HW6: Linux Virtual File System

- HW6: 150 points total, group project
- Submission notes
  - HW6 will be submitted in a single gzipped tarball named `group_xx_hw6.tar.gz` where xx is your two digit group number (with leading zeros, if applicable).
  - The top level directory in the tarball is `hw6`, which contains a git repo. You will track the following files in the repo:

    ```
    README.txt
    Makefile
    lwnfs.c
    ```

  - Just like you did in HW5, please list git commit IDs for each part.

Part 1: Creating and mounting a file system on a loop device (10 points)

A loop device is a pseudo-device that makes a file accessible as a block device. Files of this kind are often used for CD ISO images. Mounting a file containing a file system via such a loop mount makes the files within that file system accessible.

Tasks

1. Create a loop device, build & mount an ext2 filesystem, and try creating directories and files. Below is a sample session you can follow. It goes without saying that you need to understand what's going on at each step. Look at the man pages. Google stuff. You may need to boot into the LTS kernel for this to work.

   ```
   jae@vm04 ~/tmp $ sudo su
   vm04 /home/jae/tmp # dd if=/dev/zero of=./ext2.img bs=1024 count=100
   100+0 records in
   100+0 records out
   102400 bytes (102 kB) copied, 0.000626388 s, 163 MB/s
   vm04 /home/jae/tmp # losetup --find --show ext2.img
   /dev/loop0
   vm04 /home/jae/tmp # mkfs -t ext2 /dev/loop0
   mke2fs 1.42.9 (28-Dec-2013)
   Filesystem label=
   OS type: Linux
   Block size=1024 (log=0)
   Fragment size=1024 (log=0)
   Stride=0 blocks, Stripe width=0 blocks
   16 inodes, 100 blocks
   5 blocks (5.00%) reserved for the super user
   First data block=1
   1 block group
   8192 blocks per group, 8192 fragments per group
   16 inodes per group

   Allocating group tables: done
   Writing inode tables: done
   Writing superblocks and filesystem accounting information: done

   vm04 /home/jae/tmp # mkdir mnt
   vm04 /home/jae/tmp # mount /dev/loop0 ./mnt
   vm04 /home/jae/tmp # cd mnt
   vm04 /home/jae/tmp/mnt # ll
   total 17
   drwxr-xr-x 3 root root   1024 Apr 20 11:19 ./
   drwxr-xr-x 5 jae  users  4096 Apr 20 11:20 ../
   drwx------ 2 root root  12288 Apr 20 11:19 lost+found/
   vm04 /home/jae/tmp/mnt # mkdir sub2
   vm04 /home/jae/tmp/mnt # ll
   total 18
   drwxr-xr-x 4 root root   1024 Apr 20 11:26 ./
   drwxr-xr-x 5 jae  users  4096 Apr 20 11:20 ../
   drwx------ 2 root root  12288 Apr 20 11:19 lost+found/
   drwxr-xr-x 2 root root   1024 Apr 20 11:26 sub2/
   vm04 /home/jae/tmp/mnt # cd sub2
   vm04 /home/jae/tmp/mnt/sub2 # ll
   total 2
   drwxr-xr-x 2 root root 1024 Apr 20 11:26 ./
   drwxr-xr-x 4 root root 1024 Apr 20 11:26 ../
   vm04 /home/jae/tmp/mnt/sub2 # mkdir sub2.1
   vm04 /home/jae/tmp/mnt/sub2 # ll
   total 3
   drwxr-xr-x 3 root root 1024 Apr 20 11:27 ./
   drwxr-xr-x 4 root root 1024 Apr 20 11:26 ../
   drwxr-xr-x 2 root root 1024 Apr 20 11:27 sub2.1/
   vm04 /home/jae/tmp/mnt/sub2 # touch file2.1
   vm04 /home/jae/tmp/mnt/sub2 # ll
   total 3
   drwxr-xr-x 3 root root 1024 Apr 20 11:28 ./
   drwxr-xr-x 4 root root 1024 Apr 20 11:26 ../
   -rw-r--r-- 1 root root    0 Apr 20 11:28 file2.1
   drwxr-xr-x 2 root root 1024 Apr 20 11:27 sub2.1/
   vm04 /home/jae/tmp/mnt/sub2 # cd ../../
   vm04 /home/jae/tmp # umount mnt
   vm04 /home/jae/tmp # losetup --find
   /dev/loop1
   vm04 /home/jae/tmp # losetup --detach /dev/loop0
   vm04 /home/jae/tmp # losetup --find
   /dev/loop0
   vm04 /home/jae/tmp # ll mnt/
   total 8
   drwxr-xr-x 2 root root  4096 Apr 20 11:20 ./
   drwxr-xr-x 5 jae  users 4096 Apr 20 11:20 ../
   vm04 /home/jae/tmp # exit
   exit
   jae@vm04 ~/tmp $
   ```

2. In the sample session shown above, files and directories are created. Make sure you see the number of links each file or directory has, and make sure you understand why.

   Also try creating some hard links and symlinks. Make sure you understand how they affect the link counts.

3. Assuming you built your 4.1.18-UNI kernel with a reduced `.config` using `make localmodconfig`, you won't be able to create and mount a loop device. Identify the kernel config option(s) required for creating and mounting loop devices. Rebuild your 4.1.18-UNI kernel and try running through the session again.

Deliverables

In your README.txt, write the following:

1. A statement that you understand what inode is, what link is, and how link counts are affected by subdirectories and hard links.
2. Kernel config option(s) required for using loop devices.

Part 2: Learning Linux VFS (20 points)

Linux VFS tutorial

1. Get up to date on the reading assignments.
2. Go through the Linux VFS tutorial at lwn.net.

   Git-clone the tutorial's source code, and read the code along with the tutorial. You need to understand how the code works.

   ```
   git clone  https://gist.github.com/3496839.git  hw6
   ```

3. The git repo is the latest version of the code for the tutorial, but unfortunately it does not work on our 4.1.18 kernel. We will fix it later. For now, you can refer to the following sample session as you read the article and code.

   ```
   jae@vm04 ~/kb/lwnfs-jae $ mkdir mnt
   jae@vm04 ~/kb/lwnfs-jae $ sudo insmod lwnfs.ko
   jae@vm04 ~/kb/lwnfs-jae $ sudo mount -t lwnfs none ./mnt
   jae@vm04 ~/kb/lwnfs-jae $ sudo su
   vm04 /home/jae/kb/lwnfs-jae # cd mnt
   vm04 /home/jae/kb/lwnfs-jae/mnt # ll
   total 4
   drwxr-xr-x 1 root root     0 Apr 18 19:35 ./
   drwxr-xr-x 5 jae  users 4096 Apr 18 19:35 ../
   -rw-r--r-- 1 root root     0 Apr 18 19:35 counter
   drw-r--r-- 1 root root     0 Apr 18 19:35 subdir/
   vm04 /home/jae/kb/lwnfs-jae/mnt # ll subdir/
   total 0
   drw-r--r-- 1 root root 0 Apr 18 19:35 ./
   drwxr-xr-x 1 root root 0 Apr 18 19:35 ../
   -rw-r--r-- 1 root root 0 Apr 18 19:35 subcounter
   vm04 /home/jae/kb/lwnfs-jae/mnt # cat counter
   0
   vm04 /home/jae/kb/lwnfs-jae/mnt # cat counter
   1
   vm04 /home/jae/kb/lwnfs-jae/mnt # cat counter
   2
   vm04 /home/jae/kb/lwnfs-jae/mnt # echo 1000000 > counter
   vm04 /home/jae/kb/lwnfs-jae/mnt # cat counter
   1000000
   vm04 /home/jae/kb/lwnfs-jae/mnt # cat counter
   1000001
   vm04 /home/jae/kb/lwnfs-jae/mnt # cat counter
   1000002
   vm04 /home/jae/kb/lwnfs-jae/mnt # echo 2000 > subdir/subcounter
   vm04 /home/jae/kb/lwnfs-jae/mnt # cat subdir/subcounter
   2000
   vm04 /home/jae/kb/lwnfs-jae/mnt # cat subdir/subcounter
   2001
   vm04 /home/jae/kb/lwnfs-jae/mnt # cat subdir/subcounter
   2002
   vm04 /home/jae/kb/lwnfs-jae/mnt # exit
   exit
   jae@vm04 ~/kb/lwnfs-jae $ sudo umount mnt
   jae@vm04 ~/kb/lwnfs-jae $ ll mnt/
   total 8
   drwxr-xr-x 2 jae users 4096 Apr 18 19:35 ./
   drwxr-xr-x 5 jae users 4096 Apr 18 19:35 ../
   jae@vm04 ~/kb/lwnfs-jae $ sudo rmmod lwnfs
   jae@vm04 ~/kb/lwnfs-jae $ lsmod | grep lwn
   jae@vm04 ~/kb/lwnfs-jae :( $
   ```

Fixing it for the latest kernel

Your job in this part is to fix the code so that you can recreate the shell session shown above.

1. First you need to write a Makefile for the lwnfs module. That's easy.
2. The lwnfs module does not build on our kernel version. Fix it.
3. Even after you fix the build problem, you will notice that you are not able to mount the file system as shown in the shell session. This is due to a bug in the code. Fix it.

You are welcome to google for info. The `linux-4.1.18/Documentation` directory contains a wealth of information as well.

The fixes are very simple once you find them. Please refrain from asking or answering direct questions about them in the mailing list. The whole point of the assignment is for you to test yourself: starting from scratch, not even knowing where to look, will you be able to wade through documentations and code, search the net, find where the problem is, and figure out how to fix it?

Commit your fixes to your repo. Describe your fixes in your `README.txt`.

Part 3: Fixing link counts in lwnfs (20 points)

The `lwnfs` file system you studied and fixed in part 2 creates a subdirectory named "subdir" and two counter files, "counter" and "subcounter". Unfortunately the lwnfs code is sloppy about link counts.

Deliverables

- Fix the code so that you have correct link counts on the directories and files.

Part 4: Implementing `mkdir` in lwnfs (60 points)

Modify the lwnfs code so that the filesystem supports `mkdir` commands. That is, after you mount a lwnfs filesystem, you should be able to create any directory hierarchy under the root directory by `cd` ing and `mkdir` ing. Test it with the `ll` command (or `ls -alF` if you don't have it aliased.)

The description for this part is deliberately terse. You will need to google, read book chapters & kernel documentations, and read a fair amount of kernel code. The amount of code you need to write isn't much, but you will have to gain a pretty solid understanding of VFS in order to figure out what to do.

As you read through kernel code, you will see a lot of locking going on. You are not required to implement locking correctly. (The original lwnfs code completely ignores it too.)

Part 5: Creating files (40 points)

After you implement `mkdir`, implement file creation. Under any directory, you should be able to create a new file like this:

```
touch  new_file_name
```

The new file is a counter, just like the original "counter" and "subcounter". It behaves the same way – i.e., every time the file is read, it will return 1 + the previous number, starting from 0.

You can also create a new file initialized to a number other than 0 like this:

```
echo 12345 > another_new_file_name
```

Remember that each file keeps its own counter.

Good luck!

Last updated: 2016-04-07