

CSCI315 – Operating Systems Design

Department of Computer Science
Bucknell University

Introduction to File Systems

Ch 13.1-13.2

This set of notes is based on notes from the textbook authors, as well as L. Felipe Perrone, Joshua Stough, and other instructors.

Xiannong Meng, Fall 2021.

File System Topics

- File Concept
- Access Methods
- Directory Structure
- File System Mounting
- File Sharing
- Protection
- File System Implementation
- File System Internals


File Concept

- A file is a named collection of related information recorded on secondary storage.
- **“Contiguous” logical** address space.
- A collection of bytes, of which the meaning is interpreted by the applications. Show feep.pgm.
- File types:
 - Data
 - Programs (source code and executable)
- Files are objects with *attributes* and *operations*

Types of File Structures

- None: just a sequence of words or bytes, such as files on Linux
- Simple **record** structure such as database files:
 - Lines,
 - Fixed length,
 - Variable length.
- Complex Structures:
 - Formatted document,
 - Relocatable load file.
- Can simulate last two with first method by inserting appropriate control characters.
- Who decides:
 - Operating systems,
 - Programs.

File Attributes

- **Name** – only information kept in human-readable form.
 - **Type** – needed for systems that support different types.
 - **Location** – pointer to file location on device.
 - **Size** – current file size.
 - **Protection** – controls who can do reading, writing, executing.
 - **Time, date, and user identification** – data for protection, security, and usage monitoring.
-  Information about files is kept in the **directory** structure, which is maintained on the disk by the operating system.

File Operations

- **Create** – create a new file
- **Write** – write some data into an existing file
- **Read** – read some data from an existing file
- **Seek** – move the read/write point to a specific position
- **Delete** – remove the file from the file system
- **Truncate** – remove a portion of data, keep current attributes
- **Open** – open an existing file, get it ready for operations
- **Close** – close an existing file, no further operations can be applied to a closed file

File Operation Example in C

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    FILE * fp;
    char ch;

    fp = fopen("hello.txt", "r"); /* open file 'hello.txt' for read */
    if (!fp) {
        fprintf(stderr, "error in opening file\n");
        exit(2);
    }
    fscanf(fp, "%c", &ch);
    while (!feof(fp)) { /* if not end of the file, continue */
        /* reading and printing one char at a time */
        printf("%c", ch);
        fscanf(fp, "%c", &ch);
    }
    fclose(fp);
    return 0;
}
```

```
[xmeng@linuxremote1 files]$ cat hello.txt
Hello World!
How are you?
[xmeng@linuxremote1 files]$ gcc file-basics.c
[xmeng@linuxremote1 files]$ ./a.out
Hello World!
How are you?
[xmeng@linuxremote1 files]$
```

[http://www.eg.bucknell.edu/~cs315/
F2021/meng/code/files/file-basics.c](http://www.eg.bucknell.edu/~cs315/F2021/meng/code/files/file-basics.c)

Use System Calls

```
/* only the essential part is listed */
int main() {
    int fp;
    char ch;
    int bytes_read = 0;

    fp = open("hello.txt", O_RDONLY);
    /* error check removed for presentation ... */
    bytes_read = read(fp, &ch, sizeof(ch));
    while (bytes_read > 0) {
        printf("%c", (char)ch);
        bytes_read = read(fp, &ch, sizeof(ch));
    }
}
```

```
[xmeng@linuxremote1 files]$ cat hello.txt
Hello World!
How are you?
[xmeng@linuxremote1 files]$ gcc file-syscalls.c
[xmeng@linuxremote1 files]$ ./a.out
Hello World!
How are you?
[xmeng@linuxremote1 files]$
```


File Types: Name and Extension

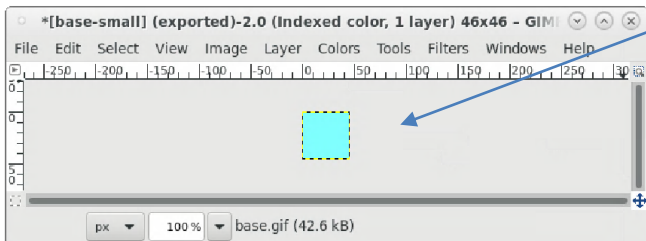
file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine- language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rtf, doc	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	ps, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes com- pressed, for archiving or storage
multimedia	mpeg, mov, rm, mp3, avi	binary file containing audio or A/V information

Linux Files

- Linux files are just a sequence of bytes, regardless of the types. Use the command **file** to find out types.
- Here are two examples of files, one is a text and the other is an image. (See next slide for details.)
 - “hello.txt” is a text file with 26 characters (bytes) in it.
 - “base-small.png” is a small random image file (a small blue square) of 125 bytes in size.

Linux Files – Sequence of Bytes

```
File Edit View Search Terminal Help
[bash xmeng@linuxremote2 34-file-intro]$ ls
base.gif base-small.png file-basics.c file-syscalls.c hello.txt
[bash xmeng@linuxremote2 34-file-intro]$ display base-small.png
```



how the image "base-small.png" looks

byte content of "base-small.png" (125 bytes)

logical addresses of the file content

byte content of "hello.txt" (26 bytes)

size of the files

```
File Edit View Search Terminal Help
[bash xmeng@linuxremote2 34-file-intro]$ xxd base-small.png
00000000: 8950 4e47 0d0a 1a0a 0000 000d 4948 4452  .PNG.....IHDR
00000010: 0000 002e 0000 002e 0103 0000 006d 77b9  .....mw.
00000020: 7400 0000 0350 4c54 4580 ffff f869 34a5  t....PLTE...i4.
00000030: 0000 0009 7048 5973 0000 07b1 0000 07b1  ....pHYs.....
00000040: 0106 c561 8600 0000 0774 494d 4507 e40a  ...a.....TIME...
00000050: 1e0f 0a0a aa11 8d90 0000 000d 4944 4154  .....IDAT
00000060: 18d3 6360 1805 9401 0001 4200 01ac 8649  ..c`.....B....I
00000070: c200 0000 0049 454e 44ae 4260 82      .....IEND.B`.
[bash xmeng@linuxremote2 34-file-intro]$ xxd hello.txt
00000000: 4865 6c6c 6f20 776f 726c 6421 0a48 6f77  Hello world!.How
00000010: 2061 7265 2079 6f75 3f0a              are you?.
[bash xmeng@linuxremote2 34-file-intro]$ ls -l hello.txt base-small.png
-rw-rw---- 1 xmeng cs 125 Oct 30 11:10 base-small.png
-rw-rw---- 1 xmeng cs 26 Oct 30 09:42 hello.txt
[bash xmeng@linuxremote2 34-file-intro]$
```

Access Methods

- **Sequential Access**

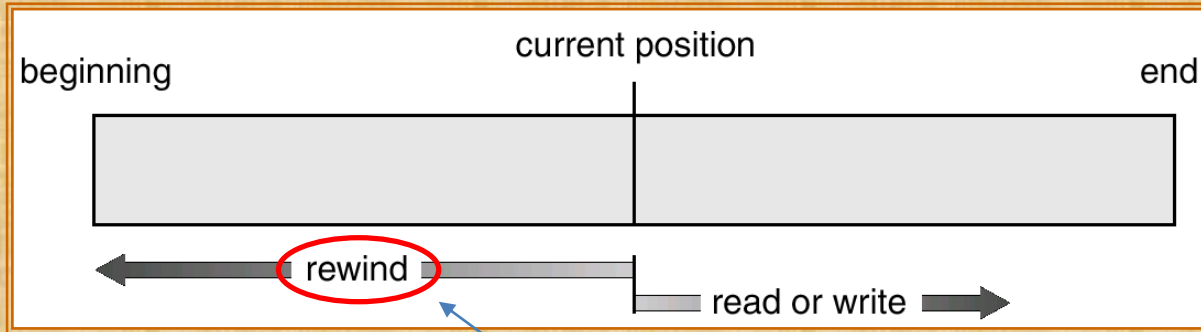
read next
write next
reset

- **Direct Access**

read n
write n
position to n
read next
write next
rewrite n

n = relative block number

Sequential-access File



Where does the term "rewind" come from?
It was from tape-based file storage medium in old days!

Try: `man 3 rewind`

Sequential Access Files

- Because most files are logically organized in sequence, accesses are in sequence.
- Linux files can be “directly accessed” through the operation of “`lseek()`” (see **man lseek**)

Linux lseek(3) Example

```
fp = open("hello.txt", O_RDONLY);
lseek(fp, 6, SEEK_SET); // move forward 6 bytes
// "Hello world!" so we should read 'w' now
bytes_read = read(fp, &ch, sizeof(ch));
printf("char read = ['%c']\n", ch);
```

Only relevant segment is shown here. For the complete program, see <http://www.eg.bucknell.edu/~cs315/F2021/meng/code/files/file-lseek.c>

```
File Edit View Search Terminal Help
[bash xmeng@linuxremote2 34-file-intro]$ cat hello.txt
Hello world!
How are you?
[bash xmeng@linuxremote2 34-file-intro]$ gcc file-lseek.c
[bash xmeng@linuxremote2 34-file-intro]$ ./a.out
char read = ['w']
[bash xmeng@linuxremote2 34-file-intro]$ █
```

Example of Index and Relative Files

