CSC 4304 - Systems Programming
Fall 2008

LECTURE - VII

FILES & DIRECTORIES - II

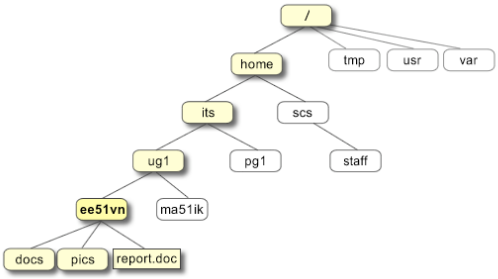
Tevfik Koşar

Louisiana State University
September 23rd, 2008

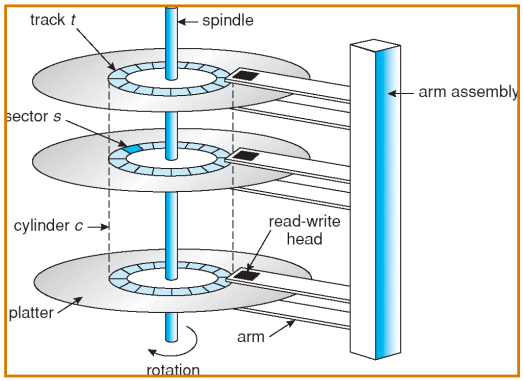
In Today's Class

- Directory Structure in UNIX
- inode manipulation
- implementing "pwd"
- set-UID-bit

Unix Directory Tree Structure



Disk Architecture

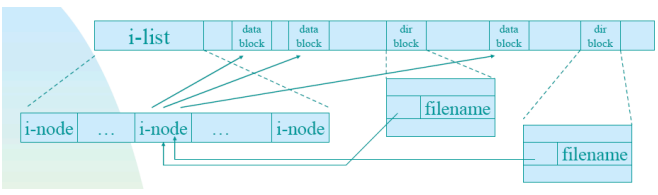


Disk Regions

block 0 block 1 block 2 block 3 ...

boot block: boot info for OS
superblock: organization of the file system
inode table: links to files and dirs
data area: actual contents of files

inode table



Directories

- Directory is a special file that contains list of names of files and their inode numbers
- to see contents of a directory:

```
$ls -1ia .
9535554 .
9535489 ..
9535574 .bash_history
9535555 bin
9535584 .emacs.d
9535560 grading
9535803 hw1
9535571 test
9535801 .viminfo
```


Example

Example inode listing

```
$ ls -iaR demodir
865 .       193 ..      277 a       520 c     491 y

demodir/a:
277 .       865 ..      402 x

demodir/c:
520 .       865 ..      651 d1        247 d2

demodir/c/d1:
651 .       520 ..      402 xlink

demodir/c/d2:
247 .       520 ..      680 xcopy
```


Directories - System View

- user view vs system view of directory tree
  - representation with “dirlists (directory files)”
- The real meaning of “A file is in a directory”
  - directory has a link to the inode of the file
- The real meaning of “A directory contains a subdirectory”
  - directory has a link to the inode of the subdirectory
- The real meaning of “A directory has a parent directory”
  - “..” entry of the directory has a link to the inode of the parent directory

Link Counts

- The kernel records the number of links to any file/ directory.
- The *link count* is stored in the inode.
- The *link count* is a member of *struct stat* returned by the *stat* system call.

Change Links

- What will be the resulting changes in directory tree?
- rename y c/d1/y.old
- cp a/x c
- ln -s a/x c/d2/x

Implementing “pwd”

1. “.” is 247
   chdir ..
2. 247 is called “d2”
   “.” is 520
   chdir ..
3. 520 is called “c”
   “.” is 865
   chdir ..
4. 865 is called “demodir”
   “.” is 193
   chdir ..

Implement “pwd” in C

```
#include      <stdio.h>
#include      <sys/types.h>
#include      <sys/stat.h>
#include      <dirent.h>

 ino_t    get_inode(char *);
 void     printpathto(ino_t);
 void     inum_to_name(ino_t, char *, int);

 int main()
 {
              printpathto(get_inode("."));
              putchar('\n');                  /* then add newline */
              return 0;
 }
```


- **void printpathto(ino_t this_inode)**

```
//  prints path leading down to an object with this inode

{
    ino_t    my_inode;
    char     its_name[BUFSIZ];

    if (get_inode("..") != this_node)
    {
         chdir(“..”);                  /* up one dir    */
         inum_to_name(this_inode, its_name, BUFSIZ);    /* get its name  */
         my_inode = get_inode(“.”);
         printpathto(my_inode);
         printf(“%s”, its_name);
     }
 }
```


- **ino_t get_inode(char *fname);**

```
   // returns inode number for the file

{
    struct stat info;

    if ( stat(fname, &info) == -1 ){
         fprinf(stderr, “Cannot stat!”);
         exit(1);
    }
    return info.st_ino;
}
```


- **void inum_to_name(ino_t inode_to_find, char *namebuf, int buflen)**

```
/*
 *  looks through current directory for a file with this inode
 *  number and copies its name into namebuf
 */
{
    DIR      *dir_ptr;                /* the directory */
    struct dirent    *direntp;        /* each entry    */

    dir_ptr = opendir( "." );
    if ( dir_ptr == NULL ){
         fprintf(stderr, "cannot open a directory\n");
         exit(1);
    }

    //search directory for a file with specified inum

    while ( ( direntp = readdir(dir_ptr) ) != NULL ){
         if (direntp->d_ino == inode_to_find)
         {
              strcpy( namebuf, direntp->d_name, buflen);
              namebuf[buflen-1] = ‘\0’
              closedir( dir_ptr );
              return;
         }
    }
    fprintf(stderr, "error looking for inum %d\n", inode_to_find);
    exit(1);
}
```


Set-User-ID Bit

- How can a regular user change his/her password?

```
cs4304_kos@classes:~> ls -l /etc/passwd
-rw-r--r--  1 root root 70567 2008-09-23 09:28 /etc/passwd
```

- Permission is given to the program, not to you!

```
cs4304_kos@classes:~> ls -l /usr/bin/passwd
-rwsr-xr-x  1 root shadow 79520 2005-09-09 15:56 /usr/bin/passwd

0400 : set user ID
0200 : set group ID
0100 : sticky bit - keep the program in swap device
```

Setting mode bits in your Code

- fd = creat ("newfile", 4755);
- chmod("/tmp/myfile", 0644);

- umask (022); -> turn off the bits --- -w- -w-

Summary

- Directory Structure in UNIX
- inode manipulation
- implementing "pwd"
- set-UID-bit

- Next Class: UNIX Process Environment
- Try "pwd"

Acknowledgments

- Advanced Programming in the Unix Environment by R. Stevens
- The C Programming Language by B. Kernighan and D. Ritchie
- Understanding Unix/Linux Programming by B. Molay
- Lecture notes from B. Molay (Harvard), T. Kuo (UT-Austin), G. Pierre (Vrije), M. Matthews (SC), and B. Knicki (WPI).