

CSIS0230A Principle of Operating Systems(Class A)

Tutorial 10

Examining an ext2 filesystem

In this tutorial, we will use a utility called `lde` (Linux Disk Editor) to examine the contents of a disk partition, and tries to recover a deleted file in it. To be on the safe side, we will not use a real partition, but instead use a regular file for that purpose.

1. Preparation

To begin with the tutorial, login as root, and remain in the home directory of root. Now do the following to create and populate a filesystem on the file `test.img`:

1. Create a file `test.img` of size 40M, containing all zeros. This can be done by using the `dd` command, copying from the device `/dev/zero` with a block size of 1M and a count of 40. See the man page of `dd` for details.
2. Use `mke2fs` to create a filesystem in `test.img`.
3. Create a directory `testmnt`, and mount the newly created filesystem into this directory. (You'll need some mount option to do this. Just follow the instructions.) Type `mount` to make sure that the filesystem is actually mounted.
4. Change into the `testmnt` directory, and `untar` the file `t10test.tgz` in the `/root` directory into the `testmnt` directory.
5. Delete the file `lde/changelog` in the `testmnt` directory.
6. `cd` out of the `testmnt` directory and unmount it.

2. Starting and using lde

`Lde` is designed as a tool used to recover deleted files. It is not the most bug-free program in the world, but it is still better than having to dump out all the bits of the filesystem by writing your own program. You can download it from the address

<http://www.ibiblio.org/pub/Linux/system/filesystems/!INDEX.html>

To speed things up, it has already been installed in our lab. To start `lde`, make sure the device or file is not currently mounted, and type `lde` followed by the name of the device or file holding the filesystem. In case you start the X windows, note that `lde` expects a window of 25 rows.

After a beginning message, the program greet you with the **superblock view**, which display information contained within the superblock. Now **try to use the filesystem size to derive all the information that you see**. Note that a "zone" means a data block.

Other than the superblock view, `lde` has three other primary views: the inode view, the block view and the recovery view. You can switch between these four views by typing `s`, `i`, `b` and `r` respectively, and can exit the program from a primary view by typing `q`. **The four primary views are completely independent** in the location that is displayed. The program maintains a current inode number and a current block number. They are shown in the title bar.

The block view: in the block view, you can see a **hex-dump of a portion of the disk**. This way you can see arbitrary information within the filesystem. Within this view, you can move the cursor around by using the cursor keys and page-up and page-down, and when the cursor is move across a block boundary, the current block number is updated. If you want to jump to a

particular block, you can type #, which allow you to type in an arbitrary block number. Finally, you can interpret the current block as a directory and show its content by typing d. A directory popup is then shown (see below).

The inode view: in the inode view, you can see all the information stored within the current inode. Remember that an inode stores all information about a file, directory, device, pipe, socket or symbolic link, except the filename which is stored as data of the directory. Arrow keys will bring you around the fields of the view. If you want to change the current inode, you can use page-up and page-down, or you can type # which allow you to directly enter an inode number to go to. Finally, if the cursor is currently in one of the direct blocks or indirect blocks, you can type the capital B to switch to the block, or type d to show the directory popup for that block.

The recovery view: this is used for recovering files. It stores a “fake inode”, to contain direct blocks and indirect blocks numbers. In the recovery view, you can dump all these blocks to a file by typing r. In the inode or block view, if you find a block that you want to save, you can type a key corresponding to the characters displayed in the right side of the title bar. This copies the block number to the fake inode. If you are in the inode view, you can copy a whole inode to the fake inode by typing the capital R.

The directory popup: in the directory popup, you can find information that are stored within a directory block in an easy-to-read format. You can navigate the directories by using up and down arrows to select a directory and typing enter to switch to it, or you can type q to get back to the previous primary view.

3. Your tasks

After familiarizing yourselves with lde, do the followings:

1. Show the group descriptors of group 0 (i.e., first group) in the block view. By using the tutorial notes, read out the group descriptor information.
2. Show the block bitmap of group 0 in the block view. Determine which data blocks are used. (Note: there is an error in the reading material. Used blocks are represented by bit 1, and free blocks are represented by bit 0.)
3. Show the inode bitmap of group 0 in the block view. Determine which inodes are used.
4. Find the inode table of group 0 in the block view. Where is the end of that inode table?
5. Now go to the inode view, and visit inode 2 (the root directory). (Inode 1–10 are reserved, many for things that are not yet implemented. Look at `/usr/include/linux/ext2_fs.h` if you want to know what they are reserved for.) Compare the block view of the data block and the directory popup.
6. Find the inode for the the file `lde-2.5/UNERASE`. What blocks are used for its content? Using the block view, show the block content of the first few blocks and the indirect block.
7. Find the inode for the symbolic link `lde`. Where is the link target stored?
8. There is a hard link in the `lde-2.5` directory. Can you find it? How it differs from the symbolic link?
9. Find the inode for the deleted file. Try to recover it into a new file of the `/root` directory.