The LNM Institute of Information Technology

Title: To manage and store large files in file systems on Linux Platform.

Course Name: Operating Systems

Instructor Name: Gaurav Somani

Team Members:

Kamal Kishor Sharma – y10uc139

Rachit Mathur – y10uc245

Pankaj Gupta – y10uc208

ABSTRACT

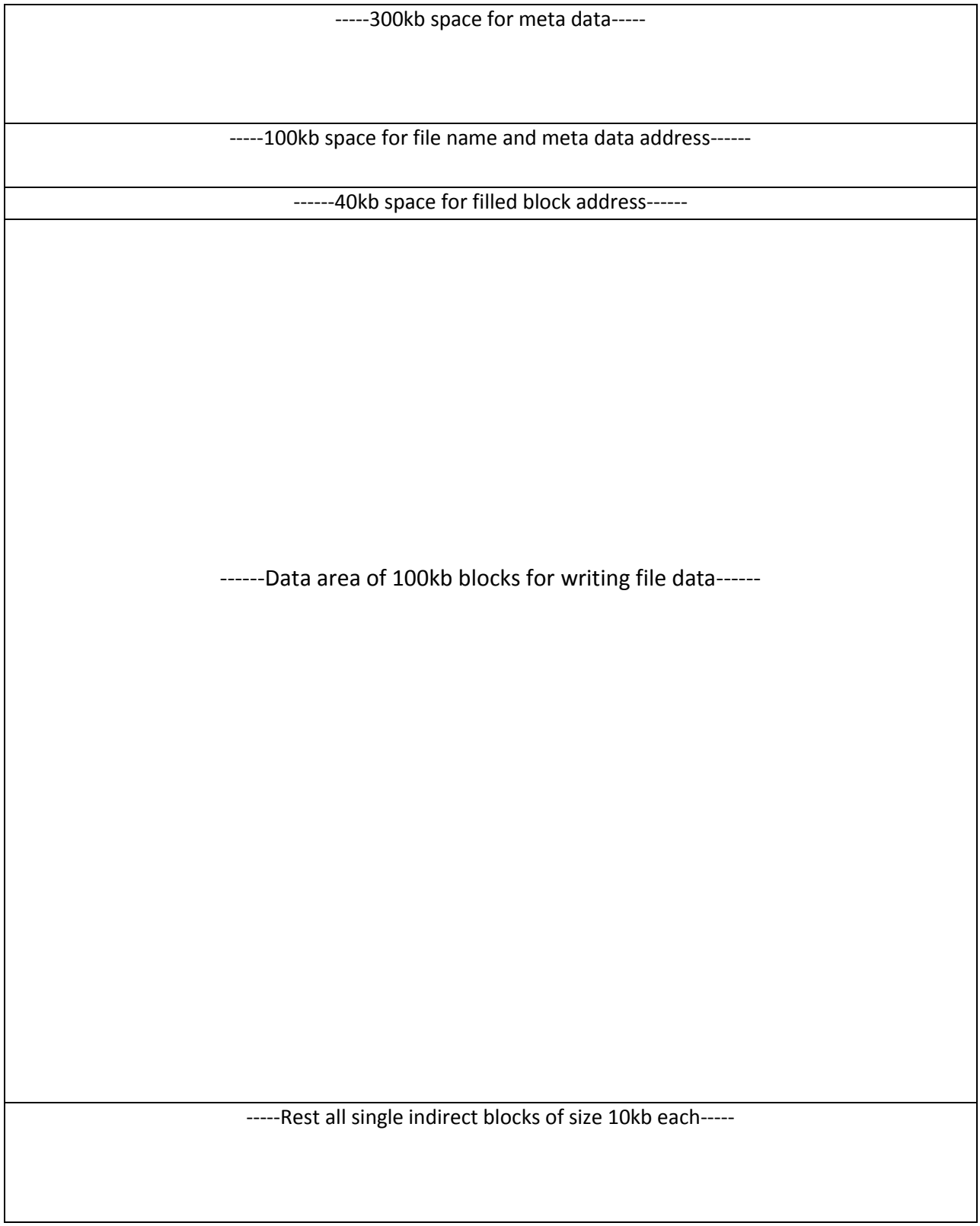
The major problem of the current file systems is overhead due to cluster switching which involves retrieving and updating of cluster connection information saved on the file allocation table. Because of this overhead, response times of file read and write operation cannot be guaranteed in the file systems. In this report, to solve this problem, we present design and implementation of a file system to store and manage large files.

Introduction

In the modern world of high speed internet and networking, data transfer and storage from one server to another with a fast and efficient way is the major concern of all IT developers.

Below table shows how much overhead is present in each file system.

File system	Allocated kilobytes (out of 1G)	Overhead %
ext2	16948	1.6%
ext3 [2]	33352	3.2%
ext4 [2]	33288	3.2%
Xfs	5132	0.5%
ntfs [3]	5748	0.5%
msdos & vfat	2076	0.2%
reiserfs [3]	32916	3.1%
btrfs [3]	4224	0.4%
nilfs2 [3]	2060	0.2%
jfs [3]	4364	0.4%
gfs [3]	16612	1.6%
gfs2 [3,4]	132576	12%



Flowchart: Showing how space is allocated in file.

Critical View:

At the initial stage of project we read 2-3 chapter of a book to get an overview of what a file system is and how it work. Then we design a code which can do create, read, write, rename and delete function for both directory and files. But after discussion with instructor we came to know that, this code will not run with mkfs system call on empty large file or disk partition because we are using the system calls that are already define on Linux platform. Then we really came to know what we have to do. When we store data in small files, we have to manage a large metadata which lead to a wastage of disk space. To deal with this situation we have design metadata in such a way that it will only store important information like file name, size, permission, mode and single indirect block address. Therefore by the use of basic headers files and C language, we made a code which has function to create, read, write and stat (for file information) function.

Conclusion:

A file system is created, which can store files of large size up to 12MB using single indirect block allocation method on a large file disk space.

Reference:

http://rwmj.wordpress.com/2009/11/08/filesystem-metadata-overhead/

http://linux.die.net/man/2/open
http://pubs.opengroup.org/onlinepubs/007904975/functions/write.html

www.nobius.org/~dbg/practical-file-system-design.pdf

http://wikipedia.com