REPORT

Name: Chris Banci
Date: May 9, 2017
Course: CS433 - Operating Systems
Assignment: 5 - Simulation of Page Replacement Algorithms

Description:

This program is an implementation of simulator which simulates various paging replacement algorithms using parameters passed in the command line. The user will be able to set the paging size and memory size. In regards to this, the parameters must be a power of 2. The paging replacement algorithms that will be tested in this simulator are First-In-First-Out (FIFO), Least-Recently-Used (LRU) and Random and will read references from a file.

Furthermore, in order to study the performance of each paging algorithm, this program keep track of all paging faults and all flushes that occurs when the simulation is running, then display the results.

Implementation:

The simulator contains three paging algorithms:

1) **Random** = In this paging algorithm, a random page is replaced in memory.
2) **LRU** = In this paging algorithm, the least recently used page is replaced in memory. LRU keeps track of page usage overtime to achieve this.
3) **FIFO** = In this paging algorithm, the frame that's been in memory the longest is replaced.

With the use of a dirty bit, we can determine if a page has been modified or not. This is will used with paging replacement algorithms.

Analysis:

For the analysis, I have tested each of the three paging algorithms with different page sizes and memory sizes. For page size, i tested: 256, 1mb, 4mb and 8gb. For the memory size, i have tested 1mb, 16mb, 32mb, and 64mb

Paging algorithm	Random
Page size	256
Memory size	1024

Total page faults	4966710
Total flushes	1999946
Elapsed Time	915 ms

Paging algorithm	Random
Page size	1024
Memory size	16384
Total page faults	4999946
Total flushes	1999946
Elapsed Time	916 ms

Paging algorithm	Random
Page size	4096
Memory size	32768
Total page faults	4966979
Total flushes	1999946
Elapsed Time	921 ms

Paging algorithm	Random
Page size	8192
Memory size	65536
Total page faults	4966979
Total flushes	1999946
Elapsed Time	917 ms

Paging algorithm	FIFO

Page size	256
Memory size	1024
Total page faults	4999936
Total flushes	1999936
Elapsed Time	916 ms

Paging algorithm	FIFO
Page size	1024
Memory size	16384
Total page faults	4999936
Total flushes	1999936
Elapsed Time	916 ms

Paging algorithm	FIFO
Page size	4096
Memory size	32768
Total page faults	4966912
Total flushes	1999946
Elapsed Time	832 ms

Paging algorithm	FIFO
Page size	8192
Memory size	65536
Total page faults	4966912
Total flushes	1999946
Elapsed Time	836 ms

Paging algorithm	LRU
Page size	256
Memory size	1024
Total page faults	4967106
Total flushes	1999946
Elapsed Time	843 ms

Paging algorithm	LRU
Page size	1024
Memory size	16384
Total page faults	4967106
Total flushes	1999946
Elapsed Time	930 ms

Paging algorithm	LRU
Page size	4096
Memory size	32768
Total page faults	4967106
Total flushes	1999946
Elapsed Time	871 ms

Paging algorithm	LRU
Page size	8192
Memory size	65536
Total page faults	49667106
Total flushes	1999946

Elapsed Time	896 ms

From the analysis above, it can be concluded that LRU is the best out of the three paging algorithms.

Included Files:

Source:

- main.cpp // driver file
- simulation.cpp // implementation file, contains paging algorithms and other

Header:

- simulation.h // header file for buffer.

Other:

- assign5 // the executable.
- makefile // used to compile the program into an executable.

How to run:

To compile this program, use the makefile which will compile the source files and create an executable called assign4.

To run the executable, enter ***./assign5 <X> <Y>*** in the console.

- ***<X>*** being the size of the page,
- ***<Y>*** being the size o f the memory,

Example:

```
./assign5 256 1024
```

This will set the pagesize to 256 and memory size to 1024.