Atta Ebrahimi

CS1550 Project 3

How to run:

Compile: javac vmsim.java

Run: "java vmsim -n numframes -a opt|clock|nru|aging [-r refresh rate] tracefile"

-r and refresh rate are optional

ex. java vmsim -n 64 -a nru -r 50 gcc.trace

Virtual Memory Simulator

Algorithm Data (gcc.trace):

**OPT:**

|  |  |
| --- | --- |
| **Number of Frames** | 8 |
| **Total memory accesses** | 1,000,000 |
| **Total page faults** | 118,480 |
| **Total writes to disk** | 15,031 |

|  |  |
| --- | --- |
| **Number of Frames** | 16 |
| **Total memory accesses** | 1,000,000 |
| **Total page faults** | 80,307 |
| **Total writes to disk** | 11,316 |

|  |  |
| --- | --- |
| **Number of Frames** | 32 |
| **Total memory accesses** | 1,000,000 |
| **Total page faults** | 55,802 |
| **Total writes to disk** | 8,274 |

|  |  |
| --- | --- |
| **Number of Frames** | 64 |
| **Total memory accesses** | 1,000,000 |
| **Total page faults** | 38,050 |
| **Total writes to disk** | 5,730 |

**NRU (refresh rate = 50):**

|  |  |
| --- | --- |
| **Number of Frames** | 8 |
| **Total memory accesses** | 1,000,000 |
| **Total page faults** | 198,579 |
| **Total writes to disk** | 18,894 |

|  |  |
| --- | --- |
| **Number of Frames** | 16 |
| **Total memory accesses** | 1,000,000 |
| **Total page faults** | 125,989 |
| **Total writes to disk** | 12,463 |

|  |  |
| --- | --- |
| **Number of Frames** | 32 |
| **Total memory accesses** | 1,000,000 |
| **Total page faults** | 117,991 |
| **Total writes to disk** | 10,327 |

|  |  |
| --- | --- |
| **Number of Frames** | 64 |
| **Total memory accesses** | 1,000,000 |
| **Total page faults** | 113,427 |
| **Total writes to disk** | 9778 |

**Clock:**

|  |  |
| --- | --- |
| **Number of Frames** | 8 |
| **Total memory accesses** | 1,000,000 |
| **Total page faults** | 181,856 |
| **Total writes to disk** | 29,401 |

|  |  |
| --- | --- |
| **Number of Frames** | 16 |
| **Total memory accesses** | 1,000,000 |
| **Total page faults** | 121,682 |
| **Total writes to disk** | 16,376 |

|  |  |
| --- | --- |
| **Number of Frames** | 32 |
| **Total memory accesses** | 1,000,000 |
| **Total page faults** | 87,686 |
| **Total writes to disk** | 12,293 |

|  |  |
| --- | --- |
| **Number of Frames** | 64 |
| **Total memory accesses** | 1,000,000 |
| **Total page faults** | 61,640 |
| **Total writes to disk** | 9,346 |

**Random:**

|  |  |
| --- | --- |
| **Number of Frames** | 8 |
| **Total memory accesses** | 1,000,000 |
| **Total page faults** | 232,932 |
| **Total writes to disk** | 40,195 |

|  |  |
| --- | --- |
| **Number of Frames** | 16 |
| **Total memory accesses** | 1,000,000 |
| **Total page faults** | 155,260 |
| **Total writes to disk** | 25,483 |

|  |  |
| --- | --- |
| **Number of Frames** | 32 |
| **Total memory accesses** | 1,000,000 |
| **Total page faults** | 108,150 |
| **Total writes to disk** | 17,611 |

|  |  |
| --- | --- |
| **Number of Frames** | 64 |
| **Total memory accesses** | 1,000,000 |
| **Total page faults** | 75,644 |
| **Total writes to disk** | 12,261 |

With these numbers, it is apparent that the clock algorithm would be best to implement on a computer’s operating system. This is only if the computer has between 8 and 64 frames though. If the computer had more frames, we would have to do more tests.

For NRU, I messed around with the refresh period and found that 50-55 is a great value to work with. Any more or less than that was not as good.