Lab 10b/11a/11b – Inodes and Protection in Linux

Operating Systems CS SH3 Term 2, Winter 2018

Dr. Neerja Mhaskar

Labs that are not scheduled for a Lab Test are not mandatory. These are practice labs, designed to help you on your assignments.

Lab Format: The practice labs will be posted a day before or on the day of the lab on the course website. You can choose to solve it beforehand and come in with your solutions and check the correctness of your solution with your TA.

The TAs will also be available to answer any questions you might have on your assignments.

Solutions to practice labs will not be posted online.

Introduction to the lab

In these practice labs you will study the file control blocks called **inodes** in UNIX and LINUX systems. Each inode stores the attributes and disk block location(s) of the files and directories on the system.

Obtaining file attributes using ls –li or stat command

1. Create a simple text file named file1.txt.
2. Create the content “This is file 1” in the file and save it.
3. Obtain the inode number and other few file attributes for file1.txt with the following command:

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

4. On my machine this produces output similar to the following:

```
400740 -rw-r--r-- 1 oscreader oscreader 15 Mar 24 18:52
file1.txt,
```

where the number in bold is the inode number of the file. Note: The inode number of file1.txt is likely to be different on your system.

5. You can obtain detailed file attributes and the inode number with the **stat** command.

```
stat file1.txt
```

6. This produces output similar to the following:

```
File: 'file1.txt'
Size: 15          Blocks: 16     IO Block: 4096   regular file
Device: 801h/2049d     Inode: 400740      Links: 1
Access: (0644/-rw-r--r--)  Uid: ( 1000/oscreader)
Gid: (   1000/oscreader)
Access: 2017-03-24 18:52:30.776000000 -0600
Modify: 2017-03-24 18:52:30.628000000 -0600
Change: 2017-03-24 18:52:30.760000000 -0600
Birth: -
```

About **`Script`** command

`script` command is used to take a copy of everything which is output to the terminal and place it in a log file. The script command should be followed by the name of the log file (e.g. **`script`** output.txt). **`exit`** command stops the logging to a file initiated by the **`script`** command and closes the file.

Outline of the practice labs

You are to write a C program that takes the text file **`file1.txt`** you created before, as a command line argument and outputs the following file attributes stored in its inode.

1. Inode number
2. File size
3. Blocks
4. User ID
5. File permissions (in the same format as seen in output of stat command)
6. Time of last access
7. Time of last data modification
8. Last status change time

After you have your program working, you are to log the output of the following commands using the **`script`** command. (See sample output.txt file.)

1. **`gcc –o lab10b11a11b lab10b11a11b.c`**
2. **`./ lab10b11a11b file1.txt`**
3. **`chmod`** command to change the permissions on the file1.txt file. In particular you are to change the permissions on the file 'file1.txt' as follows:
    1. Owner/user – Read, Write
    2. Group – Read, Write
    3. Others – Read
4. **`./ lab10b11a11b file1.txt`**
5. **`stat file1.txt`**

Note: see lecture notes on chapter 11 and 12 to use `chmod` command.

C program - In particular your C program needs to do the following:

1. Use the **`stat`** data structure to output file attributes. More information about stat structure can be found at: **http://pubs.opengroup.org/onlinepubs/7908799/xsh/sysstat.h.html**

2. To be able to use the stat structure add the following header files:

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h> /*Contains the stat structure
definition*/
```

3. Declare a variable of type **`stat`**: E.g.:
   `struct stat fileAttributes;`

4. Use the following members of **`stat`** to print the required file attributes.

   a. Inode: `fileAttributes.st_ino`
   b. Size (in bytes): `fileAttributes.st_size`
   c. Blocks: `fileAttributes.st_blocks`
   d. File Permissions: `fileAttributes.st_mode`
   e. Uid: `fileAttributes.st_uid`
   f. Time of last access: `fileAttributes.st_atime`
   g. Time of last data modification: `fileAttributes.st_mtime`
   h. Last Status Change time: `fileAttributes.st_ctime`

   To print the file permissions in the format as seen in the output of **`stat`** command use the file mode bits **`S_ISDIR(), S_IRUSR`** and so on (see the link- **http://pubs.opengroup.org/onlinepubs/7908799/xsh/sysstat.h.html** for more information on file mode bits): Sample code:

```
fileAttributes.st_mode & S_ISDIR()
fileAttributes.st_mode & S_IRUSR and so on.
```

5. Compile your program without errors.
6. Show your output.txt file to your TA.

Sample output recorded by script command:

See output.txt file posted on Avenue