Topics: Directories, File Attributes, Bits, File Operations

Approach: Write our own versions of Unix programs

Featured Commands:
ls, ls -l

Main Ideas:
A directory is a list of file names
File attributes can be numerical values
File attributes can be yes/no settings
Bitwise operations (|, &)

Agenda

stdio	fopen/getc/putc/fclose vs open/read/write/close
ls1	opendir(), readdir(), closedir() not sorted, and does not support -l
file attributes	owner, dates, size, type, permissions look at chmod(), chown(), chgrp(), touch
stat1	stat() system call
stat2	coding yes-no attributes as bits binary, octal, bit operations
ls2	combine ls1 and stat2 still not sorted, and what about ls -l filename

Sections

0. First, one more item about buffering:
   * people asked after class about using open,read,write,close vs fopen, getc, putc, fclose -- what's the difference?
   * Answer - the standard library adds buffering
     a FILE points to a struct containing a (a) ptr to a buffer, (b) some pointers to keep track of where we are, (c) file descriptor, (d) type of open(read/write/both)

     Thus, getc or fgets do what our utmplib do.
     The include file <libio.h> line 271 has the def

1. files have content AND properties
2. Using ls
3. A brief review of the Unix directory tree
4. Is this just like who? open, read, show, close
5. What is a directory?
6. Reading a directory
7. How close are we to a complete version of ls?
8. How do we add the -l option?
9. How does stat() work?
10. What info do we need from what stat tells us?
11. How do we decode the type and file permissions?
12. How do we convert uid to logname?
13. How do we convert gid to group?
14. Building stat2.c : format info just like ls -l
15. Can we combine stat2.c with ls1.c ?