

COMP 304- Operating Systems: Assignment 3

Due: June 8, 11.45 am

Notes: This is an individual assignment. No late assignment will be accepted. Please submit your answers through blackboard and bring a hard copy to exam. This assignment is worth 3% of your total grade.

Problem 1

(10 points) Consider the memory management methods of contiguous allocation, paging, and segmentation. Compare these methods with respect to the following: external fragmentation and code sharing among processes.

Problem 2

(10 points) A system with 40-bit logical (virtual) address uses a two-level page table. Logical addresses are split into an 12-bit top-level page table field, a 16-bit second-level page table field, and an offset.

What is the size of a page in this system (assume that the size of a memory location is 1 Byte)? How many pages are there in the address space? Explain the reason.

Problem 3

(10 points) Consider the following page reference string:

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6

How many page faults would occur for the page replacement algorithms: LRU, FIFO, Optimal when 6 frames are allocated for a process? Remember all frames are initially empty, so your first unique pages will all cost one fault each.

Problem 4

(20 points) Create two simple files file1.txt and file3.txt. Add some content to both and answer the following questions. Next, obtain the inode number of this file with the command

```
ls -li file1.txt
```

This will produce output similar to the following:

```
16980 -rw-r--r- 2 os os 22 May 2018 16:13 file1.txt
```

where the inode number is boldfaced. (The inode number of file1.txt is likely to be different on your system.)

The UNIX `ln` command creates a link between a source and target file. This command works as follows:

```
ln [-s] <source file><target file>
```

UNIX provides two types of links: (1) hard links and (2) soft links. Enter the following command to create a hard link between `file1.txt` and `file2.txt`:

```
ln file1.txt file2.txt
```

a) What are the inode values of `file1.txt` and `file2.txt`? Are they the same or different? Do the two files have the same or different contents?

b) Next, edit `file2.txt` and change its contents. After you have done so, examine the contents of `file1.txt`. Are the contents of `file1.txt` and `file2.txt` the same or different? Check their inode numbers again. Are they the same?

Next, enter the following command which removes `file1.txt`:

```
rm file1.txt
```

c) Does `file2.txt` still exist as well?

Now examine the man pages for both the `rm` and `unlink` commands. Afterwards, remove `file2.txt` by entering the command

```
strace rm file2.txt
```

d) The `strace` command traces the execution of system calls as the command `rm file2.txt` is run. What system call is used for removing `file2.txt`?

Now create a soft link to `file3.txt` by entering the following command:

```
ln -s file3.txt file4.txt
```

After you have done so, obtain the inode numbers of `file3.txt` and `file4.txt` using the command

```
ls -li file*.txt
```

e) Are the inodes the same, or is each unique?

f) Next, edit the contents of `file4.txt`. Have the contents of `file3.txt` been altered as well?

g) Last, delete `file3.txt`. After you have done so, explain what happens when you attempt to edit `file4.txt`.

h) Explain how hard link and soft link work based on these experiments. Where each can be used?