Unix and Network Programming

With Linux and C++

Nathan Warner

Northern Illinois
University

Computer Science
Northern Illinois University
February 16, 2023
United States

Contents

Commands

- **more, less, pg**: Display contents of file one page at a time
- **head**: Display beginning portion of file (Default: 10 lines)
- **tail**: Display end portion of file
- **wc**: Count file content (-l -w -c) (lines, words, characters)
- **diff**: Compare two files line by line
- **gzip, gunzip, zcat**: compress file content (.gz files)
- **sort**: Sort file contents (-r -n -t -k -f) (reverse, numeric, field delimiter, field1[,field2], ignore case)
- **quota -v**: Disk quota
- **lpr**: Send files to printer, -P to specify printer (lpcsl, lpfrl, etc)
- **lpq**: Show print queue
- **lprm**: Remove job from print queue

Permissions

Unix uses discretionary access control (DAC) model

- Each directory/file has owner
- Owner has discretion over access control details

With the exception of the super user

2.1 Changing Permissions

There are four categories regarding permissions

- User
- Group
- Other
- All

To change the permissions of a file, we use the `chmod` command

```
chmod -options mode file/directory
```

2.1.1 Changing Permissions: Symbolic mode

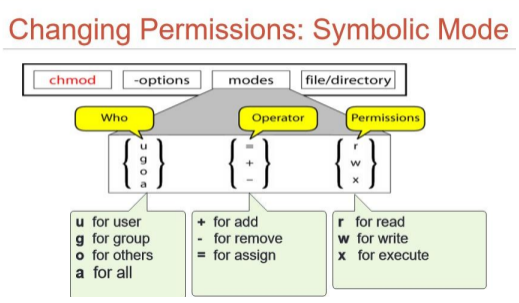

Figure 1

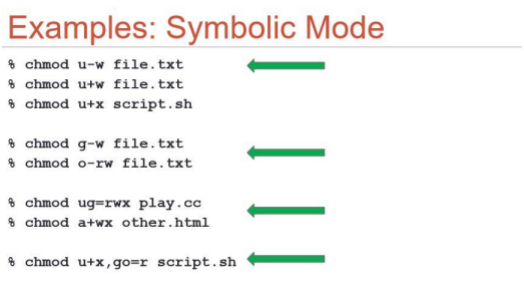

Figure 2

2.1.2 Changing permissions: Octal mode

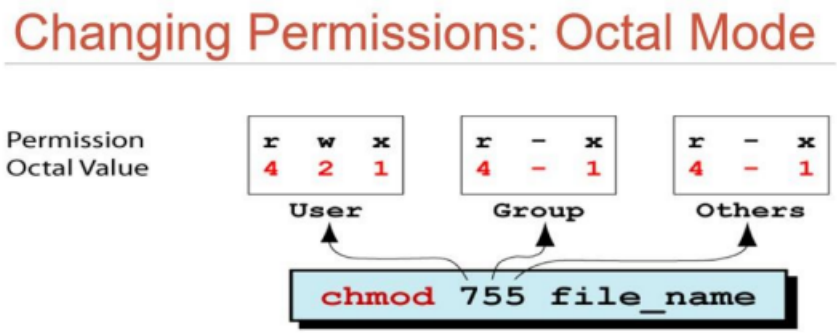

Figure 3

2.1.3 Exercise: Changing permissions

Suppose we want to change the permissions of "myfile". We want

- Read, write, and execute for user
- Read and execute for group
- Execute for other

```
chmod u=rwx, g=rx, o=x myfile
chmod 751 myfile
```

2.2 Special Permissions

3 additional permissions can be set on files and directories

- Set user ID (SUID)
- Set group ID (SGID)
- Sticky bit

2.2.1 Set user ID (SUID)

Concept 1: SUID is used for executable files, it makes executables run with permissions of file owner, rather than invoker

For example, the passwd command uses this permission. This allows user access to otherwise protected system files while changing password

2.2.2 Set group ID (SGID)

Concept 2:

- **For executables**: The logic for SGID is the same as SUID, but for group owner rather than file owner
- **For directories**: A file created in the directory will be owned by the group owner of the directory, not the group of the user who created the file

2.2.3 Sticky Bit

Concept 3:

- **For executables**: Executable is kept in memory even after it ended
- **For directories**: Files can only be deleted by the user that created it

2.2.4 Displaying special permissions

Concept 4: The `ls -l` command does not display special permission bits. However, since special permissions require execute, they mask the execute permission when displayed with `ls -l`

2.2.5 Setting special permissions (octal)

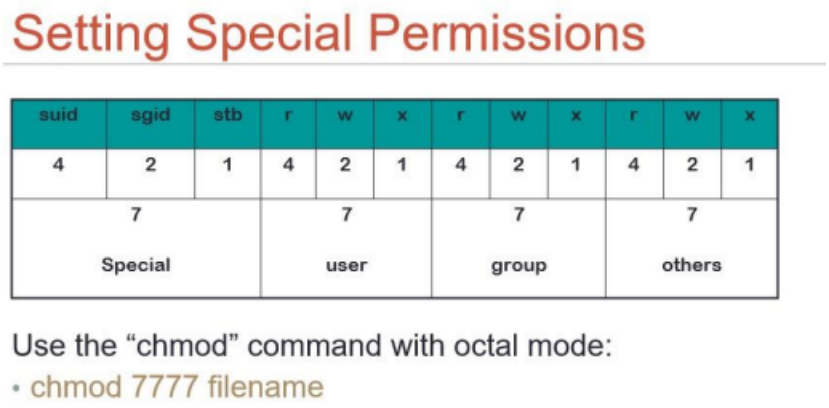

Figure 4

2.2.6 Setting special permissions (Symbolic)

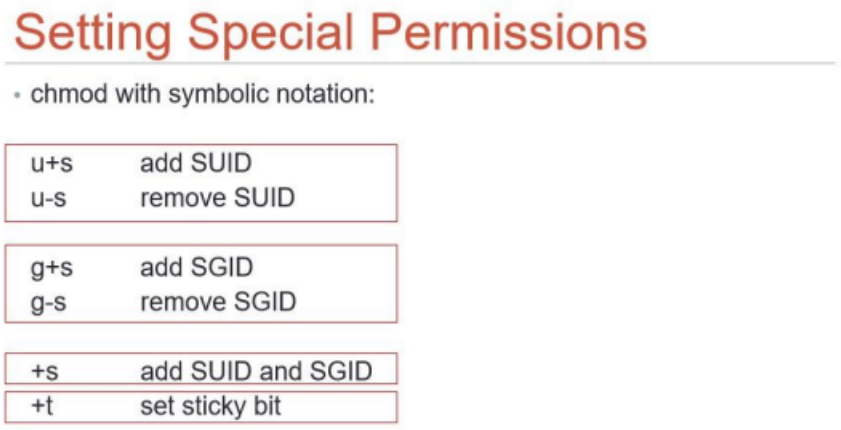

Figure 5

2.3 User mask (umask)

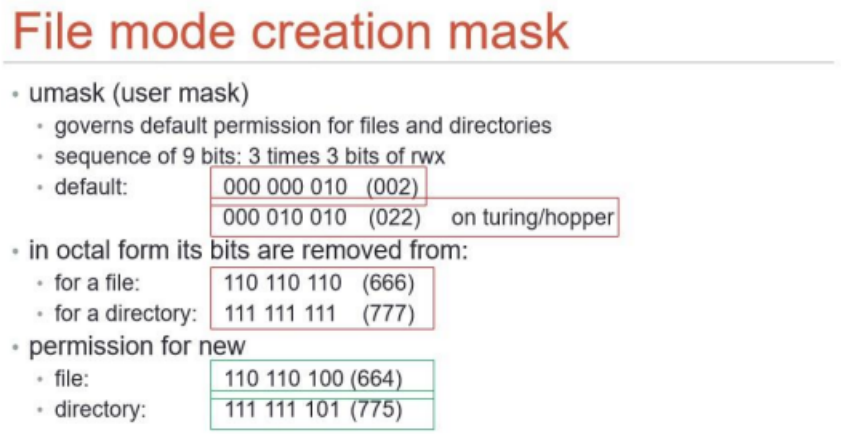

Figure 6

2.3.1 Examples

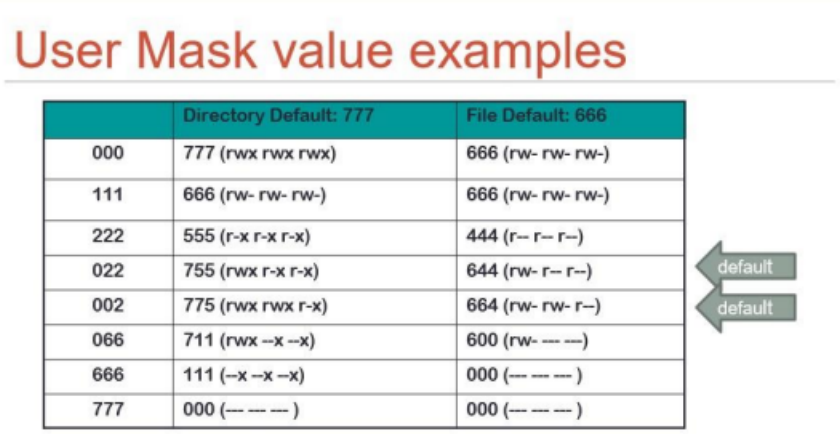

Figure 7

Network Utilitys

- Login to another computer
  - telnet, rlogin, rsh, ssh
- Copy files to another computer
  - scp
  - ftp, sftp

3.1 Login to another computer

- telnet rlogin, rsh no longer used
  - Transmit username/password without encryption
- ssh
  - Invokes shell on remote computer securely
  - **Used to:** Remote login and run command on remote computer

3.2 ssh

3.2.1 Syntax

```
ssh [user@]hostname [command]
```

This command logs in user to hostname, or if command is given, runs it on remote host

3.2.2 Common options

- **-l**: login-name
- **-X**: enable X11 forwarding

3.2.3 Examples

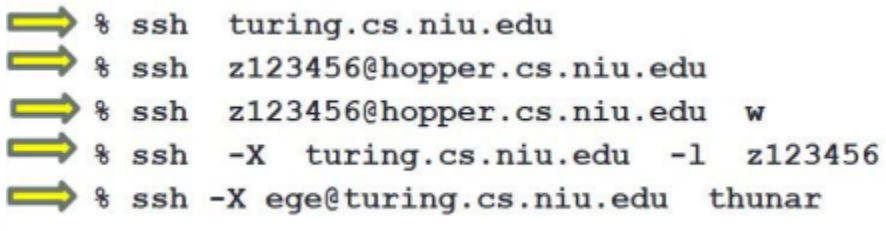

Figure 8

3.3 Copy files to another computer

3.3.1 Currently in use

- ftp

3.3.2 Secure, encrypted, part of OpenSSH

- **sftp**: Secure file transfer
- **scp**: Secure copy to remote host

3.4 ftp

3.4.1 Syntax

```
ftp hostname
```

This will prompt for userid and password

3.4.2 Anonymous ftp

- **Userid**: ftp or anonymous
- **Password**: Your email address

3.4.3 Commands

- **help**
- **ls**
- **cd**
- **put, get**
  - copy a file from local to remote host, or vice versa
- **mput, mget**
  - put/get multiple files, can use wildcards
- **bye**

3.5 sftp (Secure file transfer)

3.5.1 Syntax

```
sftp user@hostname
```

- Will prompt for password
- Same commands as ftp

3.6 scp

3.6.1 Syntax

```
scp source target
```

- source and target use extended form of pathname

  ```
  user@host:pathname
  ```

3.6.2 Common options

- **-r**: Recursively copy entire directories
- **-C**: Enables compression
- **-l**: Limit bandwidth, specified in Kbit/s

3.6.3 Examples

```
scp screenshot.png z123456@turing.cs.niu.edu:
scp z123456@hopper.cs.niu.edu:assign1.cc .
```