## Abstract

This report contains information about a series of tests that compare three of the scheduling schemes for the Linux operating system. Each test was conducted with varying amounts of identical processes in an isolated testing environment. The collected data shows the correlations between the scheduling scheme, the number of processes and the type of computation performed.

## Introduction

## Method

In order to collect data on for these tests, I edited some code written by Andy Sayler for a previous CSCI 3753 course to allow for each program to run with different scheduling schemes and with a different number of processes. I also created a file from the pi-sched.c file provided that does the same thing with the added ability to write the calculated value of pi into the /dev/null device file native to the system. This acts as a sort of *mixed* computation program for testing.

Once all of the programs were written, I edited the included script to run each of the programs with a specific scheduling scheme and number of processes. I also wrote a script which ran the previously mentioned script with a given input (the number of processes) so that the whole process of running the tests could be automated. The two scripts combined then write data to nine .csv files, each each corresponding to a different type of computation and a different scheduling scheme. Once all the data was collected, I used Microsoft Excel to construct graphs of each type of output so that I could make reasonable assumptions about the effects of the scheduling scheme and number of processes on the programs’ execution.

## Analysis

## Conclusion

## References

I used the code included in the assignment zip as a start or this assignment. I added functionality to each to fit the full purpose of this assignment. Each edited file contains the original file information as well as comments that state my using of the file.

## Appendix A

Compute Bound FIFO

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Trial | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 1 | 4.15 | 4.14 | 0 | 99% | 5 | 4 |
| 2 | 1 | 4.23 | 4.22 | 0.01 | 99% | 12 | 4 |
| 3 | 1 | 4.1 | 4.09 | 0.01 | 99% | 6 | 4 |
| AVG | 1 | 4.165 | 4.155 | 0.01 | 0.99 | 9 | 4 |
| 1 | 5 | 12.99 | 20.93 | 0.05 | 161% | 21 | 10 |
| 2 | 5 | 12.64 | 20.48 | 0 | 162% | 20 | 10 |
| 3 | 5 | 12.6 | 20.55 | 0.02 | 163% | 20 | 10 |
| AVG | 5 | 12.62 | 20.515 | 0.01 | 1.625 | 20 | 10 |
| 1 | 10 | 21.82 | 41.52 | 0.03 | 190% | 57 | 14 |
| 2 | 10 | 21.97 | 41.65 | 0.04 | 189% | 49 | 15 |
| 3 | 10 | 22.19 | 42.18 | 0.01 | 190% | 49 | 15 |
| AVG | 10 | 22.08 | 41.915 | 0.025 | 1.895 | 49 | 15 |
| 1 | 20 | 44.33 | 84.22 | 0.01 | 190% | 94 | 25 |
| 2 | 20 | 44.28 | 84.18 | 0.01 | 190% | 92 | 25 |
| 3 | 20 | 44.26 | 84.24 | 0.01 | 190% | 93 | 25 |
| AVG | 20 | 44.27 | 84.21 | 0.01 | 1.9 | 92.5 | 25 |
| 1 | 50 | 110.91 | 210.52 | 0.02 | 189% | 226 | 55 |
| 2 | 50 | 110.7 | 210.17 | 0.04 | 189% | 239 | 55 |
| 3 | 50 | 110.21 | 209.33 | 0.05 | 189% | 235 | 55 |
| AVG | 50 | 110.455 | 209.75 | 0.045 | 1.89 | 237 | 55 |
| 1 | 100 | 221.5 | 421.02 | 0.06 | 190% | 447 | 105 |
| 2 | 100 | 222.54 | 422.8 | 0.03 | 190% | 450 | 105 |
| 3 | 100 | 222.1 | 422.04 | 0.06 | 190% | 448 | 105 |
| AVG | 100 | 222.32 | 422.42 | 0.045 | 1.9 | 449 | 105 |
| 1 | 200 | 439.67 | 835.52 | 0.09 | 190% | 883 | 205 |
| 2 | 200 | 443.19 | 842.41 | 0.08 | 190% | 898 | 205 |
| AVG | 200 | 441.43 | 838.965 | 0.085 | 1.9 | 890.5 | 205 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
|  | 1 | 4.165 | 4.155 | 0.01 | 0.99 | 9 | 4 |
|  | 5 | 12.62 | 20.515 | 0.01 | 1.625 | 20 | 10 |
|  | 10 | 22.08 | 41.915 | 0.025 | 1.895 | 49 | 15 |
|  | 20 | 44.27 | 84.21 | 0.01 | 1.9 | 92.5 | 25 |
|  | 50 | 110.455 | 209.75 | 0.045 | 1.89 | 237 | 55 |
|  | 100 | 222.32 | 422.42 | 0.045 | 1.9 | 449 | 105 |
|  | 200 | 441.43 | 838.965 | 0.085 | 1.9 | 890.5 | 205 |

Compute Bound RR

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Trial | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 1 | 4.14 | 4.14 | 0 | 99% | 6 | 4 |
| 2 | 1 | 4.22 | 4.2 | 0.02 | 99% | 8 | 4 |
| 3 | 1 | 4.23 | 4.22 | 0.01 | 99% | 5 | 4 |
| AVG | 1 | 4.225 | 4.21 | 0.015 | 0.99 | 6.5 | 4 |
| 1 | 5 | 11.83 | 22.48 | 0.02 | 190% | 126 | 10 |
| 2 | 5 | 10.78 | 20.55 | 0.01 | 190% | 116 | 10 |
| 3 | 5 | 10.78 | 20.49 | 0.04 | 190% | 116 | 10 |
| AVG | 5 | 10.78 | 20.52 | 0.025 | 1.9 | 116 | 10 |
| 1 | 10 | 21.73 | 41.24 | 0.01 | 189% | 253 | 14 |
| 2 | 10 | 22.3 | 42.34 | 0.02 | 189% | 263 | 18 |
| 3 | 10 | 22.22 | 42.24 | 0 | 190% | 262 | 18 |
| AVG | 10 | 22.26 | 42.29 | 0.01 | 1.895 | 262.5 | 18 |
| 1 | 20 | 44.27 | 84.22 | 0.02 | 190% | 521 | 38 |
| 2 | 20 | 44.05 | 83.81 | 0 | 190% | 515 | 30 |
| 3 | 20 | 44.19 | 84.03 | 0 | 190% | 525 | 25 |
| AVG | 20 | 44.12 | 83.92 | 0 | 1.9 | 520 | 27.5 |
| 1 | 50 | 110.75 | 210.64 | 0.02 | 190% | 1314 | 85 |
| 2 | 50 | 110.56 | 210.31 | 0.02 | 190% | 1305 | 61 |
| 3 | 50 | 110.72 | 210.64 | 0.04 | 190% | 1298 | 77 |
| AVG | 50 | 110.64 | 210.475 | 0.03 | 1.9 | 1301.5 | 69 |
| 1 | 100 | 220.92 | 420.17 | 0.05 | 190% | 2633 | 156 |
| 2 | 100 | 222.07 | 422.34 | 0.06 | 190% | 2603 | 106 |
| 3 | 100 | 221.54 | 421.32 | 0.04 | 190% | 2616 | 161 |
| AVG | 100 | 221.805 | 421.83 | 0.05 | 1.9 | 2609.5 | 133.5 |
| 1 | 200 | 442.58 | 841.6 | 0.15 | 190% | 5229 | 334 |
| 2 | 200 | 443.63 | 843.54 | 0.14 | 190% | 5251 | 257 |
| AVG | 200 | 443.105 | 842.57 | 0.145 | 1.9 | 5240 | 295.5 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
|  | 1 | 4.225 | 4.21 | 0.015 | 0.99 | 6.5 | 4 |
|  | 5 | 10.78 | 20.52 | 0.025 | 1.9 | 116 | 10 |
|  | 10 | 22.26 | 42.29 | 0.01 | 1.895 | 262.5 | 18 |
|  | 20 | 44.12 | 83.92 | 0 | 1.9 | 520 | 27.5 |
|  | 50 | 110.64 | 210.475 | 0.03 | 1.9 | 1301.5 | 69 |
|  | 100 | 221.805 | 421.83 | 0.05 | 1.9 | 2609.5 | 133.5 |
|  | 200 | 443.105 | 842.57 | 0.145 | 1.9 | 5240 | 295.5 |

Compute Bound OTHER

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Trial | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 1 | 4.18 | 4.08 | 0.06 | 98% | 243 | 1 |
| 2 | 1 | 4.21 | 4.09 | 0.08 | 99% | 286 | 1 |
| 3 | 1 | 4.1 | 3.95 | 0.08 | 98% | 238 | 1 |
| AVG | 1 | 4.155 | 4.02 | 0.08 | 0.985 | 262 | 1 |
| 1 | 5 | 11.16 | 20.91 | 0.3 | 190% | 4163 | 7 |
| 2 | 5 | 10.51 | 20.54 | 0.1 | 196% | 3379 | 7 |
| 3 | 5 | 10.77 | 20.59 | 0.16 | 192% | 3648 | 9 |
| AVG | 5 | 10.64 | 20.565 | 0.13 | 1.94 | 3513.5 | 8 |
| 1 | 10 | 21.18 | 41.02 | 0.24 | 194% | 11325 | 18 |
| 2 | 10 | 21.46 | 41.36 | 0.28 | 194% | 11309 | 18 |
| 3 | 10 | 21.36 | 41.08 | 0.39 | 194% | 11269 | 12 |
| AVG | 10 | 21.41 | 41.22 | 0.335 | 1.94 | 11289 | 15 |
| 1 | 20 | 42.42 | 83.07 | 0.41 | 196% | 22753 | 23 |
| 2 | 20 | 42.87 | 83.94 | 0.36 | 196% | 22969 | 20 |
| 3 | 20 | 41.71 | 81.56 | 0.46 | 196% | 22570 | 31 |
| AVG | 20 | 42.29 | 82.75 | 0.41 | 1.96 | 22769.5 | 25.5 |
| 1 | 50 | 104.34 | 206.16 | 0.65 | 198% | 56400 | 99 |
| 2 | 50 | 104.96 | 207.08 | 0.64 | 197% | 56392 | 87 |
| 3 | 50 | 103.69 | 204.8 | 0.62 | 198% | 55711 | 85 |
| AVG | 50 | 104.325 | 205.94 | 0.63 | 1.975 | 56051.5 | 86 |
| 1 | 100 | 205.13 | 406.71 | 0.89 | 198% | 109630 | 194 |
| 2 | 100 | 204.1 | 404.78 | 0.78 | 198% | 109068 | 152 |
| 3 | 100 | 206.35 | 409.17 | 0.92 | 198% | 110451 | 105 |
| AVG | 100 | 205.225 | 406.975 | 0.85 | 1.98 | 109759.5 | 128.5 |
| 1 | 200 | 412.91 | 820.31 | 1.08 | 198% | 220030 | 327 |
| 2 | 200 | 412.12 | 819.03 | 1.31 | 199% | 220703 | 229 |
| AVG | 200 | 412.515 | 819.67 | 1.195 | 1.985 | 220366.5 | 278 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
|  | 1 | 4.155 | 4.02 | 0.08 | 0.985 | 262 | 1 |
|  | 5 | 10.64 | 20.565 | 0.13 | 1.94 | 3513.5 | 8 |
|  | 10 | 21.41 | 41.22 | 0.335 | 1.94 | 11289 | 15 |
|  | 20 | 42.29 | 82.75 | 0.41 | 1.96 | 22769.5 | 25.5 |
|  | 50 | 104.325 | 205.94 | 0.63 | 1.975 | 56051.5 | 86 |
|  | 100 | 205.225 | 406.975 | 0.85 | 1.98 | 109759.5 | 128.5 |
|  | 200 | 412.515 | 819.67 | 1.195 | 1.985 | 220366.5 | 278 |

IO Bound FIFO

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Trial | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 1 | 12.41 | 0 | 1.33 | 10% | 1287 | 13637 |
| 2 | 1 | 7.28 | 0 | 1.21 | 16% | 1980 | 13443 |
| 3 | 1 | 8.25 | 0 | 1.34 | 16% | 3667 | 13419 |
| AVG | 1 | 7.765 | 0 | 1.275 | 0.16 | 2823.5 | 13431 |
| 1 | 5 | 7.23 | 0 | 1.5 | 20% | 3266 | 12572 |
| 2 | 5 | 2 | 0 | 0.88 | 44% | 4711 | 12475 |
| 3 | 5 | 4.33 | 0 | 1.12 | 25% | 2483 | 12863 |
| AVG | 5 | 3.165 | 0 | 1 | 0.345 | 3597 | 12669 |
| 1 | 10 | 5.82 | 0 | 1.3 | 22% | 4046 | 12642 |
| 2 | 10 | 3.32 | 0 | 1 | 30% | 3695 | 12335 |
| 3 | 10 | 4.5 | 0 | 1.12 | 24% | 3541 | 12476 |
| AVG | 10 | 3.91 | 0 | 1.06 | 0.27 | 3618 | 12405.5 |
| 1 | 20 | 3.25 | 0 | 1.02 | 31% | 3690 | 12515 |
| 2 | 20 | 3.79 | 0 | 1.04 | 27% | 2496 | 12728 |
| 3 | 20 | 3.16 | 0 | 0.99 | 31% | 4783 | 12224 |
| AVG | 20 | 3.475 | 0 | 1.015 | 0.29 | 3639.5 | 12476 |
| 1 | 50 | 4.64 | 0 | 1.15 | 24% | 3582 | 12586 |
| 2 | 50 | 3.45 | 0 | 0.98 | 28% | 3536 | 12442 |
| 3 | 50 | 4.58 | 0 | 1.08 | 23% | 3691 | 12445 |
| AVG | 50 | 4.015 | 0 | 1.03 | 0.255 | 3613.5 | 12443.5 |
| 1 | 100 | 4.25 | 0 | 1.11 | 26% | 3424 | 12661 |
| 2 | 100 | 5.42 | 0 | 1.25 | 23% | 3656 | 12533 |
| 3 | 100 | 3.26 | 0 | 1.02 | 31% | 3878 | 12606 |
| AVG | 100 | 4.34 | 0 | 1.135 | 0.27 | 3767 | 12569.5 |
| 1 | 200 | 3.21 | 0 | 1 | 31% | 3544 | 12473 |
| 2 | 200 | 1.42 | 0 | 0.89 | 62% | 4510 | 12364 |
| AVG | 200 | 2.315 | 0 | 0.945 | 0.465 | 4027 | 12418.5 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
|  | 1 | 7.765 | 0 | 1.275 | 0.16 | 2823.5 | 13431 |
|  | 5 | 3.165 | 0 | 1 | 0.345 | 3597 | 12669 |
|  | 10 | 3.91 | 0 | 1.06 | 0.27 | 3618 | 12405.5 |
|  | 20 | 3.475 | 0 | 1.015 | 0.29 | 3639.5 | 12476 |
|  | 50 | 4.015 | 0 | 1.03 | 0.255 | 3613.5 | 12443.5 |
|  | 100 | 4.34 | 0 | 1.135 | 0.27 | 3767 | 12569.5 |
|  | 200 | 2.315 | 0 | 0.945 | 0.465 | 4027 | 12418.5 |

IO Bound RR

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Trial | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 1 | 15.65 | 0 | 1.34 | 8% | 1569 | 13455 |
| 2 | 1 | 14.82 | 0 | 1.5 | 10% | 2000 | 13345 |
| 3 | 1 | 9.28 | 0 | 1.26 | 13% | 2366 | 12976 |
| AVG | 1 | 13.25 | 0 | 1.366667 | 0.103333 | 1978.333 | 13258.67 |
| 1 | 5 | 6.53 | 0 | 1.24 | 19% | 3139 | 12673 |
| 2 | 5 | 9.03 | 0 | 1.3 | 14% | 3366 | 12472 |
| 3 | 5 | 6.38 | 0 | 1.38 | 21% | 4092 | 12572 |
| AVG | 5 | 7.313333 | 0 | 1.306667 | 0.18 | 3532.333 | 12572.33 |
| 1 | 10 | 5.04 | 0 | 1.1 | 22% | 3957 | 12401 |
| 2 | 10 | 5.92 | 0 | 1.27 | 21% | 2963 | 12694 |
| 3 | 10 | 7.72 | 0 | 1.38 | 17% | 2475 | 12793 |
| AVG | 10 | 6.226667 | 0 | 1.25 | 0.2 | 3131.667 | 12629.33 |
| 1 | 20 | 7.58 | 0 | 1.38 | 18% | 2928 | 12867 |
| 2 | 20 | 7.39 | 0 | 1.44 | 19% | 2931 | 12671 |
| 3 | 20 | 9.47 | 0 | 1.47 | 15% | 1921 | 12868 |
| AVG | 20 | 8.146667 | 0 | 1.43 | 0.173333 | 2593.333 | 12802 |
| 1 | 50 | 5.08 | 0 | 1.18 | 23% | 3318 | 12636 |
| 2 | 50 | 7.37 | 0 | 1.24 | 16% | 2486 | 12817 |
| 3 | 50 | 5.07 | 0 | 1.13 | 22% | 2984 | 12527 |
| AVG | 50 | 5.84 | 0 | 1.183333 | 0.203333 | 2929.333 | 12660 |
| 1 | 100 | 5.75 | 0 | 1.2 | 20% | 2542 | 12796 |
| 2 | 100 | 4.95 | 0 | 1.12 | 22% | 3976 | 12358 |
| 3 | 100 | 6.69 | 0 | 1.44 | 21% | 2897 | 12619 |
| AVG | 100 | 5.796667 | 0 | 1.253333 | 0.21 | 3138.333 | 12591 |
| 1 | 200 | 3.23 | 0 | 0.99 | 30% | 2945 | 12581 |
| 2 | 200 | 7.9 | 0 | 1.46 | 18% | 2359 | 12939 |
| AVG | 200 | 5.565 | 0 | 1.225 | 0.24 | 2652 | 12760 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
|  | 1 | 13.25 | 0 | 1.366667 | 0.103333 | 1978.333 | 13258.67 |
|  | 5 | 7.313333 | 0 | 1.306667 | 0.18 | 3532.333 | 12572.33 |
|  | 10 | 6.226667 | 0 | 1.25 | 0.2 | 3131.667 | 12629.33 |
|  | 20 | 8.146667 | 0 | 1.43 | 0.173333 | 2593.333 | 12802 |
|  | 50 | 5.84 | 0 | 1.183333 | 0.203333 | 2929.333 | 12660 |
|  | 100 | 5.796667 | 0 | 1.253333 | 0.21 | 3138.333 | 12591 |
|  | 200 | 5.565 | 0 | 1.225 | 0.24 | 2652 | 12760 |

IO Bound OTHER

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Trial | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 1 | 13.96 | 0 | 1.45 | 10% | 1611 | 13747 |
| 2 | 1 | 8.37 | 0 | 1.27 | 15% | 2782 | 13073 |
| 3 | 1 | 7.34 | 0 | 1.19 | 16% | 3494 | 13650 |
| AVG | 1 | 9.89 | 0 | 1.303333 | 0.136667 | 2629 | 13490 |
| 1 | 5 | 3.91 | 0 | 1.24 | 31% | 3046 | 12774 |
| 2 | 5 | 3.91 | 0 | 1.16 | 29% | 3072 | 12775 |
| 3 | 5 | 2.5 | 0 | 0.89 | 35% | 2961 | 12601 |
| AVG | 5 | 3.44 | 0 | 1.096667 | 0.316667 | 3026.333 | 12716.67 |
| 1 | 10 | 1.63 | 0 | 0.88 | 53% | 3562 | 12466 |
| 2 | 10 | 2.09 | 0 | 0.87 | 41% | 4059 | 12258 |
| 3 | 10 | 1.53 | 0 | 0.83 | 54% | 2817 | 12689 |
| AVG | 10 | 1.75 | 0 | 0.86 | 0.493333 | 3479.333 | 12471 |
| 1 | 20 | 1.87 | 0 | 0.84 | 45% | 3152 | 12437 |
| 2 | 20 | 3.17 | 0 | 0.95 | 30% | 3085 | 12559 |
| 3 | 20 | 1.88 | 0 | 0.84 | 44% | 3772 | 12404 |
| AVG | 20 | 2.306667 | 0 | 0.876667 | 0.396667 | 3336.333 | 12466.67 |
| 1 | 50 | 2.58 | 0 | 0.86 | 33% | 3085 | 12539 |
| 2 | 50 | 2.2 | 0 | 0.85 | 38% | 4399 | 12303 |
| 3 | 50 | 1.87 | 0 | 0.8 | 43% | 4147 | 12175 |
| AVG | 50 | 2.216667 | 0 | 0.836667 | 0.38 | 3877 | 12339 |
| 1 | 100 | 1.82 | 0 | 0.82 | 44% | 4431 | 12241 |
| 2 | 100 | 2.43 | 0 | 0.92 | 38% | 4404 | 12251 |
| 3 | 100 | 1.54 | 0 | 0.83 | 54% | 3649 | 12472 |
| AVG | 100 | 1.93 | 0 | 0.856667 | 0.453333 | 4161.333 | 12321.33 |
| 1 | 200 | 1.81 | 0 | 0.84 | 46% | 4404 | 12339 |
| 2 | 200 | 1.38 | 0 | 0.93 | 67% | 2086 | 12932 |
| AVG | 200 | 1.595 | 0 | 0.885 | 0.565 | 3245 | 12635.5 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
|  | 1 | 9.89 | 0 | 1.303333 | 0.136667 | 2629 | 13490 |
|  | 5 | 3.44 | 0 | 1.096667 | 0.316667 | 3026.333 | 12716.67 |
|  | 10 | 1.75 | 0 | 0.86 | 0.493333 | 3479.333 | 12471 |
|  | 20 | 2.306667 | 0 | 0.876667 | 0.396667 | 3336.333 | 12466.67 |
|  | 50 | 2.216667 | 0 | 0.836667 | 0.38 | 3877 | 12339 |
|  | 100 | 1.93 | 0 | 0.856667 | 0.453333 | 4161.333 | 12321.33 |
|  | 200 | 1.595 | 0 | 0.885 | 0.565 | 3245 | 12635.5 |

Mixed FIFO

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Trial | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 1 | 10.74 | 10.63 | 0.1 | 99% | 10 | 4 |
| 2 | 1 | 10.48 | 10.4 | 0.07 | 99% | 19 | 4 |
| 3 | 1 | 10.49 | 10.43 | 0.05 | 99% | 12 | 4 |
| AVG | 1 | 10.57 | 10.48667 | 0.073333 | 0.99 | 13.66667 | 4 |
| 1 | 5 | 32.78 | 52.72 | 0.19 | 161% | 51 | 10 |
| 2 | 5 | 32.37 | 52.39 | 0.14 | 162% | 51 | 10 |
| 3 | 5 | 32.25 | 51.76 | 0.25 | 161% | 48 | 10 |
| AVG | 5 | 32.46667 | 52.29 | 0.193333 | 1.613333 | 50 | 10 |
| 1 | 10 | 55.12 | 104.45 | 0.47 | 190% | 113 | 15 |
| 2 | 10 | 55.51 | 104.75 | 0.34 | 189% | 113 | 15 |
| 3 | 10 | 56.52 | 106.76 | 0.35 | 189% | 118 | 15 |
| AVG | 10 | 55.71667 | 105.32 | 0.386667 | 1.893333 | 114.6667 | 15 |
| 1 | 20 | 111.79 | 211.93 | 0.62 | 190% | 226 | 25 |
| 2 | 20 | 112.43 | 213.11 | 0.74 | 190% | 230 | 25 |
| 3 | 20 | 112.48 | 213.12 | 0.58 | 189% | 228 | 25 |
| AVG | 20 | 112.2333 | 212.72 | 0.646667 | 1.896667 | 228 | 25 |
| 1 | 50 | 280.23 | 530.58 | 1.44 | 189% | 561 | 55 |
| 2 | 50 | 276.5 | 523.95 | 1.53 | 190% | 552 | 55 |
| 3 | 50 | 280.41 | 531.87 | 1.5 | 190% | 565 | 55 |
| AVG | 50 | 279.0467 | 528.8 | 1.49 | 1.896667 | 559.3333 | 55 |
| 1 | 100 | 558.1 | 1057.78 | 3 | 190% | 1127 | 105 |
| 2 | 100 | 560.78 | 1062.1 | 3.3 | 189% | 1123 | 105 |
| 3 | 100 | 561.52 | 1064.28 | 3.03 | 190% | 1126 | 105 |
| AVG | 100 | 560.1333 | 1061.387 | 3.11 | 1.896667 | 1125.333 | 105 |
| 1 | 200 | 1121.65 | 2126.49 | 5.96 | 190% | 2246 | 205 |
| 2 | 200 | 1123.33 | 2126.96 | 6.11 | 189% | 2251 | 205 |
| AVG | 200 | 1122.49 | 2126.725 | 6.035 | 1.895 | 2248.5 | 205 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
|  | 1 | 10.57 | 10.48667 | 0.073333 | 0.99 | 13.66667 | 4 |
|  | 5 | 32.46667 | 52.29 | 0.193333 | 1.613333 | 50 | 10 |
|  | 10 | 55.71667 | 105.32 | 0.386667 | 1.893333 | 114.6667 | 15 |
|  | 20 | 112.2333 | 212.72 | 0.646667 | 1.896667 | 228 | 25 |
|  | 50 | 279.0467 | 528.8 | 1.49 | 1.896667 | 559.3333 | 55 |
|  | 100 | 560.1333 | 1061.387 | 3.11 | 1.896667 | 1125.333 | 105 |
|  | 200 | 1122.49 | 2126.725 | 6.035 | 1.895 | 2248.5 | 205 |

Mixed RR

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Trial | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 1 | 10.81 | 10.69 | 0.11 | 99% | 10 | 4 |
| 2 | 1 | 10.59 | 10.52 | 0.06 | 99% | 10 | 4 |
| 3 | 1 | 10.54 | 10.48 | 0.05 | 99% | 8 | 4 |
| AVG |  | 10.646667 | 10.563333 | 0.0733333 | 0.99 | 9.3333333 | 4 |
|  |  |  |  |  |  |  |  |
| 1 | 5 | 27.84 | 52.76 | 0.21 | 190% | 304 | 9 |
| 2 | 5 | 27.3 | 51.77 | 0.21 | 190% | 300 | 9 |
| 3 | 5 | 27.24 | 51.65 | 0.09 | 189% | 307 | 11 |
| AVG |  | 27.46 | 52.06 | 0.17 | 1.8966667 | 303.66667 | 9.6666667 |
|  |  |  |  |  |  |  |  |
| 1 | 10 | 56.25 | 106.71 | 0.31 | 190% | 651 | 16 |
| 2 | 10 | 56.69 | 107.46 | 0.32 | 190% | 663 | 18 |
| 3 | 10 | 56.2 | 106.55 | 0.31 | 190% | 648 | 19 |
| AVG |  | 56.38 | 106.90667 | 0.3133333 | 1.9 | 654 | 17.666667 |
|  |  |  |  |  |  |  |  |
| 1 | 20 | 111.82 | 212.08 | 0.69 | 190% | 1322 | 23 |
| 2 | 20 | 111.85 | 212.19 | 0.54 | 190% | 1329 | 31 |
| 3 | 20 | 112.06 | 212.39 | 0.69 | 190% | 1305 | 23 |
| AVG |  | 111.91 | 212.22 | 0.64 | 1.9 | 1318.6667 | 25.666667 |
|  |  |  |  |  |  |  |  |
| 1 | 50 | 279.21 | 529.73 | 1.44 | 190% | 3303 | 96 |
| 2 | 50 | 281.04 | 533.14 | 1.58 | 190% | 3318 | 86 |
| 3 | 50 | 280.59 | 532.23 | 1.48 | 190% | 3293 | 71 |
| AVG |  | 280.28 | 531.7 | 1.5 | 1.9 | 3304.6667 | 84.333333 |
|  |  |  |  |  |  |  |  |
| 1 | 100 | 560.8 | 1063.96 | 3.02 | 190% | 6569 | 143 |
| 2 | 100 | 560.34 | 1063.19 | 2.91 | 190% | 6639 | 178 |
| 3 | 100 | 560.15 | 1062.67 | 3 | 190% | 6619 | 145 |
| AVG |  | 560.43 | 1063.2733 | 2.9766667 | 1.9 | 6609 | 155.33333 |
|  |  |  |  |  |  |  |  |
| 1 | 200 | 1122.03 | 2128.56 | 6.01 | 190% | 13210 | 377 |
| 2 | 200 | 1119.89 | 2124.66 | 5.82 | 190% | 13255 | 390 |
|  |  |  |  |  |  |  |  |
| AVG |  | 1120.96 | 2126.61 | 5.915 | 1.9 | 13232.5 | 383.5 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
|  | 1 | 10.646667 | 10.563333 | 0.0733333 | 0.99 | 9.3333333 | 4 |
|  | 5 | 27.46 | 52.06 | 0.17 | 1.8966667 | 303.66667 | 9.6666667 |
|  | 10 | 56.38 | 106.90667 | 0.3133333 | 1.9 | 654 | 17.666667 |
|  | 20 | 111.91 | 212.22 | 0.64 | 1.9 | 1318.6667 | 25.666667 |
|  | 50 | 280.28 | 531.7 | 1.5 | 1.9 | 3304.6667 | 84.333333 |
|  | 100 | 560.43 | 1063.2733 | 2.9766667 | 1.9 | 6609 | 155.33333 |
|  | 200 | 1120.96 | 2126.61 | 5.915 | 1.9 | 13232.5 | 383.5 |

Mixed OTHER

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Trial | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 1 | 10.9 | 10.62 | 0.18 | 99% | 321 | 4 |
| 2 | 1 | 10.57 | 10.34 | 0.08 | 98% | 584 | 4 |
| 3 | 1 | 10.56 | 10.43 | 0.09 | 99% | 151 | 4 |
| AVG | 1 | 10.67667 | 10.46333 | 0.116667 | 0.986667 | 352 | 4 |
| 1 | 5 | 27.55 | 52.98 | 0.48 | 194% | 8937 | 8 |
| 2 | 5 | 26.78 | 51.74 | 0.48 | 194% | 8939 | 13 |
| 3 | 5 | 26.85 | 51.73 | 0.39 | 194% | 8874 | 12 |
| AVG | 5 | 27.06 | 52.15 | 0.45 | 1.94 | 8916.667 | 11 |
| 1 | 10 | 53.82 | 103.66 | 0.89 | 194% | 28373 | 15 |
| 2 | 10 | 54.36 | 105.02 | 0.92 | 194% | 28634 | 21 |
| 3 | 10 | 54.29 | 104.72 | 1.07 | 194% | 28618 | 13 |
| AVG | 10 | 54.15667 | 104.4667 | 0.96 | 1.94 | 28541.67 | 16.33333 |
| 1 | 20 | 106.18 | 208.21 | 1.48 | 197% | 57440 | 42 |
| 2 | 20 | 107.22 | 209.34 | 1.58 | 196% | 57933 | 28 |
| 3 | 20 | 108.47 | 210.19 | 1.88 | 195% | 58737 | 24 |
| AVG | 20 | 107.29 | 209.2467 | 1.646667 | 1.96 | 58036.67 | 31.33333 |
| 1 | 50 | 266.4 | 521.94 | 3.38 | 197% | 143977 | 53 |
| 2 | 50 | 263.97 | 519.41 | 2.9 | 197% | 141693 | 84 |
| 3 | 50 | 265.91 | 521.67 | 3.47 | 197% | 143116 | 78 |
| AVG | 50 | 265.4267 | 521.0067 | 3.25 | 1.97 | 142928.7 | 71.66667 |
| 1 | 100 | 522.54 | 1033.18 | 5.28 | 198% | 280999 | 115 |
| 2 | 100 | 523.11 | 1033.95 | 5.17 | 198% | 281483 | 164 |
| 3 | 100 | 524.84 | 1037.4 | 5.43 | 198% | 281866 | 157 |
| AVG | 100 | 523.4967 | 1034.843 | 5.293333 | 1.98 | 281449.3 | 145.3333 |
| 1 | 200 | 1039.31 | 2060.05 | 8.91 | 199% | 555630 | 230 |
| 2 | 200 | 1044.05 | 2070.7 | 8.45 | 199% | 558683 | 393 |
| AVG | 200 | 1041.68 | 2065.375 | 8.68 | 1.99 | 557156.5 | 311.5 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  | Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
|  | 1 | 10.67667 | 10.46333 | 0.116667 | 0.986667 | 352 | 4 |
|  | 5 | 27.06 | 52.15 | 0.45 | 1.94 | 8916.667 | 11 |
|  | 10 | 54.15667 | 104.4667 | 0.96 | 1.94 | 28541.67 | 16.33333 |
|  | 20 | 107.29 | 209.2467 | 1.646667 | 1.96 | 58036.67 | 31.33333 |
|  | 50 | 265.4267 | 521.0067 | 3.25 | 1.97 | 142928.7 | 71.66667 |
|  | 100 | 523.4967 | 1034.843 | 5.293333 | 1.98 | 281449.3 | 145.3333 |
|  | 200 | 1041.68 | 2065.375 | 8.68 | 1.99 | 557156.5 | 311.5 |

Compiled Data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Compute Bound FIFO | | | | | | |
| Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 4.165 | 4.155 | 0.01 | 0.99 | 9 | 4 |
| 5 | 12.62 | 20.515 | 0.01 | 1.625 | 20 | 10 |
| 10 | 22.08 | 41.915 | 0.025 | 1.895 | 49 | 15 |
| 20 | 44.27 | 84.21 | 0.01 | 1.9 | 92.5 | 25 |
| 50 | 110.455 | 209.75 | 0.045 | 1.89 | 237 | 55 |
| 100 | 222.32 | 422.42 | 0.045 | 1.9 | 449 | 105 |
| 200 | 441.43 | 838.965 | 0.085 | 1.9 | 890.5 | 205 |
|  |  |  |  |  |  |  |
| Compute Bound RR | | | | | | |
| Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 4.225 | 4.21 | 0.015 | 0.99 | 6.5 | 4 |
| 5 | 10.78 | 20.52 | 0.025 | 1.9 | 116 | 10 |
| 10 | 22.26 | 42.29 | 0.01 | 1.895 | 262.5 | 18 |
| 20 | 44.12 | 83.92 | 0 | 1.9 | 520 | 27.5 |
| 50 | 110.64 | 210.475 | 0.03 | 1.9 | 1301.5 | 69 |
| 100 | 221.805 | 421.83 | 0.05 | 1.9 | 2609.5 | 133.5 |
| 200 | 443.105 | 842.57 | 0.145 | 1.9 | 5240 | 295.5 |
|  |  |  |  |  |  |  |
| Compute Bound OTHER | | | | | | |
| Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 4.155 | 4.02 | 0.08 | 0.985 | 262 | 1 |
| 5 | 10.64 | 20.565 | 0.13 | 1.94 | 3513.5 | 8 |
| 10 | 21.41 | 41.22 | 0.335 | 1.94 | 11289 | 15 |
| 20 | 42.29 | 82.75 | 0.41 | 1.96 | 22769.5 | 25.5 |
| 50 | 104.325 | 205.94 | 0.63 | 1.975 | 56051.5 | 86 |
| 100 | 205.225 | 406.975 | 0.85 | 1.98 | 109759.5 | 128.5 |
| 200 | 412.515 | 819.67 | 1.195 | 1.985 | 220366.5 | 278 |
|  |  |  |  |  |  |  |
| IO Bound FIFO | | | | | | |
| Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 7.765 | 0 | 1.275 | 0.16 | 2823.5 | 13431 |
| 5 | 3.165 | 0 | 1 | 0.345 | 3597 | 12669 |
| 10 | 3.91 | 0 | 1.06 | 0.27 | 3618 | 12405.5 |
| 20 | 3.475 | 0 | 1.015 | 0.29 | 3639.5 | 12476 |
| 50 | 4.015 | 0 | 1.03 | 0.255 | 3613.5 | 12443.5 |
| 100 | 4.34 | 0 | 1.135 | 0.27 | 3767 | 12569.5 |
| 200 | 2.315 | 0 | 0.945 | 0.465 | 4027 | 12418.5 |
|  |  |  |  |  |  |  |
| IO Bound RR | | | | | | |
| Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 13.25 | 0 | 1.366667 | 0.103333 | 1978.333 | 13258.67 |
| 5 | 7.313333 | 0 | 1.306667 | 0.18 | 3532.333 | 12572.33 |
| 10 | 6.226667 | 0 | 1.25 | 0.2 | 3131.667 | 12629.33 |
| 20 | 8.146667 | 0 | 1.43 | 0.173333 | 2593.333 | 12802 |
| 50 | 5.84 | 0 | 1.183333 | 0.203333 | 2929.333 | 12660 |
| 100 | 5.796667 | 0 | 1.253333 | 0.21 | 3138.333 | 12591 |
| 200 | 5.565 | 0 | 1.225 | 0.24 | 2652 | 12760 |
|  |  |  |  |  |  |  |
| IO Bound OTHER | | | | | | |
| Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 9.89 | 0 | 1.303333 | 0.136667 | 2629 | 13490 |
| 5 | 3.44 | 0 | 1.096667 | 0.316667 | 3026.333 | 12716.67 |
| 10 | 1.75 | 0 | 0.86 | 0.493333 | 3479.333 | 12471 |
| 20 | 2.306667 | 0 | 0.876667 | 0.396667 | 3336.333 | 12466.67 |
| 50 | 2.216667 | 0 | 0.836667 | 0.38 | 3877 | 12339 |
| 100 | 1.93 | 0 | 0.856667 | 0.453333 | 4161.333 | 12321.33 |
| 200 | 1.595 | 0 | 0.885 | 0.565 | 3245 | 12635.5 |
|  |  |  |  |  |  |  |
| Mixed FIFO | | | | | | |
| Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 10.57 | 10.48667 | 0.073333 | 0.99 | 13.66667 | 4 |
| 5 | 32.46667 | 52.29 | 0.193333 | 1.613333 | 50 | 10 |
| 10 | 55.71667 | 105.32 | 0.386667 | 1.893333 | 114.6667 | 15 |
| 20 | 112.2333 | 212.72 | 0.646667 | 1.896667 | 228 | 25 |
| 50 | 279.0467 | 528.8 | 1.49 | 1.896667 | 559.3333 | 55 |
| 100 | 560.1333 | 1061.387 | 3.11 | 1.896667 | 1125.333 | 105 |
| 200 | 1122.49 | 2126.725 | 6.035 | 1.895 | 2248.5 | 205 |
|  |  |  |  |  |  |  |
| Mixed RR | | | | | | |
| Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 10.64667 | 10.56333 | 0.073333 | 0.99 | 9.333333 | 4 |
| 5 | 27.46 | 52.06 | 0.17 | 1.896667 | 303.6667 | 9.666667 |
| 10 | 56.38 | 106.9067 | 0.313333 | 1.9 | 654 | 17.66667 |
| 20 | 111.91 | 212.22 | 0.64 | 1.9 | 1318.667 | 25.66667 |
| 50 | 280.28 | 531.7 | 1.5 | 1.9 | 3304.667 | 84.33333 |
| 100 | 560.43 | 1063.273 | 2.976667 | 1.9 | 6609 | 155.3333 |
| 200 | 1120.96 | 2126.61 | 5.915 | 1.9 | 13232.5 | 383.5 |
|  |  |  |  |  |  |  |
| Mixed OTHER | | | | | | |
| Processes | Wall (Actual time elapsed) | User (Time spent executing in userspace) | System (Time spent executing in kernel space) | CPU (Percentage of CPU used) | I-Switched (Number of times involuntarily context switched) | V-Switched (Number of times voluntarily context switched) |
| 1 | 10.67667 | 10.46333 | 0.116667 | 0.986667 | 352 | 4 |
| 5 | 27.06 | 52.15 | 0.45 | 1.94 | 8916.667 | 11 |
| 10 | 54.15667 | 104.4667 | 0.96 | 1.94 | 28541.67 | 16.33333 |
| 20 | 107.29 | 209.2467 | 1.646667 | 1.96 | 58036.67 | 31.33333 |
| 50 | 265.4267 | 521.0067 | 3.25 | 1.97 | 142928.7 | 71.66667 |
| 100 | 523.4967 | 1034.843 | 5.293333 | 1.98 | 281449.3 | 145.3333 |
| 200 | 1041.68 | 2065.375 | 8.68 | 1.99 | 557156.5 | 311.5 |

## Appendix B

Data Collection Driver

#File for collecting data automatically

cp DataTemplate.csv ComputeBoundOTHER.csv

cp DataTemplate.csv ComputeBoundFIFO.csv

cp DataTemplate.csv ComputeBoundRR.csv

cp DataTemplate.csv IOBoundOTHER.csv

cp DataTemplate.csv IOBoundFIFO.csv

cp DataTemplate.csv IOBoundRR.csv

cp DataTemplate.csv MixedOTHER.csv

cp DataTemplate.csv MixedFIFO.csv

cp DataTemplate.csv MixedRR.csv

for PROCESSES in 1 5 10 20 50 100 200 500

do

echo "Running tests with $PROCESSES processes..."

for TRIAL in 1 2 3

do

echo "Trial $TRIAL"

echo -n "$TRIAL,$PROCESSES," >> ComputeBoundOTHER.csv

echo -n "$TRIAL,$PROCESSES," >> ComputeBoundFIFO.csv

echo -n "$TRIAL,$PROCESSES," >> ComputeBoundRR.csv

echo -n "$TRIAL,$PROCESSES," >> IOBoundOTHER.csv

echo -n "$TRIAL,$PROCESSES," >> IOBoundFIFO.csv

echo -n "$TRIAL,$PROCESSES," >> IOBoundRR.csv

echo -n "$TRIAL,$PROCESSES," >> MixedOTHER.csv

echo -n "$TRIAL,$PROCESSES," >> MixedFIFO.csv

echo -n "$TRIAL,$PROCESSES," >> MixedRR.csv

./testscript $PROCESSES

done

echo "" >> ComputeBoundOTHER.csv

echo "" >> ComputeBoundFIFO.csv

echo "" >> ComputeBoundRR.csv

echo "" >> IOBoundOTHER.csv

echo "" >> IOBoundFIFO.csv

echo "" >> IOBoundRR.csv

echo "" >> MixedOTHER.csv

echo "" >> MixedFIFO.csv

echo "" >> MixedRR.csv

echo

echo

done

echo "Tests complete!"

TestCode Driver

#/!/bin/bash

#File: testscript

#Author: Andy Sayler

#Project: CSCI 3753 Programming Assignment 3

#Create Date: 2012/03/09

#Modify Date: 2012/03/21

#Description:

# A simple bash script to run a signle copy of each test case

# and gather the relevent data.

# Altered by Raymond Duncan for PAX alteration date 5/2/16

ITERATIONS=100000000

BYTESTOCOPY=1024000

BLOCKSIZE=1024

NUMPROCS="$1"

TIMEFORMAT="wall=%e user=%U system=%S CPU=%P i-switched=%c v-switched=%w"

FORMAT="%e,%U,%S,%P,%c,%w"

RANDOMIN="/dev/urandom"

DUMPOUT="rwoutput"

MAKE="make -s"

echo Building code...

$MAKE clean

$MAKE

echo Starting test runs...

#Compute Bound

echo Calculating pi over $ITERATIONS iterations using SCHED\_OTHER with $NUMPROCS simultaneous processes...

(/usr/bin/time -f "$FORMAT" ./pi-sched $ITERATIONS SCHED\_OTHER $NUMPROCS > /dev/null) &>> ComputeBoundOTHER.csv

echo Calculating pi over $ITERATIONS iterations using SCHED\_FIFO with $NUMPROCS simultaneous processes...

(/usr/bin/time -f "$FORMAT" sudo ./pi-sched $ITERATIONS SCHED\_FIFO $NUMPROCS > /dev/null) &>> ComputeBoundFIFO.csv

echo Calculating pi over $ITERATIONS iterations using SCHED\_RR with $NUMPROCS simultaneous processes...

(/usr/bin/time -f "$FORMAT" sudo ./pi-sched $ITERATIONS SCHED\_RR $NUMPROCS > /dev/null) &>> ComputeBoundRR.csv

echo

echo

#I/O Bound

echo Copying $BYTESTOCOPY bytes in blocks of $BLOCKSIZE from /dev/urandom to $DUMPOUT

echo using SCHED\_OTHER with $NUMPROCS simultaneous processes...

(/usr/bin/time -f "$FORMAT" sudo ./rw $BYTESTOCOPY $BLOCKSIZE $RANDOMIN $DUMPOUT $NUMPROCS > /dev/null) &>> IOBoundOTHER.csv

echo Copying $BYTESTOCOPY bytes in blocks of $BLOCKSIZE from /dev/urandom to $DUMPOUT

echo using SCHED\_FIFO with $NUMPROCS simultaneous processes...

(/usr/bin/time -f "$FORMAT" sudo ./rw $BYTESTOCOPY $BLOCKSIZE $RANDOMIN $DUMPOUT $NUMPROCS $SCHED\_FIFO > /dev/null) &>> IOBoundFIFO.csv

echo Copying $BYTESTOCOPY bytes in blocks of $BLOCKSIZE from /dev/urandom to $DUMPOUT

echo using SCHED\_RR with $NUMPROCS simultaneous processes...

(/usr/bin/time -f "$FORMAT" sudo ./rw $BYTESTOCOPY $BLOCKSIZE $RANDOMIN $DUMPOUT $NUMPROCS $SCHED\_RR > /dev/null) &>> IOBoundRR.csv

echo

echo

#Mixed

echo Calculating pi over $(( $ITERATIONS/10 )) iterations using SCHED\_OTHER with $NUMPROCS simultaneous processes...

(/usr/bin/time -f "$FORMAT" sudo ./pi-sched-mixed $(( $ITERATIONS / 10 )) SCHED\_OTHER $NUMPROCS > /dev/null) &>> MixedOTHER.csv

echo Calculating pi over $(( $ITERATIONS/10 )) iterations using SCHED\_FIFO with $NUMPROCS simultaneous processes...

(/usr/bin/time -f "$FORMAT" sudo ./pi-sched-mixed $(( $ITERATIONS / 10 )) SCHED\_FIFO $NUMPROCS > /dev/null) &>> MixedFIFO.csv

echo Calculating pi over $(( $ITERATIONS/10 )) iterations using SCHED\_RR with $NUMPROCS simultaneous processes...

(/usr/bin/time -f "$FORMAT" sudo ./pi-sched-mixed $(( $ITERATIONS / 10 )) SCHED\_RR $NUMPROCS > /dev/null) &>> MixedRR.csv

Compute Bound Test Code

/\*

\* File: pi-sched.c

\* Author: Andy Sayler

\* Project: CSCI 3753 Programming Assignment 3

\* Create Date: 2012/03/07

\* Modify Date: 2012/03/09

\* Description:

\* This file contains a simple program for statistically

\* calculating pi using a specific scheduling policy.

Altered slightly for use in PAX by Raymond Duncan alteration date 5/2/16

\*/

/\* Local Includes \*/

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <errno.h>

#include <sched.h>

#include <unistd.h>

#include <sys/wait.h>

#define DEFAULT\_ITERATIONS 1000000

#define RADIUS (RAND\_MAX / 2)

inline double dist(double x0, double y0, double x1, double y1){

return sqrt(pow((x1-x0),2) + pow((y1-y0),2));

}

inline double zeroDist(double x, double y){

return dist(0, 0, x, y);

}

int main(int argc, char\* argv[]){

long i;

long iterations;

pid\_t pid;

struct sched\_param param;

int policy;

int numProcesses;

double x, y;

double inCircle = 0.0;

double inSquare = 0.0;

double pCircle = 0.0;

double piCalc = 0.0;

/\* Process program arguments to select iterations and policy \*/

/\* Set default iterations if not supplied \*/

if(argc < 2){

iterations = DEFAULT\_ITERATIONS;

}

/\* Set default policy if not supplied \*/

if(argc < 3){

policy = SCHED\_OTHER;

}

/\*Set default processes\*/

if(argc < 4){

numProcesses = 10;

}

/\* Set iterations if supplied \*/

if(argc > 1){

iterations = atol(argv[1]);

if(iterations < 1){

fprintf(stderr, "Bad iterations value\n");

exit(EXIT\_FAILURE);

}

}

/\* Set policy if supplied \*/

if(argc > 2){

if(!strcmp(argv[2], "SCHED\_OTHER")){

policy = SCHED\_OTHER;

}

else if(!strcmp(argv[2], "SCHED\_FIFO")){

policy = SCHED\_FIFO;

}

else if(!strcmp(argv[2], "SCHED\_RR")){

policy = SCHED\_RR;

}

else{

fprintf(stderr, "Unhandeled scheduling policy\n");

exit(EXIT\_FAILURE);

}

}

if(argc > 3){

numProcesses = atol(argv[3]);

}

/\* Set process to max prioty for given scheduler \*/

param.sched\_priority = sched\_get\_priority\_max(policy);

/\* Set new scheduler policy \*/

fprintf(stdout, "Current Scheduling Policy: %d\n", sched\_getscheduler(0));

fprintf(stdout, "Setting Scheduling Policy to: %d\n", policy);

if(sched\_setscheduler(0, policy, &param)){

perror("Error setting scheduler policy");

exit(EXIT\_FAILURE);

}

fprintf(stdout, "New Scheduling Policy: %d\n", sched\_getscheduler(0));

for(i = 1; i < numProcesses; i++){

pid = fork();

if(pid == 0){

break;

}

}

/\* Calculate pi using statistical methode across all iterations\*/

for(i=0; i<iterations; i++){

x = (random() % (RADIUS \* 2)) - RADIUS;

y = (random() % (RADIUS \* 2)) - RADIUS;

if(zeroDist(x,y) < RADIUS){

inCircle++;

}

inSquare++;

}

/\* Finish calculation \*/

pCircle = inCircle/inSquare;

piCalc = pCircle \* 4.0;

/\* Print result \*/

fprintf(stdout, "pi = %f\n", piCalc);

for(i = 1; i < numProcesses; i++){

wait(NULL);

}

return 0;

}

IO Bound Test Code

/\*

\* File: rw.c

\* Author: Andy Sayler

\* Project: CSCI 3753 Programming Assignment 3

\* Create Date: 2012/03/19

\* Modify Date: 2012/03/20

\* Description: A small i/o bound program to copy N bytes from an input

\* file to an output file. May read the input file multiple

\* times if N is larger than the size of the input file.

Altered by Raymond Duncan for PAX alteration date 5/2/16

\*/

/\* Include Flags \*/

#define \_GNU\_SOURCE

/\* System Includes \*/

#include <stdlib.h>

#include <stdio.h>

#include <unistd.h>

#include <errno.h>

#include <fcntl.h>

#include <string.h>

#include <sys/types.h>

#include <sys/stat.h>

#include <unistd.h>

#include <sys/wait.h>

#include <sched.h>

/\* Local Defines \*/

#define MAXFILENAMELENGTH 80

#define DEFAULT\_INPUTFILENAME "rwinput"

#define DEFAULT\_OUTPUTFILENAMEBASE "rwoutput"

#define DEFAULT\_BLOCKSIZE 1024

#define DEFAULT\_TRANSFERSIZE 1024\*100

int main(int argc, char\* argv[]){

int rv;

int inputFD;

int outputFD;

char inputFilename[MAXFILENAMELENGTH];

char outputFilename[MAXFILENAMELENGTH];

char outputFilenameBase[MAXFILENAMELENGTH];

ssize\_t transfersize = 0;

ssize\_t blocksize = 0;

char\* transferBuffer = NULL;

ssize\_t buffersize;

ssize\_t bytesRead = 0;

ssize\_t totalBytesRead = 0;

int totalReads = 0;

ssize\_t bytesWritten = 0;

ssize\_t totalBytesWritten = 0;

int totalWrites = 0;

int inputFileResets = 0;

long i;

long numProcs;

pid\_t pid;

struct sched\_param param;

int policy;

/\* Process program arguments to select run-time parameters \*/

/\* Set supplied transfer size or default if not supplied \*/

if(argc < 2){

transfersize = DEFAULT\_TRANSFERSIZE;

}

else{

transfersize = atol(argv[1]);

if(transfersize < 1){

fprintf(stderr, "Bad transfersize value\n");

exit(EXIT\_FAILURE);

}

}

/\* Set supplied block size or default if not supplied \*/

if(argc < 3){

blocksize = DEFAULT\_BLOCKSIZE;

}

else{

blocksize = atol(argv[2]);

if(blocksize < 1){

fprintf(stderr, "Bad blocksize value\n");

exit(EXIT\_FAILURE);

}

}

/\* Set supplied input filename or default if not supplied \*/

if(argc < 4){

if(strnlen(DEFAULT\_INPUTFILENAME, MAXFILENAMELENGTH) >= MAXFILENAMELENGTH){

fprintf(stderr, "Default input filename too long\n");

exit(EXIT\_FAILURE);

}

strncpy(inputFilename, DEFAULT\_INPUTFILENAME, MAXFILENAMELENGTH);

}

else{

if(strnlen(argv[3], MAXFILENAMELENGTH) >= MAXFILENAMELENGTH){

fprintf(stderr, "Input filename too long\n");

exit(EXIT\_FAILURE);

}

strncpy(inputFilename, argv[3], MAXFILENAMELENGTH);

}

/\* Set supplied output filename base or default if not supplied \*/

if(argc < 5){

if(strnlen(DEFAULT\_OUTPUTFILENAMEBASE, MAXFILENAMELENGTH) >= MAXFILENAMELENGTH){

fprintf(stderr, "Default output filename base too long\n");

exit(EXIT\_FAILURE);

}

strncpy(outputFilenameBase, DEFAULT\_OUTPUTFILENAMEBASE, MAXFILENAMELENGTH);

}

else{

if(strnlen(argv[4], MAXFILENAMELENGTH) >= MAXFILENAMELENGTH){

fprintf(stderr, "Output filename base is too long\n");

exit(EXIT\_FAILURE);

}

strncpy(outputFilenameBase, argv[4], MAXFILENAMELENGTH);

}

/\*Set supplied number of processes or set default\*/

if(argc < 6){

numProcs = atol(argv[5]);

}

else{

numProcs = 5;

}

/\*Set policy\*/

if(argc < 7){

policy = SCHED\_OTHER;

}

else{

//Taken from pi-sched.c

if(!strcmp(argv[2], "SCHED\_OTHER")){

policy = SCHED\_OTHER;

}

else if(!strcmp(argv[2], "SCHED\_FIFO")){

policy = SCHED\_FIFO;

}

else if(!strcmp(argv[2], "SCHED\_RR")){

policy = SCHED\_RR;

}

else{

fprintf(stderr, "Unhandeled scheduling policy\n");

exit(EXIT\_FAILURE);

}

}

//The following is a snippet taken from Andy Sayler's pi-sched.c

/\* Set process to max prioty for given scheduler \*/

param.sched\_priority = sched\_get\_priority\_max(policy);

/\* Set new scheduler policy \*/

fprintf(stdout, "Current Scheduling Policy: %d\n", sched\_getscheduler(0));

fprintf(stdout, "Setting Scheduling Policy to: %d\n", policy);

if(sched\_setscheduler(0, policy, &param)){

perror("Error setting scheduler policy");

exit(EXIT\_FAILURE);

}

fprintf(stdout, "New Scheduling Policy: %d\n", sched\_getscheduler(0));

/\* Confirm blocksize is multiple of and less than transfersize\*/

if(blocksize > transfersize){

fprintf(stderr, "blocksize can not exceed transfersize\n");

exit(EXIT\_FAILURE);

}

if(transfersize % blocksize){

fprintf(stderr, "blocksize must be multiple of transfersize\n");

exit(EXIT\_FAILURE);

}

for(i = 1; i < numProcs; i++){

pid = fork();

if(pid == 0){

break;

}

}

/\* Allocate buffer space \*/

buffersize = blocksize;

if(!(transferBuffer = malloc(buffersize\*sizeof(\*transferBuffer)))){

perror("Failed to allocate transfer buffer");

exit(EXIT\_FAILURE);

}

/\* Open Input File Descriptor in Read Only mode \*/

if((inputFD = open(inputFilename, O\_RDONLY | O\_SYNC)) < 0){

perror("Failed to open input file");

exit(EXIT\_FAILURE);

}

/\* Open Output File Descriptor in Write Only mode with standard permissions\*/

rv = snprintf(outputFilename, MAXFILENAMELENGTH, "%s-%d",

outputFilenameBase, getpid());

if(rv > MAXFILENAMELENGTH){

fprintf(stderr, "Output filenmae length exceeds limit of %d characters.\n",

MAXFILENAMELENGTH);

exit(EXIT\_FAILURE);

}

else if(rv < 0){

perror("Failed to generate output filename");

exit(EXIT\_FAILURE);

}

if((outputFD =

open(outputFilename,

O\_WRONLY | O\_CREAT | O\_TRUNC | O\_SYNC,

S\_IRUSR | S\_IWUSR | S\_IRGRP | S\_IWGRP | S\_IROTH)) < 0){

perror("Failed to open output file");

exit(EXIT\_FAILURE);

}

/\* Print Status \*/

fprintf(stdout, "Reading from %s and writing to %s\n",

inputFilename, outputFilename);

/\* Read from input file and write to output file\*/

do{

/\* Read transfersize bytes from input file\*/

bytesRead = read(inputFD, transferBuffer, buffersize);

if(bytesRead < 0){

perror("Error reading input file");

exit(EXIT\_FAILURE);

}

else{

totalBytesRead += bytesRead;

totalReads++;

}

/\* If all bytes were read, write to output file\*/

if(bytesRead == blocksize){

bytesWritten = write(outputFD, transferBuffer, bytesRead);

if(bytesWritten < 0){

perror("Error writing output file");

exit(EXIT\_FAILURE);

}

else{

totalBytesWritten += bytesWritten;

totalWrites++;

}

}

/\* Otherwise assume we have reached the end of the input file and reset \*/

else{

if(lseek(inputFD, 0, SEEK\_SET)){

perror("Error resetting to beginning of file");

exit(EXIT\_FAILURE);

}

inputFileResets++;

}

}while(totalBytesWritten < transfersize);

/\* Output some possibly helpfull info to make it seem like we were doing stuff \*/

fprintf(stdout, "Read: %zd bytes in %d reads\n",

totalBytesRead, totalReads);

fprintf(stdout, "Written: %zd bytes in %d writes\n",

totalBytesWritten, totalWrites);

fprintf(stdout, "Read input file in %d pass%s\n",

(inputFileResets + 1), (inputFileResets ? "es" : ""));

fprintf(stdout, "Processed %zd bytes in blocks of %zd bytes\n",

transfersize, blocksize);

/\* Free Buffer \*/

free(transferBuffer);

/\* Close Output File Descriptor \*/

if(close(outputFD)){

perror("Failed to close output file");

exit(EXIT\_FAILURE);

}

/\* Close Input File Descriptor \*/

if(close(inputFD)){

perror("Failed to close input file");

exit(EXIT\_FAILURE);

}

for(i = 1; i < numProcs; i++){

wait(NULL);

}

fprintf(stdout, "Process %d finished successfully\n", pid);

return EXIT\_SUCCESS;

}

Mixed Test Code

/\*

\* File: pi-sched.c

\* Author: Andy Sayler

\* Project: CSCI 3753 Programming Assignment 3

\* Create Date: 2012/03/07

\* Modify Date: 2012/03/09

\* Description:

\* This file contains a simple program for statistically

\* calculating pi using a specific scheduling policy.

Altered slightly from pi-sched for use in PAX by Raymond Duncan alteration date 5/2/16

\*/

/\* Local Includes \*/

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <errno.h>

#include <sched.h>

#include <unistd.h>

#include <sys/wait.h>

#define DEFAULT\_ITERATIONS 1000000

#define RADIUS (RAND\_MAX / 2)

inline double dist(double x0, double y0, double x1, double y1){

return sqrt(pow((x1-x0),2) + pow((y1-y0),2));

}

inline double zeroDist(double x, double y){

return dist(0, 0, x, y);

}

int main(int argc, char\* argv[]){

long i;

long iterations;

pid\_t pid;

struct sched\_param param;

int policy;

int numProcesses;

double x, y;

double inCircle = 0.0;

double inSquare = 0.0;

double pCircle = 0.0;

double piCalc = 0.0;

FILE \*fi;

/\* Process program arguments to select iterations and policy \*/

/\* Set default iterations if not supplied \*/

if(argc < 2){

iterations = DEFAULT\_ITERATIONS;

}

/\* Set default policy if not supplied \*/

if(argc < 3){

policy = SCHED\_OTHER;

}

/\*Set default processes\*/

if(argc < 4){

numProcesses = 10;

}

/\* Set iterations if supplied \*/

if(argc > 1){

iterations = atol(argv[1]);

if(iterations < 1){

fprintf(stderr, "Bad iterations value\n");

exit(EXIT\_FAILURE);

}

}

/\* Set policy if supplied \*/

if(argc > 2){

if(!strcmp(argv[2], "SCHED\_OTHER")){

policy = SCHED\_OTHER;

}

else if(!strcmp(argv[2], "SCHED\_FIFO")){

policy = SCHED\_FIFO;

}

else if(!strcmp(argv[2], "SCHED\_RR")){

policy = SCHED\_RR;

}

else{

fprintf(stderr, "Unhandeled scheduling policy\n");

exit(EXIT\_FAILURE);

}

}

if(argc > 3){

numProcesses = atol(argv[3]);

}

/\* Set process to max prioty for given scheduler \*/

param.sched\_priority = sched\_get\_priority\_max(policy);

/\* Set new scheduler policy \*/

fprintf(stdout, "Current Scheduling Policy: %d\n", sched\_getscheduler(0));

fprintf(stdout, "Setting Scheduling Policy to: %d\n", policy);

if(sched\_setscheduler(0, policy, &param)){

perror("Error setting scheduler policy");

exit(EXIT\_FAILURE);

}

fprintf(stdout, "New Scheduling Policy: %d\n", sched\_getscheduler(0));

for(i = 1; i < numProcesses; i++){

pid = fork();

if(pid == 0){

break;

}

}

fi = fopen("/dev/null","w");

/\* Calculate pi using statistical methode across all iterations\*/

for(i=0; i<iterations; i++){

x = (random() % (RADIUS \* 2)) - RADIUS;

y = (random() % (RADIUS \* 2)) - RADIUS;

if(zeroDist(x,y) < RADIUS){

inCircle++;

}

inSquare++;

if(iterations % 1 == 0){

fprintf(fi, "The current number of in circle hits is %f, and in square hits is %f", inCircle, inSquare);

}

}

fclose(fi);

/\* Finish calculation \*/

pCircle = inCircle/inSquare;

piCalc = pCircle \* 4.0;

/\* Print result \*/

fprintf(stdout, "pi = %f\n", piCalc);

for(i = 1; i < numProcesses; i++){

wait(NULL);

}

return 0;

}