	Low	cpu	1	3.113322
			2	3.085774
			3	3.076218
			4	3.060629
			5	3.167129
			6	3.052559
			7	3.073442
			8	3.378702
			9	3.284676
			10	3.409316
			avg	3.170176
SCHED_OTHER	Low	io	1	4.858287
			2	5.200969
			3	5.428231
			4	4.881179
			5	4.905814
			6	4.968016
			7	4.945884
			8	4.644300
			9	5.012675
			10	4.865288
			avg	4.971064
SCHED_OTHER	Low	mixed	1	3.585013
			2	3.668794
			3	3.689240
			4	3.600797
			5	3.628582
			6	3.654732
			7	3.635530
			8	3.649710
			9	3.655495
			10	3.683408
			avg	3.645130

Schedule	Utilization	Type	Trial	Execution Time
SCHED_OTHER	Medium	cpu	1	16.546016
			2	16.723781
			3	16.827731
			4	16.770428
			5	17.079005
			6	17.568007
			7	17.338567
			8	17.359850
			9	17.489747
			10	17.679774
			avg	17.138290
SCHED_OTHER	Medium	io	1	23.229726
			2	23.032418
			3	23.035275
			4	24.173956
			5	23.249260
			6	23.627208
			7	23.411219
			8	23.826516
			9	24.132010
			10	23.787340
			avg	23.550492
SCHED_OTHER	Medium	mixed	1	17.923470
			2	18.240552
			3	17.608088
			4	18.732967
			5	18.818474
			6	18.518211
			7	18.774386
			8	21.456718
			9	19.925959
			10	20.103890
			avg	19.010271

Schedule	Utilization	Type	Trial	Execution Time
SCHED_OTHER	High	cpu	1	87.915865
			2	86.416415
			3	83.287822
			4	83.467525
			5	88.670789
			6	90.876241
			7	91.918730
			8	84.765072
			9	86.627548
			10	86.983002
			avg	87.092900
SCHED_OTHER	High	io	1	121.480541
			2	121.367239
			3	119.486197
			4	119.011534
			5	118.936303
			6	119.131862
			7	118.061361
			8	121.140718
			9	118.960390
			10	119.760408
			avg	119.733655
SCHED_OTHER	High	mixed	1	93.420030
			2	91.236862
			3	91.372184
			4	91.478747
			5	91.419111
			6	92.016583
			7	90.145052
			8	93.487565
			9	94.055634
			10	91.831055
			avg	92.046282

Schedule	Utilization	Type	Trial	Execution Time
SCHED_FIFO	Low	cpu	1	4.767669
			2	4.232951
			3	4.217194
			4	4.217439
			5	4.230748
			6	4.330329
			7	3.834218
			8	4.058068
			9	3.966241
			10	4.018254
			avg	4.187311
SCHED_FIFO	Low	io	1	5.741159
			2	5.877660
			3	5.609448
			4	5.110795
			5	5.605161
			6	5.701053
			7	5.698759
			8	5.266880
			9	5.538235
			10	5.928820
			avg	5.607797
SCHED_FIFO	Low	mixed	1	3.956731
			2	4.341535
			3	4.126415
			4	4.356061
			5	3.914087
			6	4.065533
			7	4.023859
			8	4.067056
			9	3.961594
			10	3.817489
			avg	4.063036

Schedule	Utilization	Type	Trial	Execution Time
SCHED_FIFO	Medium	cpu	1	17.507954
			2	18.265287
			3	17.984095
			4	18.353659
			5	17.427843
			6	17.686919
			7	18.021866
			8	18.969621
			9	17.487940
			10	18.741169
			avg	18.044635
SCHED_FIFO	Medium	io	1	28.380893
			2	28.056699
			3	27.722469
			4	27.918015
			5	27.557618
			6	27.956404
			7	28.168144
			8	27.764034
			9	27.210386
			10	27.678325
			avg	27.841298
SCHED_FIFO	Medium	mixed	1	19.873724
			2	20.103292
			3	20.331504
			4	19.724655
			5	19.790082
			6	20.189537
			7	20.268001
			8	20.216373
			9	19.964954
			10	19.924743
			avg	20.038686

Schedule	Utilization	Type	Trial	Execution Time
SCHED_FIFO	High	cpu	1	89.850884
			2	89.744659
			3	89.961705
			4	89.349951
			5	89.692762
			6	90.594082
			7	87.831015
			8	87.147024
			9	86.709310
			10	84.953540
			avg	88.583493
SCHED_FIFO	High	io	1	126.372693
			2	127.417443
			3	128.292543
			4	129.334590
			5	129.293336
			6	128.910176
			7	130.221451
			8	129.878459
			9	131.673224
			10	131.214761
			avg	129.260867
SCHED_FIFO	High	mixed	1	92.218250
			2	93.045968
			3	91.826259
			4	93.013395
			5	92.342805
			6	93.398109
			7	93.194504
			8	94.171661
			9	93.371145
			10	94.035168
			avg	93.061726

Schedule	Utilization	Type	Trial	Execution Time
SCHED_RR	Low	cpu	1	4.823887
			2	4.144268
			3	4.347092
			4	4.156300
			5	4.160639
			6	4.033034
			7	4.056237
			8	4.008037
			9	4.011384
			10	3.982087
			avg	4.172296
SCHED_RR	Low	io	1	5.198447
			2	6.489720
			3	5.499373
			4	5.340822
			5	5.851695
			6	5.234775
			7	4.992583
			8	5.726227
			9	5.142166
			10	5.103642
			avg	5.457945
SCHED_RR	Low	mixed	1	4.328107
			2	4.499245
			3	4.350500
			4	4.801048
			5	3.876172
			6	3.925147
			7	4.316064
			8	3.875015
			9	4.164466
			10	3.945480
			avg	4.208124

Schedule	Utilization	Type	Trial	Execution Time
SCHED_RR	Medium	cpu	1	16.857580
			2	17.342540
			3	18.258574
			4	16.647446
			5	18.546699
			6	17.116629
			7	17.041275
			8	18.089882
			9	17.098960
			10	18.305961
			avg	17.530554
SCHED_RR	Medium	io	1	26.414129
			2	26.476345
			3	26.401803
			4	27.046376
			5	26.531291
			6	26.114130
			7	26.550292
			8	26.431817
			9	26.748029
			10	26.171264
			avg	26.488547
SCHED_RR	Medium	mixed	1	18.661498
			2	19.635452
			3	19.520348
			4	19.894632
			5	20.114134
			6	19.728331
			7	19.951527
			8	19.040728
			9	18.179519
			10	21.020220
			avg	19.574638

Schedule	Utilization	Type	Trial	Execution Time
SCHED_RR	High	cpu	1	85.113127
			2	86.124189
			3	86.252366
			4	86.529106
			5	86.564893
			6	87.501662
			7	87.151608
			8	88.686935
			9	89.373674
			10	90.036917
			avg	87.333447
SCHED_RR	High	io	1	139.077522
			2	139.556447
			3	138.275238
			4	138.918804
			5	139.911959
			6	141.536998
			7	138.044715
			8	135.848934
			9	135.838639
			10	134.731781
			avg	138.174103
SCHED_RR	High	mixed	1	97.897644
			2	96.327284
			3	96.395055
			4	97.117333
			5	97.192371
			6	98.868089
			7	98.874825
			8	99.085297
			9	98.512640
			10	97.348903
			avg	97.761944

8.2 Response Time of CFS

Schedule	Utilization	Type	Trial	Response Time
SCHED_OTHER	Low	cpu	1	.029775
			2	.020091
			3	.012965
			4	.023557
			5	.011803
			6	.017302
			7	.010763
			8	.012556
			9	.018300
			10	.022464
			avg	.017957
SCHED_OTHER	Low	io	1	.025224
			2	.027931
			3	.022903
			4	.018454
			5	.016182
			6	.027825
			7	.025980
			8	.035748
			9	.035660
			10	.019295
			avg	.025520
SCHED_OTHER	Low	mixed	1	.018082
			2	.035731
			3	.035565
			4	.019533
			5	.021661
			6	.023291
			7	.037184
			8	.025106
			9	.026795
			10	.021998
			avg	.026495

Schedule	Utilization	Type	Trial	Response Time
SCHED_OTHER	Medium	cpu	1	.096281
			2	.110186
			3	.087933
			4	.089371
			5	.070637
			6	.089415
			7	.071433
			8	.103204
			9	.106489
			10	.084033
			avg	.090898
SCHED_OTHER	Medium	io	1	.084871
			2	.103185
			3	.086338
			4	.073281
			5	.058578
			6	.066008
			7	.089326
			8	.107627
			9	.082960
			10	.107375
			avg	.085955
SCHED_OTHER	Medium	mixed	1	.108302
			2	.128181
			3	.088555
			4	.096712
			5	.094228
			6	.068670
			7	.081678
			8	.108873
			9	.081419
			10	.054916
			avg	.091153

Schedule	Utilization	Type	Trial	Response Time
SCHED_OTHER	High	cpu	1	2.493044
			2	.487741
			3	.567370
			4	.598412
			5	.485264
			6	.667362
			7	.596231
			8	.601163
			9	.623526
			10	.490976
			avg	.761109
SCHED_OTHER	High	io	1	.591600
			2	.511346
			3	.518590
			4	.528305
			5	.589642
			6	.460307
			7	.460874
			8	.449338
			9	.418420
			10	.683177
			avg	.521160
SCHED_OTHER	High	mixed	1	.531096
			2	.504026
			3	.508656
			4	.501507
			5	.481601
			6	.558121
			7	.597610
			8	.488798
			9	.478473
			10	.424585
			avg	.507447

Schedule	Utilization	Type	Trial	Response Time
SCHED_FIFO	Low	cpu	1	1.788079
			2	2.339941
			3	1.856832
			4	1.859701
			5	1.803476
			6	1.902196
			7	1.829024
			8	1.931661
			9	1.843469
			10	1.907786
			avg	1.906217
SCHED_FIFO	Low	io	1	2.553619
			2	2.701657
			3	2.406873
			4	2.394868
			5	2.326282
			6	2.609726
			7	2.530085
			8	2.133407
			9	2.549455
			10	2.291884
			avg	2.449786
SCHED_FIFO	Low	mixed	1	2.403184
			2	2.177544
			3	1.780954
			4	1.729888
			5	1.672752
			6	2.448211
			7	2.345417
			8	1.935650
			9	1.815883
			10	2.020718
			avg	2.033020

Schedule	Utilization	Type	Trial	Response Time
SCHED_FIFO	Medium	cpu	1	8.613151
			2	9.160897
			3	10.484417
			4	8.984179
			5	9.169944
			6	9.057966
			7	10.102713
			8	10.870018
			9	9.769173
			10	10.589246
			avg	9.680170
SCHED_FIFO	Medium	io	1	13.748009
			2	13.928651
			3	13.806987
			4	14.222799
			5	13.299201
			6	13.532180
			7	13.262725
			8	13.562708
			9	13.671614
			10	13.564571
			avg	13.659945
SCHED_FIFO	Medium	mixed	1	9.592414
			2	9.938778
			3	10.002375
			4	9.617395
			5	9.828413
			6	10.279999
			7	9.671231
			8	10.031096
			9	10.348455
			10	10.441911
			avg	9.975207

Schedule	Utilization	Type	Trial	Response Time
SCHED_FIFO	High	cpu	1	45.517253
			2	45.354153
			3	46.363413
			4	46.307296
			5	46.163106
			6	44.983417
			7	45.472608
			8	45.936692
			9	46.284498
			10	46.247824
			avg	45.863026
SCHED_FIFO	High	io	1	70.812604
			2	70.728756
			3	71.161349
			4	72.767261
			5	73.285520
			6	74.607732
			7	73.902446
			8	75.820110
			9	75.868966
			10	74.709167
			avg	73.366391
SCHED_FIFO	High	mixed	1	56.389336
			2	56.805692
			3	55.967300
			4	74.243835
			5	57.075412
			6	51.209485
			7	50.081463
			8	51.864769
			9	50.586947
			10	51.266033
			avg	55.549027

Schedule	Utilization	Type	Trial	Response Time
SCHED_RR	Low	cpu	1	.447936
			2	.448944
			3	.448219
			4	.448722
			5	.451383
			6	.449617
			7	.450575
			8	.388915
			9	.441489
			10	.420055
			avg	.439585
SCHED_RR	Low	io	1	.450547
			2	.449053
			3	.441349
			4	.449703
			5	.448740
			6	.434772
			7	.449258
			8	.383733
			9	.451032
			10	.367309
			avg	.432550
SCHED_RR	Low	mixed	1	.493480
			2	.447926
			3	.434324
			4	.418243
			5	.413773
			6	.384451
			7	.382458
			8	.396684
			9	.380458
			10	.375885
			avg	.412768

Schedule	Utilization	Type	Trial	Response Time
SCHED_RR	Medium	cpu	1	2.326943
			2	2.176469
			3	2.302464
			4	2.359505
			5	2.245194
			6	2.145829
			7	2.345610
			8	2.358036
			9	2.357245
			10	2.452947
			avg	2.307024
SCHED_RR	Medium	io	1	2.564563
			2	2.308179
			3	2.308199
			4	2.665195
			5	2.292393
			6	2.254763
			7	2.253465
			8	2.275580
			9	2.258530
			10	3.026400
			avg	2.420727
SCHED_RR	Medium	mixed	1	2.377238
			2	2.294038
			3	2.451860
			4	2.231386
			5	2.409059
			6	2.398041
			7	2.355530
			8	2.211466
			9	2.411408
			10	2.211740
			avg	2.335176

Schedule	Utilization	Type	Trial	Response Time
SCHED_RR	High	cpu	1	11.082622
			2	11.063431
			3	11.827796
			4	11.499834
			5	11.217948
			6	11.356375
			7	11.504727
			8	11.309596
			9	11.357104
			10	11.362759
			avg	11.358219
SCHED_RR	High	io	1	11.615277
			2	11.562313
			3	10.917932
			4	11.875607
			5	11.687035
			6	12.915274
			7	11.539156
			8	11.556079
			9	11.434712
			10	11.719167
			avg	11.682255
SCHED_RR	High	mixed	1	11.480303
			2	11.245461
			3	11.322383
			4	11.409192
			5	11.552720
			6	11.349232
			7	11.545749
			8	11.496801
			9	11.596514
			10	11.393969
			avg	11.439232

8.3 Turnaround Time of CFS

Schedule	Utilization	Type	Trial	Turnaround Time
SCHED_OTHER	Low	cpu	1	2.683808
			2	2.845356
			3	2.682009
			4	2.828021
			5	2.814909
			6	2.722009
			7	2.693313
			8	2.655308
			9	2.734307
			10	2.987498
			avg	2.764654
SCHED_OTHER	Low	io	1	4.259679
			2	4.171792
			3	4.097003
			4	4.002966
			5	4.009730
			6	4.336636
			7	3.947332
			8	4.158953
			9	3.975034
			10	4.193978
			avg	4.115310
SCHED_OTHER	Low	mixed	1	3.031443
			2	2.914330
			3	2.905814
			4	3.050252
			5	2.994757
			6	2.941995
			7	3.003292
			8	3.097569
			9	3.036036
			10	3.072483
			avg	3.004797

Schedule	Utilization	Type	Trial	Turnaround Time
SCHED_OTHER	Medium	cpu	1	14.689355
			2	14.708221
			3	13.962657
			4	13.963776
			5	13.574216
			6	14.055479
			7	14.111121
			8	13.976513
			9	15.394558
			10	14.164990
			avg	14.260089
SCHED_OTHER	Medium	io	1	21.581018
			2	21.219423
			3	21.166271
			4	20.292891
			5	20.568618
			6	21.253086
			7	21.404320
			8	21.287798
			9	21.154360
			10	21.042521
			avg	21.097031
SCHED_OTHER	Medium	mixed	1	16.355885
			2	16.928453
			3	17.744174
			4	15.589540
			5	15.708228
			6	16.656403
			7	16.008487
			8	17.355543
			9	16.342353
			10	16.084520
			avg	16.477359

Schedule	Utilization	Type	Trial	Turnaround Time
SCHED_OTHER	High	cpu	1	74.440989
			2	77.096329
			3	77.597706
			4	80.943882
			5	77.195917
			6	78.004809
			7	84.718553
			8	91.611381
			9	90.321768
			10	95.813024
			avg	82.774436
SCHED_OTHER	High	io	1	138.650437
			2	134.465075
			3	138.252449
			4	144.518427
			5	143.772998
			6	140.871598
			7	128.915328
			8	124.283309
			9	128.156826
			10	129.348439
			avg	135.123488
SCHED_OTHER	High	mixed	1	98.930642
			2	100.290346
			3	97.213620
			4	107.413937
			5	106.990227
			6	106.750659
			7	106.587639
			8	104.429753
			9	107.608186
			10	106.846614
			avg	104.306162

Schedule	Utilization	Type	Trial	Turnaround Time
SCHED_FIFO	Low	cpu	1	.755385
			2	.754999
			3	.760403
			4	.799779
			5	.663137
			6	.729554
			7	.781372
			8	.771091
			9	.815302
			10	.653374
			avg	.748440
SCHED_FIFO	Low	io	1	1.049659
			2	.824301
			3	.928255
			4	1.111109
			5	.913183
			6	1.029675
			7	.849404
			8	.996157
			9	.788037
			10	.980059
			avg	.946984
SCHED_FIFO	Low	mixed	1	.712128
			2	.767637
			3	.860447
			4	.821943
			5	.671969
			6	.827172
			7	.815762
			8	.839716
			9	.858463
			10	.755406
			avg	.793064

Schedule	Utilization	Type	Trial	Turnaround Time
SCHED_FIFO	Medium	cpu	1	.774496
			2	.776368
			3	.826651
			4	.764645
			5	.793908
			6	.805739
			7	.785515
			8	.768447
			9	.794304
			10	.812700
			avg	.790277
SCHED_FIFO	Medium	io	1	1.095736
			2	1.105379
			3	1.117092
			4	1.097927
			5	1.111461
			6	1.096562
			7	1.106568
			8	1.115655
			9	1.121483
			10	1.131976
			avg	1.109984
SCHED_FIFO	Medium	mixed	1	.840144
			2	.887506
			3	.867735
			4	.880707
			5	.893843
			6	.856566
			7	.863410
			8	.839795
			9	.899442
			10	.870925
			avg	.870007

Schedule	Utilization	Type	Trial	Turnaround Time
SCHED_FIFO	High	cpu	1	.805434
			2	.807148
			3	.743464
			4	.711729
			5	.724211
			6	.750403
			7	.778896
			8	.778270
			9	.803921
			10	.754711
			avg	.765819
SCHED_FIFO	High	io	1	1.149876
			2	1.138526
			3	1.148598
			4	1.149755
			5	1.138787
			6	1.150127
			7	1.169862
			8	1.154688
			9	1.155832
			10	1.143194
			avg	1.149925
SCHED_FIFO	High	mixed	1	.832460
			2	.841731
			3	.835322
			4	.830543
			5	.847208
			6	.877371
			7	.873527
			8	.903167
			9	.918748
			10	.907841
			avg	.866792

Schedule	Utilization	Type	Trial	Turnaround Time
SCHED_RR	Low	cpu	1	3.958025
			2	3.429795
			3	3.484253
			4	2.892537
			5	2.747446
			6	2.756412
			7	3.198352
			8	4.601408
			9	2.673358
			10	3.177458
			avg	3.291904
SCHED_RR	Low	io	1	3.925899
			2	4.962669
			3	4.455024
			4	3.948698
			5	4.751469
			6	5.239481
			7	4.136852
			8	4.822852
			9	3.891441
			10	5.080505
			avg	4.521489
SCHED_RR	Low	mixed	1	4.715432
			2	3.564962
			3	3.625617
			4	3.353560
			5	2.840617
			6	2.708833
			7	3.268703
			8	4.318927
			9	3.605347
			10	3.397312
			avg	3.539931

Schedule	Utilization	Type	Trial	Turnaround Time
SCHED_RR	Medium	cpu	1	11.462400
			2	11.734749
			3	12.011819
			4	12.772976
			5	12.162775
			6	11.767298
			7	12.224796
			8	12.333258
			9	11.858432
			10	12.034425
			avg	12.036293
SCHED_RR	Medium	io	1	19.608245
			2	17.914993
			3	17.439505
			4	17.106413
			5	18.251606
			6	17.584194
			7	17.620449
			8	16.879894
			9	17.812836
			10	18.375035
			avg	17.859317
SCHED_RR	Medium	mixed	1	12.159948
			2	13.158809
			3	11.601103
			4	12.943573
			5	12.350547
			6	12.408756
			7	11.417541
			8	12.087923
			9	12.352870
			10	12.356233
			avg	12.283730

Schedule	Utilization	Type	Trial	Turnaround Time
SCHED_RR	High	cpu	1	50.495704
			2	50.720685
			3	49.952416
			4	49.832446
			5	50.255277
			6	49.785087
			7	50.594825
			8	50.501449
			9	52.983516
			10	58.054662
			avg	51.317607
SCHED_RR	High	io	1	87.252550
			2	81.615423
			3	83.300093
			4	83.825166
			5	85.502920
			6	83.471720
			7	81.217935
			8	82.323837
			9	82.956598
			10	83.187958
			avg	83.465420
SCHED_RR	High	mixed	1	57.912147
			2	56.021378
			3	56.234628
			4	56.355909
			5	57.286122
			6	56.541501
			7	56.492660
			8	55.772067
			9	57.732711
			10	58.141057
			avg	56.849018

8.4 Execution Time of BFS

Schedule	Utilization	Type	Trial	Execution Time
SCHED_OTHER	Low	cpu	1	3.050476
			2	3.105130
			3	3.078327
			4	3.111985
			5	3.128143
			6	3.072700
			7	3.107441
			8	3.127176
			9	3.106832
			10	3.114357
			avg	3.100256
SCHED_OTHER	Low	io	1	3.161187
			2	3.276934
			3	3.276495
			4	3.244901
			5	3.236139
			6	3.419036
			7	3.299788
			8	3.315823
			9	3.453758
			10	3.222414
			avg	3.290647
SCHED_OTHER	Low	mixed	1	3.541285
			2	3.428980
			3	3.375245
			4	3.481335
			5	3.417114
			6	3.373077
			7	3.560955
			8	3.301487
			9	3.403257
			10	3.285314
			avg	3.416804

Schedule	Utilization	Type	Trial	Execution Time
SCHED_OTHER	Medium	cpu	1	15.854477
			2	15.775794
			3	16.113107
			4	16.058427
			5	15.878714
			6	15.972906
			7	15.832716
			8	15.961680
			9	15.678609
			10	15.545405
			avg	15.867183
SCHED_OTHER	Medium	io	1	17.012918
			2	16.836760
			3	17.089329
			4	16.745066
			5	16.586207
			6	16.539349
			7	16.790100
			8	16.909341
			9	16.703020
			10	16.975618
			avg	16.818770
SCHED_OTHER	Medium	mixed	1	17.014851
			2	16.993121
			3	17.196389
			4	17.469941
			5	17.101072
			6	17.552146
			7	16.945087
			8	17.633391
			9	16.845377
			10	17.552058
			avg	17.230343

Schedule	Utilization	Type	Trial	Execution Time
SCHED_OTHER	High	cpu	1	81.506915
			2	81.078191
			3	81.831499
			4	81.617476
			5	81.191212
			6	82.032392
			7	82.423082
			8	81.816409
			9	81.994060
			10	82.027391
			avg	81.751862
SCHED_OTHER	High	io	1	87.236685
			2	87.129539
			3	87.374060
			4	87.170068
			5	88.693243
			6	88.014258
			7	88.741125
			8	90.818835
			9	87.281841
			10	91.109499
			avg	88.356915
SCHED_OTHER	High	mixed	1	87.804176
			2	90.809356
			3	90.282483
			4	89.473266
			5	88.344753
			6	91.236950
			7	89.821557
			8	90.263133
			9	90.136362
			10	89.690332
			avg	89.786236

Schedule	Utilization	Type	Trial	Execution Time
SCHED_FIFO	Low	cpu	1	3.603275
			2	3.630984
			3	3.328510
			4	3.239451
			5	3.319335
			6	3.645505
			7	3.565032
			8	3.699188
			9	3.547206
			10	3.498191
			avg	3.507667
SCHED_FIFO	Low	io	1	3.855346
			2	3.549734
			3	3.627153
			4	4.265454
			5	3.287910
			6	3.496590
			7	3.488063
			8	3.627615
			9	3.575925
			10	3.454624
			avg	3.622841
SCHED_FIFO	Low	mixed	1	4.010396
			2	3.776258
			3	3.809140
			4	3.682535
			5	3.825809
			6	3.794203
			7	3.991235
			8	3.783310
			9	3.666232
			10	3.813138
			avg	3.815225

Schedule	Utilization	Type	Trial	Execution Time
SCHED_FIFO	Medium	cpu	1	16.099366
			2	16.560516
			3	17.403689
			4	17.275705
			5	17.252448
			6	17.453940
			7	17.315604
			8	17.208000
			9	17.230861
			10	17.827863
			avg	17.162799
SCHED_FIFO	Medium	io	1	18.511190
			2	18.600525
			3	17.960715
			4	18.218530
			5	19.705208
			6	18.967103
			7	20.284493
			8	19.574449
			9	19.029583
			10	19.326024
			avg	19.017782
SCHED_FIFO	Medium	mixed	1	18.318534
			2	20.533580
			3	19.587665
			4	20.397149
			5	19.140760
			6	20.275054
			7	21.039787
			8	19.772468
			9	19.912595
			10	21.865973
			avg	20.084356

Schedule	Utilization	Type	Trial	Execution Time
SCHED_FIFO	High	cpu	1	90.071385
			2	83.504283
			3	81.979106
			4	89.597640
			5	86.297950
			6	85.765636
			7	86.016167
			8	91.791243
			9	91.438396
			10	90.323764
			avg	87.678557
SCHED_FIFO	High	io	1	92.470460
			2	92.916520
			3	94.023782
			4	97.198504
			5	99.477202
			6	100.750224
			7	98.028396
			8	98.231702
			9	95.970876
			10	97.929062
			avg	96.699672
SCHED_FIFO	High	mixed	1	95.383766
			2	97.113729
			3	98.592690
			4	97.826817
			5	101.060273
			6	98.028809
			7	102.005094
			8	107.865629
			9	102.945106
			10	98.300617
			avg	99.912253

Schedule	Utilization	Type	Trial	Execution Time
SCHED_RR	Low	cpu	1	3.875441
			2	3.456891
			3	3.716245
			4	3.632656
			5	4.092471
			6	3.589672
			7	3.718323
			8	4.163718
			9	3.367274
			10	3.466300
			avg	3.707899
SCHED_RR	Low	io	1	4.363085
			2	3.845227
			3	3.849536
			4	3.926205
			5	4.651334
			6	3.840403
			7	3.772023
			8	3.743614
			9	3.817757
			10	4.012979
			avg	3.982216
SCHED_RR	Low	mixed	1	4.482845
			2	3.816849
			3	3.759525
			4	3.789929
			5	4.787760
			6	3.872292
			7	4.217242
			8	4.237890
			9	4.368774
			10	4.419292
			avg	4.175239

Schedule	Utilization	Type	Trial	Execution Time
SCHED_RR	Medium	cpu	1	17.863797
			2	18.405668
			3	17.942015
			4	18.098654
			5	18.205578
			6	18.036021
			7	17.373177
			8	17.408317
			9	18.033012
			10	18.178475
			avg	17.954471
SCHED_RR	Medium	io	1	19.890356
			2	19.251160
			3	19.313216
			4	19.431954
			5	20.594353
			6	18.939111
			7	19.515339
			8	19.725853
			9	19.108904
			10	21.803579
			avg	19.757382
SCHED_RR	Medium	mixed	1	19.751690
			2	19.820719
			3	19.630981
			4	19.106144
			5	19.783981
			6	19.119501
			7	18.734108
			8	19.155539
			9	19.770186
			10	18.749857
			avg	19.362270

Schedule	Utilization	Type	Trial	Execution Time
SCHED_RR	High	cpu	1	87.506789
			2	87.421710
			3	87.176070
			4	86.753762
			5	86.289147
			6	86.712578
			7	88.071883
			8	88.145742
			9	87.507202
			10	86.917228
			avg	87.250211
SCHED_RR	High	io	1	92.310844
			2	96.490035
			3	96.744996
			4	98.010646
			5	93.521493
			6	101.165408
			7	96.391161
			8	96.584936
			9	98.482611
			10	100.098232
			avg	96.980036
SCHED_RR	High	mixed	1	86.986053
			2	85.614611
			3	86.193241
			4	87.799907
			5	87.891323
			6	86.403160
			7	89.352697
			8	88.573268
			9	90.956802
			10	90.227689
			avg	87.999875

9 Appendix B - Description of Files

pa3.c - This is the test code that will evaluate the performance of the various scheduling mechanisms. Usage will be printed if the function is called with incorrect parameters, or you can read this document for usage.

pa3.pdf - This is the thing you are reading right now.

pa3.tex - This is the code that generated this report.

extime.sh - The script that will test the execution time of all schedulers.

restime.sh - The script that will test response time.

turntime.sh - the script that will test turnaround time.

Makefile - this will compile the c code. The entire process is as follows: make, then run a script, then read stdout.