Ahmed Alelg
201507470
ICS-431:Operating Systems
Due Date: 3^{rd} May, 2020

Homework - 5

Term: 192

Theoretical part

Chapter 9: Virtual Memory

Problem 9.9

The page table shown in Figure 9.31 is for a system with 16-bit virtual and physical addresses and with 4,096-byte pages. The reference bit is set to 1 when the page has been referenced. Periodically, a thread zeroes out all values of the reference bit. A dash for a page frame indicates the page is not in memory. The page-replacement algorithm is localized LRU,and all numbers are provided in decimal.

a. Convert the following virtual addresses (in hexadecimal) to the equivalent physical addresses. You may provide answers in either hexadecimal or decimal. Also set the reference bit for the appropriate entry in the page table.

   - `0xE12C`
   - `0x3A9D`
   - `0xA9D9`
   - `0x7001`
   - `0xACA1`

b. Using the above addresses as a guide, provide an example of a logical address (in hexadecimal) that results in a page fault.

c. From what set of page frames will the LRU page-replacement algorithm choose in resolving a page fault?

Page	Page Frame	Reference Bit
0	9	0
1	1	0
2	14	0
3	10	0
4	–	0
5	13	0
6	8	0
7	15	0
8	–	0
9	0	0
10	5	0
11	4	0
12	–	0
13	–	0
14	3	0
15	2	0

Solution:

a. We have $2^{16}/2^{12} = 2^4 = 16$ pages and frames. Hence the first 4 bits of every address are used for address translation as follows:

   - `0xE12C` $\xrightarrow{E:3}$ `0x312C`
   - `0x3A9D` $\xrightarrow{3:A}$ `0xAA9D`
   - `0xA9D9` $\xrightarrow{A:5}$ `0x59D9`
   - `0x7001` $\xrightarrow{7:F}$ `0xF001`
   - `0xACA1` $\xrightarrow{A:5}$ `0x5CA1`

Finally, the reference bit will be as follows:

page	ref bit
1	0
2	0
3	1
4	0
5	1
6	0
7	0
8	0
9	0
10	1
11	0
12	0
13	0
14	0
15	1

b. `0x4200`

c. $[1, 2, 4, 6, 7, 8, 9, 11, 12, 13, 14, 3, 15, 10]$

Problem 9.12
Discuss situations in which the most frequently used (MFU) page-replacement algorithm generates fewer page faults than the least recently used (LRU) page-replacement algorithm. Also discuss under what circumstances the opposite holds.

Solution: In MFU, a counter is kept for each page reference, and swipes out the pages with the highest count. While LRU maintains a register to timestamp each page when accessed and swipes pages with the most oldest timestamp. For a memory of size 3 and sequence $1, 2, 3, 2, 4, 2$, LRU will swipe 1 to bring 4, and referencing 2 will not cause a page fault. but MFU will swipe 2 to bring 4 and a page fault will occur when referenceing 2. Conversly, with $1, 2, 3, 2, 4, 1$ LRU will cause page fault, while MFU will not.

Chapter 10: File system

Problem 10.3
What are the advantages and disadvantages of providing mandatory locks instead of advisory locks whose use is left to users' discretion?

Solution:

- **Advantages:** File consistency is ensured by the OS this is particualry important for critical systems such as banks, and the overhead for the programmers is removed.
- **Disadvantages:** OS needs to check every open/read/write system calls and keeps track of opened files and their associated processes.

Problem 10.8
Discuss the advantages and disadvantages of supporting links to files that cross mount points (that is, the file link refers to a file that is stored in a different volume).

Solution:

- **Advantages:** Increase usability since the user can use short relative paths, and saves more space in the hardisk.
- **Disadvantages:** OS needs to keeps track and checks if a file location has changed, it also needs to change it in the link.

Chapter 11: Implementing File Systems

Problem 11.6
Discuss how performance optimizations for file systems might resultin difficulties in maintaining the consistency of the systems in the event of computer crashes.

Solution: Consider the cached and not yet written information (e.g partition table, directory structre, etc...) in the main memory. if the system crashed then they will be lost and hence consistancy is violated.

Problem 11.10
Assume that in a particular augmentation of a remote-file-access pro-tocol, each client maintains a name cache that caches translations fromfile names to corresponding file handles. What issues should we take into account in implementing the name cache?

Solution: Files could be renamed or deleted; which could case consistancy issues in translation.

Chapter 12: Mass-Storage Structure

Problem 12.10
Compare the throughput achieved by a RAID level 5 organization with that achieved by a RAID level 1 organization for the following:

a. Read operations on single blocks

b. Read operations on multiple contiguous blocks

Solution:

a. RAID 1 preforms better because it can run two threads to fetch the block from the two hardisks, while RAID 5 will need to bring the parity block as well.

b. RAID 5 preforms better, since there are 3 or more disks running at the same time to grab the contiguous blocks.

Problem 12.16
Write a program that implements the following disk-scheduling algorithms:

a. FCFS

b. SSTF

c. SCAN

d. C-SCAN

e. LOOK

f. C-LOOK

Your program will service a disk with 5,000 cylinders numbered 0 to 4,999. The program will generate a random series of 1,000 cylinderrequests and service them according to each of the algorithms listedabove. The program will be passed the initial position of the disk head(as a parameter on the command line) and report the total amount ofhead movement required by each algorithm

Solution: Check the attachments for the source code.

```
Request sequence [4602, 2858, 648, 4553, 589, 2610, 4439, 192, 4053, 2920] ...
start head 2006
   fcfs
Total movements:  1568927
----------
   sstf
Total movements:  779080
----------
   scan
Total movements:  7000
----------
  cscan
Total movements:  4997
----------
   look
Total movements:  7982
----------
  clook
Total movements:  4987
----------
```

Figure 1: Result for a random sequence between 0 and 4999 of length 1000

```
----------
   fcfs
----------
 53 --- 45 --►  98
 98 --- 85 --► 183
183 ◄---146 --- 37
 37 --- 85 --► 122
122 ◄---108 --- 14
 14 ---110 --► 124
124 ◄--- 59 --- 65
 65 --- 2  --►  67
----------
Total movements:  640
```

(a) FCFS

```
----------
   sstf
----------
 53 ---12---► 65
 65 ---2 ---► 67
 67 ◄---30--- 37
 37 ◄---23--- 14
 14 ---84---► 98
 98 ---24---► 122
122 ---2 ---► 124
124 ---59---► 183
----------
Total movements:  236
```

(b) sstf

```
----------
   scan
----------
 53 ◄---16--- 37
 37 ◄---23--- 14
 14 ---51---► 65
 65 ---2 ---► 67
 67 ---31---► 98
 98 ---24---► 122
122 ---2 ---► 124
124 ---59---► 183
----------
Total movements:  208
```

(c) scan (elevator)

```
----------
  cscan
----------
[37, 14] [16, 75, 77, 101, 132, 134]
 53 ◄---16--- 37
 37 ◄---23--- 14
 14 ◄---14--- 0
  0 ---16---► 16
 16 ---59---► 75
 75 ---2 ---► 77
 77 ---24---► 101
101 ---31---► 132
132 ---2 ---► 134
----------
Total movements:  187
```

(d) cscan

```
----------
   look
----------
 53 --- 12 --►  65
 65 --- 2  --►  67
 67 --- 31 --►  98
 98 --- 24 --► 122
122 --- 2  --► 124
124 --- 59 --► 183
183 ◄---146 --- 37
 37 ◄--- 23 --- 14
----------
Total movements:  299
```

(e) look

```
----------
  clook
----------
 53 ---12---► 65
 65 ---2 ---► 67
 67 ---31---► 98
 98 ---24---► 122
122 ---2 ---► 124
124 ---59---► 183
183 ---14---► 197
197 ---23---► 220
----------
Total movements:  167
```

(f) clook

Figure 2: Testing for the sequence $98, 183, 37, 122, 14, 124, 65, 67$ with initial position of 53.

Theoretical part

Chapter 9: Virtual Memory Management

Problem 9.26
Write a program that implements the FIFO, LRU, and optimal page-replacement algorithms presented in this chapter. First, generate a random page-reference string where page numbers range from 0 to 9. Apply the random page-reference string to each algorithm, and record the number of page faults incurred by each algorithm. Implement the replacement algorithms so that the number of page frames can vary from 1 to 7. Assume that demand paging is used

Solution: *Check the attachments for the source code.*

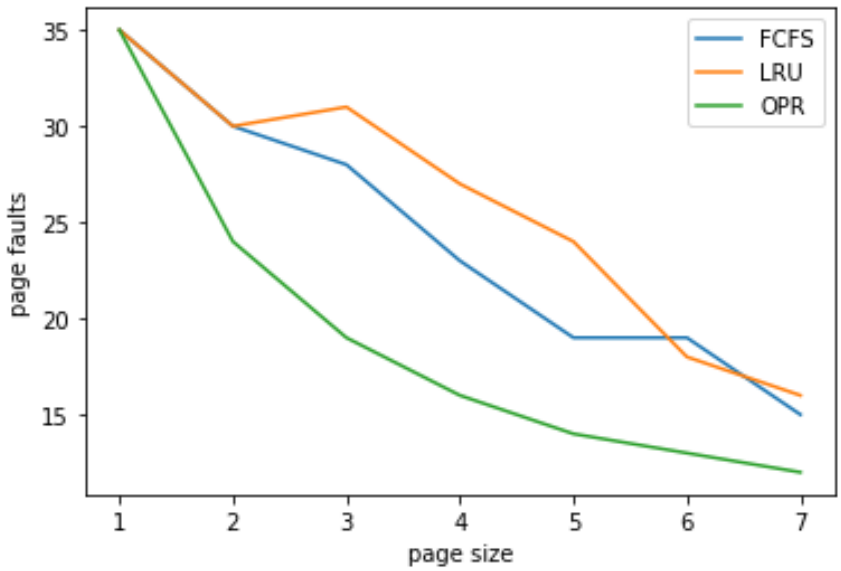

Figure 3: Simulation of sequence of length 20 between 0 and 9, varying the size between 1 and 7

```
n =  1
fcfs:  35
lru:  35
opr:  35
----------
n =  2
fcfs:  30
lru:  30
opr:  24
----------
n =  3
fcfs:  28
lru:  31
opr:  19
----------
n =  4
fcfs:  23
lru:  27
opr:  16
----------
n =  5
fcfs:  19
lru:  24
opr:  14
----------
n =  6
fcfs:  19
lru:  18
opr:  13
----------
n =  7
fcfs:  15
lru:  16
opr:  12
----------
```

Figure 4

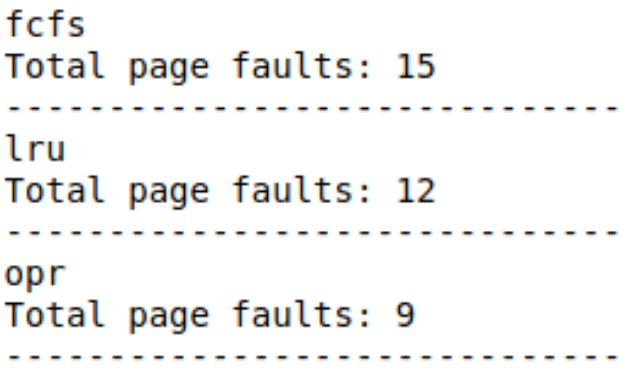

Figure 5: Testing on $7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1$ with size of 3