For project 3 of operating system, gcc.trace is mainly used for performance analysis.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number Of Frames | Refresh Rate | Total Memory Access | Page Faults | Write to Disk |
| 8 | 1 | 1000000 | 703469 | 101920 |
| 8 | 2 | 1000000 | 380264 | 60378 |
| 8 | 3 | 1000000 | 307108 | 43536 |
| 8 | 4 | 1000000 | 262094 | 33252 |
| 8 | 5 | 1000000 | 214989 | 21937 |
| 8 | 11 | 1000000 | 183183 | 17407 |
| 8 | 12 | 1000000 | 181485 | 17474 |
| 8 | 13 | 1000000 | 179584 | 17631 |
| 8 | 14 | 1000000 | 178152 | 17698 |
| 8 | 15 | 1000000 | 177016 | 17869 |
| 8 | 16 | 1000000 | 176355 | 18083 |
| 8 | 17 | 1000000 | 175583 | 18116 |
| 8 | 18 | 1000000 | 175025 | 18228 |
| 8 | 19 | 1000000 | 174456 | 18342 |
| 8 | 20 | 1000000 | 174524 | 18428 |
| 8 | 21 | 1000000 | 174329 | 18560 |
| 8 | 30 | 1000000 | 179499 | 19242 |
| 8 | 50 | 1000000 | 197711 | 18928 |
| 8 | 100 | 1000000 | 235415 | 17651 |

*Table 1. NRU with 8 frames*

*Graph 1. NRU - Page Fault Numbers vs. Refresh Rate with 8 frames.*

Based on table 1 and graph 1, the best refresh rate for the gcc.trace using 8 frames is 21. Because it generates the lowest number of page faults. We will use this fresh rate in subsequent analysis.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number of frames | Refresh Rate | TAU | Total Memory Access | Page Faults | Write to Disk |
| 8 | 21 | 5 | 1000000 | 177925 | 40898 |
| 8 | 21 | 6 | 1000000 | 178012 | 39882 |
| 8 | 21 | 7 | 1000000 | 178148 | 38964 |
| 8 | 21 | 8 | 1000000 | 178155 | 38022 |
| 8 | 21 | 9 | 1000000 | 178321 | 37021 |
| 8 | 21 | 10 | 1000000 | 178363 | 36118 |
| 8 | 21 | 11 | 1000000 | 178638 | 35132 |
| 8 | 21 | 12 | 1000000 | 178808 | 34090 |
| 8 | 21 | 13 | 1000000 | 179013 | 33161 |
| 8 | 21 | 15 | 1000000 | 179652 | 30948 |
| 8 | 21 | 16 | 1000000 | 180030 | 29792 |
| 8 | 21 | 17 | 1000000 | 180521 | 28454 |
| 8 | 21 | 18 | 1000000 | 181104 | 27087 |
| 8 | 21 | 19 | 1000000 | 182217 | 25462 |
| 8 | 21 | 20 | 1000000 | 184452 | 23323 |
| 8 | 21 | 21 | 1000000 | 184452 | 23323 |

*Table 2. Working Set with 8 frames, refresh = 21*

*Graph 2. Work set - Page Fault Numbers vs. Refresh Rate with 8 frames at refresh = 21*

Initially for working set algorithm, a refresh time = 21 is used to find the best tau. From table 2 and graph 2, the best tau is 5.

Subsequently various value of refresh is tested to find the best combination of refresh and tau that result in fewest page faults.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Number of frames | Refresh Rate | TAU | Total Memory Access | Page Faults | Write to Disk |
| 8 | 13 | 5 | 1000000 | 176225 | 34435 |
| 8 | 13 | 6 | 1000000 | 176415 | 32814 |
| 8 | 13 | 7 | 1000000 | 176662 | 31141 |
| 8 | 13 | 8 | 1000000 | 177085 | 29426 |
| 8 | 13 | 9 | 1000000 | 177593 | 27542 |
| 8 | 13 | 10 | 1000000 | 178704 | 25360 |
| 8 | 13 | 11 | 1000000 | 180635 | 23195 |
| 8 | 13 | 12 | 1000000 | 188210 | 19979 |
| 8 | 13 | 13 | 1000000 | 188210 | 19979 |
| 8 | 13 | 15 | 1000000 | 188210 | 19979 |
| 8 | 13 | 16 | 1000000 | 188210 | 19979 |
| 8 | 13 | 17 | 1000000 | 188210 | 19979 |

*Table 3. Working set - Page Fault Numbers vs. Refresh Rate with 8 frames at refresh = 13*

*Graph 3. Working set - Page Fault Numbers vs. Refresh Rate with 8 frames at refresh = 13*

From table 3 and graph 3, the best combination of refresh time and tau is determined to be 13 and 5 accordingly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 8 | 16 | 32 | 64 |
| OPT | 118480 | 80307 | 55802 | 38050 |
| Clock | 181856 | 121682 | 87686 | 61640 |
| NRU | 174329 | 151435 | 146721 | 136656 |
| Work | 176225 | 132102 | 101227 | 75313 |

*Table 4. Opt, Clock, NRU & Working set comparison with frame number at 8, 16, 32 & 64*

*Graph 4. Opt, Clock, NRU & Working set comparison with frame number at 8, 16, 32 & 64*

Table 4 and graph 4 demonstrate the overall performance of these four page replacement algorithms. It is quite clear that opt has the fewest page faults therefore is the best algorithm to use in operating system. However, because human kind can never tell future, opt is just a theoretical algorithm showcases the best performance. All page replacement algorithms generate fewer page faults as the number of frames increases. For the rest three algorithms, clock implements a second chance algorithm seems to have the fewest page faults over time. Note that the working set’s performance increases more dramatically than NRU’s indicating that a good combination of refresh time and tau will result in a good performance. Overall, based on the amount of data collected, clock has the best performance among the three practicable algorithms. Working set algorithm may outperform clock if an optimized combination of refresh time and tau is implemented. Thus the real operating system might use working set as a standard page replacement algorithm.