Middle East Technical University   Department of Computer Engineering

CENG 334

Introduction to Operating Systems

Spring 2016-2017

HW3

Due date:10th June 2017, 23:55

1 Objectives

This assignment aims to get you familiar with file system structures by an implementation of a defragmentation tool for ext2 file system. Defragmentation is reordering of disk blocks to reduce external fragmentation. Fragmentation can happen with addition and removal of new files or expansion and contraction of existing ones. Defragmentation improves read times especially in spinning disk drives by putting files in continuous blocks.

2 EXT2

Following diagram shows the structure of a ext2 system on disk. The first 1024 bytes of the disk is always reserved as a boot block. After the boot block, a number of block groups exist. Each block group starts with its super block, then a number of group descriptors are placed. A bitmap of existing blocks and a bitmap of existing inodes occupies one block of disk each. The size of the inode table that comes after the inode bitmap depends on the number of inodes created while formatting. The rest of the block group consists of data blocks.

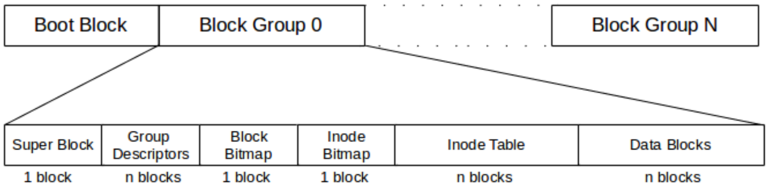

Figure 1: EXT2 structure

For more information on EXT2 file system, you should look at OSDev wiki page on EXT2. Also this page by Dave Poirer contains more details.

But important details are:

- Inode and block numbering starts at 1.
- Block numbering starts at the beginning of the disk. Super block of the first group resides in block 1.

- The root inode always resides in inode number 2.
- The first 11 inodes are reserved.
- There is always a lost+find directory in root.
- Disk sectors are 512 bytes.

2.1 Details

You can create a disk image with 128 blocks of size 1024 with the following command:

```
$ dd if=/dev/zero of=image.img bs=1024 count=128
```

The created disk image can be formatted with `mke2fs`. Folloing commands formats the disk and forces the creation of 32 inodes.

```
$ mke2fs -N 32 image.img
```

You can mount the image by creating a loopback device by:

```
$ mkdir mnt
$ sudo mount -o loop image.img mnt
```

and unmount with:

```
$ umount mnt
```

On lab computers or any other computer where you do not have administrative privileges, you can use FUSE based `fuseext2` to mount your image to a folder you own.

```
$ fuseext2 -o rw+ image.img mnt
```

and unmount with:

```
$ fusermount -u mnt
```

In this homework, you are **not** expected to handle more then one block group. But different block sizes and different number of inodes and different number of blocks must be handled.

You can inspect differences between two image files using a `xxd` hex dump and `diff`. Following command is an example for this purpose.

```
$ diff <(xxd image1.img) <(xxd image2.img) > images.diff
```

2.2 Implementation, Compilation & Execution Details

You should look at `hw3.c` file for some ideas for your implementation. Also given header file `ext2.h` has helpful comments.

Any standard library can be used in your code. You **have to** provide a makefile with your implementation which creates an executable named `defragext2`.

Your code will be executed with `./defragext2 imagefile`. You can assume a valid, uncorrupted ext2 image file. Your program have to give the following output on standard output while defragmenting the image file.

1. A list of inodes and its is associated blocks before defragmentation.
2. A list of inodes and its is associated blocks after defragmentation.

Single, double and triple indirects have to be marked appropriately, ie. `s1`, `d1`, `t1` . An example output:

```
$ ./defragext2 imagefile
12 14
12 16
12 17
12 19
13 12
###
12 7
12 8
12 9
12 10
13 11
```

Example shows blocks for inodes 12 and 13 before and after defragmentation.

3 Regulations

1. Your code have to be in C/C++.

2. Submission will be done via COW. Create a tar.gz file named *hw3.tar.gz* that contains all your source code files and a makefile. The executable should be named *defragext2*.

3. Following sequence of commands should compile and run you code, otherwise you lose 10 points.

   ```
   $ tar -xf hw3.tar.gz
   $ make all
   $ ./defragext2
   ```

4. Your codes will be evaluated with a black-box approach and have to compile and run on lab machines.

5. Please ask your questions related to the homework on piazza instead of email. Your friends, who may face with same problem, can see your questions and answers.

6. Do not cheat.