

OS Tutorial : Assignment 3

Huan Wang, Changli Zhang
huanwang, clzhang@uvic.ca

Part 1 : diskinfo

- * Displays basic information about the file system. Mainly rely on the information from boot sector.
 - * OS Name: starting byte:3, length:8 bytes (boot sector)
 - * Label of the disk: boot sector & root directory
 - * Total size of the disk: total sector count * 512
 - * Free size of the disk: rely on FAT entries: if the value fat entry == 0x00, then this is a empty sector.

Part 1 : diskinfo

- * The number of files in the disk:
 - * starting from root directory, traverse the directory tree.
 - * # of files = # of directory entries in each directory
 - * However, skip a directory entry if:
 - (1) the value of it's attribute field is 0x0F or
 - (2) the 4-th bit of it's attribute field is 1 or
 - (3) the Volume Label bit of its attribute field is set as 1 or
 - (4) the directory entry is free (see documentation for details)
- * Number of FAT copies: Boot sector
- * Sectors per FAT: Boot sector

Part 2 : disklist

- * Display all the contents of all directories in the file system
 - * First column: F/D: directory entries (attribute field) in each directory.
 - * 10 characters to show the file size: directory entries (last field)
 - * File name: directory entries (first field)
 - * Creation date & creation time: directory entry.
 - * Date: Year: high 7 bits + 1980, Month: middle 4 bits, Day: low 5 bits
 - * Time: Hours: high 5 bits, minutes: middle 6 bits

Part 3 : diskget

- * Copies a file from the file system (root directory) to the current directory in Linux:

`./diskget disk.IMA ANS1.PDF`

- (1) Convert the given filename to upper case, then search this filename from directory entries in root folder.
- (2) If filename matching, get the first cluster number & file size.
- (3) Rely on FAT entries to copy.
- (4) If (Last cluster of file && reach the filesize value), stop copy.

Part 4 : diskput

- * Copies a file from the current Linux directory into the root directory of the file system:

`./diskput disk.IMA foo.txt`

- (1) Create a new directory entry in root folder
- (2) Check if the disk has enough space to store the file
- (3) Go through the FAT entries to find unused sectors in disk and copy the file content to these sectors.
- (4) Update the **first cluster number** field of directory entry we just created, update the FAT entries we used.

Disk Read / Write

- * When the disk file is small: **mmap** - Map File into Memory

- * `#include <sys/mman.h>`
- * `void *mmap (void *addr, size_t length, int prot, int flags, int fd, off_t offset)`
- * E.g., `mmap(0, filesize, PROT_READ, MAP_SHARED, fp, 0);`
`mmap(p, filesize);`

- * When the disk file is large: File Pointer

- * `#include <stdio.h>`
- * `size_t fread (void *ptr, size_t size, size_t nmemb, FILE *stream)`
- * `size_t fwrite (const void *ptr, size_t size, size_t nmemb, FILE *stream)`
- * `int fseek (FILE *stream, long int offset, int whence)`