

COMP 304 - Operating Systems

Assignment 3

Dilara Deveci

0068182

Problem 1

Contiguous Allocation suffers from external fragmentation when total memory space exists to satisfy a request, but it is not contiguous. It does not enable code sharing among processes since each process has an independent memory space.

Segmentation suffers from external segmentation as memory as processes are loaded and removed from the memory, the free memory space is broken into little pieces. It enables code sharing among processes if two or more processes share the same code segment.

Paging does not suffer from external fragmentation as the memory allocated does not have to be contiguous. It enables code sharing among processes if two or more processes share the same page of code.

Problem 2

32-bit virtual address

page size: 8 KB

- a) Page offset = 8 KB = 2^{13} -> 13 bit page offset
Virtual page number = $32 - 13 = 19$ bits
No. of entries in Page Table = 2^{19}
Total size = $2^{19} * 4 = 2$ MB
- b) 28-bit physical address
 2^{28} pages, thus physical memory size = $2^{28} = 32$ MB
- c) No. of entries in page table = Virtual address space size / page size
 $28 - 13 = 15$, so 2^{15} pages.

Problem 3

LRU:

1	2	3	4	2	1	5	2	1	2	3
1	1	1	4		4	5				5
	2	2	2		2	2				3
		3	3		1	1				1

No. of page faults: 7

FIFO:

1	2	3	4	2	1	5	2	1	2	3
1	1	1	4		4	4	2			2
	2	2	2		1	1	1			3
		3	3		3	5	5			5

No. of page faults: 8

Optimal:

1	2	3	4	2	1	5	2	1	2	3
1	1	1	1			1				1
	2	2	2			2				2
		3	4			5				3

No. of page faults: 6

Problem 4

a) P1:

Working Set: {1, 2, 3, 4, 7}

Working Set Size: 5

P2:

Working Set: {3, 4, 5, 6}

Working Set Size: 4

P3:

Working Set: {1, 2, 7, 8, 9}

Working Set Size: 5

b) Systems suffers from trashing. In order to suffer from trashing $D = \text{total of WWS}$ should be greater than total number of frames.

$$D = 5+4+5 = 14$$

Total number of frames = 10.

$14 > 10$, so suffers from trashing.

c) At time 7:

$$WS_1 = \{1, 2, 3, 4, 5, 7\}$$

$$WSS_1 = 6$$

$$WS_2 = \{1, 3, 4\}$$

$$WSS_2 = 3$$

$$WS_3 = \{2, 3, 7, 8, 9\}$$

$$WSS_3 = 5$$

$$D = 6+3+5 = 14$$

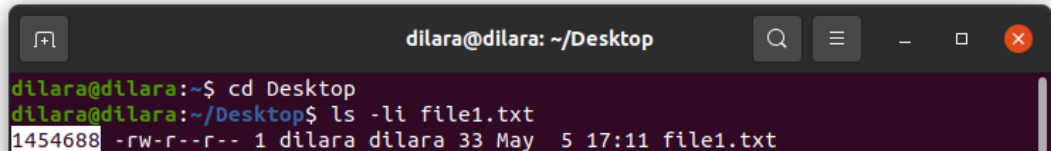
So number of frames should be 14 so that the system does not suffer from trashing at time 7.

Problem 5

**Answers for both part a and b (screenshots and the answers for the questions) are provided simultaneously below.

When I run the command “ls -li file1.txt”, this is obtained.

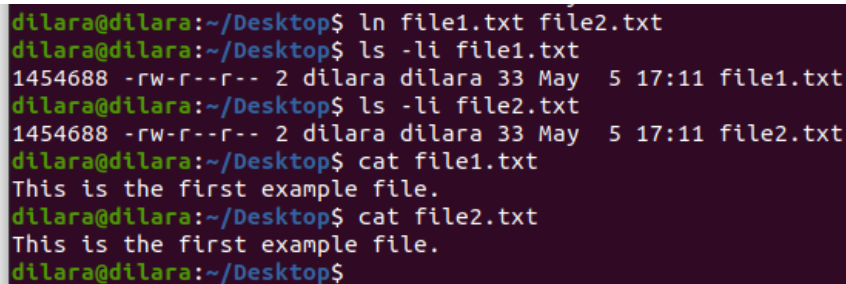
The inode number is **1454688**



```
dilara@dilara: ~/Desktop
dilara@dilara:~$ cd Desktop
dilara@dilara:~/Desktop$ ls -li file1.txt
1454688 -rw-r--r-- 1 dilara dilara 33 May  5 17:11 file1.txt
```

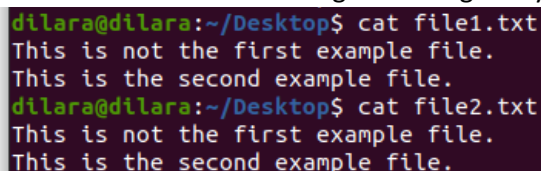
After that, “ln file1.txt file2.txt” command is run in order to create a hard link between file1.txt and file2.txt and this is the output on terminal.

The inode values of file1.txt and file2.txt are 1454688; they are both the same. And the content of the files are same as can be seen from the terminal screenshot provided below.



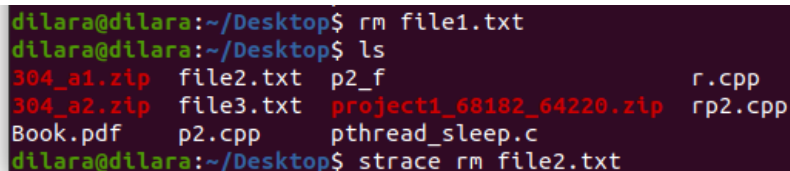
```
dilara@dilara:~/Desktop$ ln file1.txt file2.txt
dilara@dilara:~/Desktop$ ls -li file1.txt
1454688 -rw-r--r-- 2 dilara dilara 33 May  5 17:11 file1.txt
dilara@dilara:~/Desktop$ ls -li file2.txt
1454688 -rw-r--r-- 2 dilara dilara 33 May  5 17:11 file2.txt
dilara@dilara:~/Desktop$ cat file1.txt
This is the first example file.
dilara@dilara:~/Desktop$ cat file2.txt
This is the first example file.
dilara@dilara:~/Desktop$
```

After editing the content of file2.txt (Changing it into “This is not the first example file. This is the second example file.”), content of both files can be seen below. Their content is the same. Both of them have changed although only file2.txt’s content was modified.



```
dilara@dilara:~/Desktop$ cat file1.txt
This is not the first example file.
This is the second example file.
dilara@dilara:~/Desktop$ cat file2.txt
This is not the first example file.
This is the second example file.
```

Next, the “rm file1.txt” command is run to remove file1.txt. After the deletion, the file2.txt still exists. The files on the Desktop can be seen here; there is no file1.txt but there is file2.txt.



```
dilara@dilara:~/Desktop$ rm file1.txt
dilara@dilara:~/Desktop$ ls
304_a1.zip  file2.txt  p2_f  r.cpp
304_a2.zip  file3.txt  project1_68182_64220.zip  rp2.cpp
Book.pdf   p2.cpp    pthread_sleep.c
dilara@dilara:~/Desktop$ strace rm file2.txt
```

Afterwards, file2.txt is removed by entering the command “strace rm file2.txt” command. As it can be from the screenshot below, the system call is used for removing file2.txt is the one with “execve(“/usr/bin/rm”);”; the first line

Next, I created a soft link to file3.txt by entering the “ln -s file3.txt file4.txt” command. Then, the inodes of the files; file3.txt and file4.txt is obtained with the “ls -li file*.txt” command

```
dilara@dilara:~/Desktop$ ln -li file*.txt
ln: invalid option -- 'l'
Try 'ln --help' for more information.
dilara@dilara:~/Desktop$ ln -s file3.txt file4.txt
dilara@dilara:~/Desktop$ ls -li file*.txt
1454686 -rw-r--r-- 1 dilara dilara 34 May 5 17:11 file3.txt
1454702 lrwxrwxrwx 1 dilara dilara 9 May 28 15:03 file4.txt -> file3.txt
```

After that, when the content of file4.txt is edited (Changed into “This is the file for soft links. This is file4”), the content of file3.txt is also altered. It can be seen on the screenshot provided below.

```
dilara@dilara:~/Desktop$ cat file3.txt
This is the file for soft links.
This is file4
dilara@dilara:~/Desktop$ cat file4.txt
This is the file for soft links.
This is file4
dilara@dilara:~/Desktop$ rm file3.txt
dilara@dilara:~/Desktop$
```

Lastly, once the file3.txt is deleted, file4.txt does not exist anymore. when I attempted to edit file4.txt an error was raised. Here is the error.

