Seniv Volodymyr 309358

Laboratory 4.

The main idea of the task is to perform a simulation where we are creating a command file that maps 8 pages of physical memory address on each of the 64 virtual pages.

Page fault occurs at the moment when program tries to access a memory block which is not placed in the physical memory (Ram). The page address exists in the table, but the mapping can not be performed because the address of the frame does not exist. It means that this page is allocated in the virtual memory. This fault signals to the operating system that it should allocate a memory (frame) in the Ram, move this page from the virtual memory(HDD, SSD) to the Page table to map this page with the allocated frame. After that this page is moved to the Ram and the execution starts.

Page table Is a table where the pages of the process are mapped with the frames(where the page lies in the Ram) in physical memory. In the case when program requests a page from memory, and the page lies on the storage device, then a page fault is generated. Operating system handles it. While doing it, it chooses a page that is being removed from Ram and places needed page onto free place. After that Page Table is updated.

At first, we edit the memory.conf file to assign the first eight virtual pages to 8 different random physical pages. Then on the commands file, we have 64 READ instructions, so that we can read from each of the 64 virtual pages.

With the run of simulation, the mapping is done as it is written in the input files until it reaches the virtual page 31. After that the page fault caused. No physical pages are bonded with these virtual addresses.

In this simulation the First In First Out (FIFO) algorithm.

Commands:

// Enter READ/WRITE commands into this file

// READ <OPTIONAL number type: bin/hex/oct> <virtual memory address or random>

// WRITE <OPTIONAL number type: bin/hex/oct> <virtual memory address or random>

READ 11386

READ 22383

READ 37141

READ 59601

READ 78117

READ 85765

READ 99924

READ 119460

READ 133556

READ 154951

READ 174278

READ 185627

READ 212108

READ 213915

READ 235100

READ 259602

READ 266951

READ 285726

READ 295471

READ 313990

READ 334896

READ 358839

READ 371307

READ 379050

READ 407997

READ 419199

READ 436136

READ 455435

READ 464743

READ 484808

READ 495559

READ 520154

READ 527247

READ 544486

READ 571445

READ 574648

READ 601959

READ 608242

READ 634464

READ 650334

READ 665303

READ 680123

READ 700084

READ 718045

READ 736765

READ 752113

READ 764461

READ 772474

READ 797201

READ 811811

READ 823332

READ 851304

READ 865084

READ 873704

READ 898206

READ 915878

READ 927862

READ 936529

READ 951949

READ 978808

READ 990300

READ 1008584

READ 1022333

Memory.conf:

```
// memset  virt page #  physical page #  R (read from)  M (modified) inMemTime (ns) lastTo
memset 0 15 0 0 0 0
memset 1 11 0 0 0 0
memset 2 25 0 0 0 0
memset 3 13 0 0 0 0
memset 4 8 0 0 0 0
memset 5 5 0 0 0 0
memset 6 16 0 0 0 0
memset 7 27 0 0 0 0


// enable_logging 'true' or 'false'
// When true specify a log_file or leave blank for stdout
enable_logging true

// log_file <FILENAME>
// Where <FILENAME> is the name of the file you want output
// to be print to.
log_file tracefile

// page size, defaults to 2^14 and cannot be greater than 2^26
// pagesize <single page size (base 10)> or <'power' num (base 2)>
pagesize 16384

// addressradix sets the radix in which numerical values are displayed
// 2 is the default value
// addressradix <radix>
addressradix 16

// numpages sets the number of pages (physical and virtual)
// 64 is the default value
// numpages must be at least 2 and no more than 64
// numpages <num>
numpages 64
```

Tracefile:

READ 2c7a ... okay

READ 576f ... okay

READ 9115 ... okay

READ e8d1 ... okay

READ 13125 ... okay

READ 14f05 ... okay

READ 18654 ... okay

READ 1d2a4 ... okay

READ 209b4 ... okay

READ 25d47 ... okay

READ 2a8c6 ... okay

READ 2d51b ... okay

READ 33c8c ... okay

READ 3439b ... okay

READ 3965c ... okay

READ 3f612 ... okay

READ 412c7 ... okay

READ 45c1e ... okay

READ 4822f ... okay

READ 4ca86 ... okay

READ 51c30 ... okay

READ 579b7 ... okay

READ 5aa6b ... okay

READ 5c8aa ... okay

READ 639bd ... okay

READ 6657f ... okay

READ 6a7a8 ... okay

READ 6f30b ... okay

READ 71767 ... okay

READ 765c8 ... okay

READ 78fc7 ... okay

READ 7efda ... okay

READ 80b8f ... page fault

READ 84ee6 ... page fault

READ 8b835 ... page fault

READ 8c4b8 ... page fault

READ 92f67 ... page fault

READ 947f2 ... page fault

READ 9ae60 ... page fault

READ 9ec5e ... page fault

READ a26d7 ... page fault

READ a60bb ... page fault

READ aaeb4 ... page fault

READ af4dd ... page fault

READ b3dfd ... page fault

READ b79f1 ... page fault

READ baa2d ... page fault

READ bc97a ... page fault

READ c2a11 ... page fault

READ c6323 ... page fault

READ c9024 ... page fault

READ cfd68 ... page fault

READ d333c ... page fault

READ d54e8 ... page fault

READ db49e ... page fault

READ df9a6 ... page fault

READ e2876 ... page fault

READ e4a51 ... page fault

READ e868d ... page fault

READ eef78 ... page fault

READ f1c5c ... page fault

READ f63c8 ... page fault

READ f997d ... page fault