CS 3423 Operating Systems
Fall Semester 2017
Prof. Pai H. Chou

Assignment 10 - Part One

Due Date: Sunday, December 3, 2017, 11:59pm
Up to one-day late submission without penalty
Up to one-week late submission with 20% penalty
Submit electronically on iLMS

What to submit: One zip file named <studentID>-`hw10.zip` (replace <studentID> with your own student ID). It should contain four files:

- one PDF file named **hw10.pdf** for Section 1. Write your answers in **English**. Check your spelling and grammar. Include your name and student ID!
- The programming assignment will be posted separately with its own due date.

1. [30 points] Problem Set

1. [20 points] **11.2** Contrast the performance of the three techniques for allocating disk blocks (contiguous, linked, and indexed) for both sequential and random file access. **You must elaborate to receive full credit.**

2. [10 points] **11.8** Consider a file system that uses inodes to represent files. Disk blocks are 8 KB in size, and a pointer to a disk block requires 4 bytes. This file system has 12 direct disk blocks, as well as single, double, and triple indirect disk blocks. What is the maximum size of a file that can be stored in this file system? **You must show your calculation to receive credit.**

2. [30 points] Programming Problems

Strictly speaking, this is not really a programming problem, but more like an interactive experimentation.

11.13 [modified preparation instruction] Before starting this problem, create two text files named `file1.txt, file3.txt` (but **not** `file2.txt`!!) in a Unix or Linux-like system (i.e., uses inodes in its file system) with unique contents. 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 Sep 14 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. A hard link creates a separate target file that has the same inode as the source file. Enter the following command to create a hard link between `file1.txt` and `file2.txt`:

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

2.1 [5 points]

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?

2.2 [5 points]

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?

2.3 [5 points]

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

```
rm file1.txt
```

Does `file2.txt` still exist as well?

2.4 [5 points]

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

```
strace rm file2.txt
```

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`?

2.5 [10 points]

A soft link (or symbolic link) creates a new file that “points” to the name of the file it is linking to. In the source code available with this text, 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
```

Are the inodes the same, or is each unique? Next, edit the contents of `file4.txt`. Have the contents of `file3.txt` been altered as well? Last, delete `file3.txt`. After you have done so, explain what happens when you attempt to edit `file4.txt`.

3. Programming Exercise - to be posted separately with its own due date