Unix Commands

An Advanced Introduction to Unix/C Programming

John Dempsey
COMP-232 Programming Languages
California State University, Channel Islands

Unix Commands

- Unix has over a 1,000 user and system administration commands.
- In general, each command does a specific task.
- Unix becomes more powerful when commands are grouped together.

  % **grep bash /etc/passwd | wc –l**
- To learn more about any command, command options, or file formats, you can run the **man** command.

man

- man display the manual page for Unix commands and file formats.

- Man pages are grouped by sections:
    1. Executable programs or shell commands, e.g, ls, cat, more
    2. System calls (functions provided by the kernel)
    3. Library calls (functions within program libraries)
    4. Special files (usually found in /dev)
    5. File formats and conventions, e.g. /etc/passwd
    6. Games
    7. Miscellaneous e.g. man(7), groff(7)
    8. System administration commands (usually only for root user)
    9. Kernel routines [Non standard]

man Examples

% **man ls** ← Displays man page for the ls command.

% **man passwd** ← Displays man page for the passwd command in section 1.

% **man passwd.5** ← Displays man page for the passwd file in section 5.
% **man –s5 passwd**

% **man read | cat** ← Displays man page for system call read in section 2.

Piping the output to **cat** displays the entire man page so you can scroll up and down the page.

% **man getc | cat** ← Displays man page for the getc library call in section 3.

man for Commands

For commands like ls, more, and cat, you'll see the following sections:

NAME
ls – list directory contents

SYNOPSIS
ls [OPTION …] [FILE …]

DESCRIPTION
Provides a detailed description for command.
Lists and explains the various options for the ls command, like: -l.
Describes exit status codes after running the ls command, e.g., explaining what exit codes 0, 1, and 2 mean.

AUTHOR
Written by Richard M. Stallman and David MacKenzie.

REPORTING BUGS
Explains how to report bugs.

COPYRIGHT
Copyright © 2018 Free Software Foundation, Inc.

SEE ALSO
Lists other related commands to the ls command.

man for System Calls

For system calls like read, fork, and write, you'll see the following sections:

NAME
read - read from a file descriptor

SYNOPSIS
#include <unistd.h>
ssize_t read(int fd, void *buf, size_t count);

DESCRIPTION
Provides a detailed description for the system call.

RETURN VALUE
On success, the number of bytes returned. On error, -1 is returned, and errno is set.

ERRORS
Lists various errors that can be returned by the system call.

NOTES
Provides additional details for system call.

BUGS
Provides details on any known bugs.

SEE ALSO
close(2), fcntl(2), ioctl(2), lseek(2), open(2), pread(2), readdir(2), readlink(2), readv(2), select(2), write(2), fread(3) ← **Lists other related commands to system call with section number.**

man for Library Calls

For library calls like getc, fread, and printf, you'll see the following sections:

NAME

fgetc, fgets, getc, getchar, ungetc - input of characters and strings

SYNOPSIS

```
#include <stdio.h>

int fgetc(FILE *stream);
char *ggets(char *s, int size, FILE *stream);
int getc(FILE *stream);
int getchar(void);
int ungetc(int c, FILE *stream);
```

DESCRIPTION

Provides a detailed description for each of the library calls.

RETURN VALUE

fgetc(), getc() and getchar() return the character read as an unsigned char cast to an int or EOF on end of file or error.

ATTRIBUTES

Mentions any additional attributes for the library calls.

CONFORMING TO

POSIX.1-2001, POSIX.1-2008, C89, C99.

BUGS

Mentions any known bugs.

SEE ALSO

read(2), write(2), ferror(3), fgetwc(3), fgetws(3), fopen(3), fread(3), fseek(3), getline(3), gets(3), getwchar(3), puts(3), scanf(3), ungetwc(3), unlocked_stdio(3), feature_test_macros(7)

man for Files

For Unix files like group, passwd, and shadow, you'll see the following sections:

NAME

passwd - the password file

DESCRIPTION

/etc/passwd contains one line for each user account, with seven fields delimited by colons (":").

These fields are:

- login name
- optional encrypted password

etc …

FILES

/etc/passwd
User account information.

/etc/shadow
optional encrypted password file

/etc/passwd-
Backup file for /etc/passwd.

SEE ALSO

crypt(3), getent(1), getpwnam(3), login(1), passwd(1), pwck(8), pwconv(8), pwunconv(8), shadow(5), su(1), sulogin(8).

pwd – print working directory

Displays the full path name of the current directory you are in.

% **pwd**

/home/john

ls – list directory contents

% **ls**	← Lists files and directories in current directory
% **ls –a**	← Include files/directories starting with .
% **ls -l**	← List long
% **ls -lR**	← List long and list subdirectories recursively
% **ls -ld**	← List current directory in long format
% **ls -1 ..**	← List files in one column
% **ls -F**	← List files and append one of */=>@| to entries
% **ls -l ../..**	← List files in the directory 2 levels above your current directory.

cd – change directory

% **cd** ← Change to your home directory.

% **cd /home/john** ← Change to /home/john directory.

% **cd /usr/local/bin** ← Change to /usr/local/bin directory.

% **cd ..** ← Go up one directory.

% **cd ../../dir1** ← Go up two directories and down dir1.

% **cd ~/bin** ← Change to bin in your home directory.

more – Page file content one screenful

% **more myfile.txt**

% **more /etc/passwd**

% **more ../dir1/filename.txt**

Type ‘v’ enter vi to edit the file being viewed.
Type ‘q’ to quit.
Type ‘:n’ to view next file, e.g., % **more a.c b.c c.c**

cat – Concatenate files/Print on screen

% **cat myfile.txt** ← Displays all lines in myfile.txt at once.

% **cat a.txt b.txt > c.txt** ← Concatenates contents in a.txt with b.txt to create c.txt.

% **cat –n /etc/passwd** ← Displays line numbers.

% **man ls | cat** ← Displays the entire man page for ls at once.

head – Display top few lines of a file

john@oho:~$ **head /etc/passwd** ← **Displays first 10 lines (default) in file.**

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
```

john@oho:~$ **head -5 /etc/passwd** ← **Displays first 5 lines in file.**

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
```

tail – Display last few lines of a file

```
john@oho:~$ tail /etc/services                    ← Displays last 10 lines of file.
vboxd           20012/udp
binkp           24554/tcp                       # binkp fidonet protocol
asp             27374/tcp                       # Address Search Protocol
asp             27374/udp
csync2          30865/tcp                       # cluster synchronization tool
dircproxy       57000/tcp                       # Detachable IRC Proxy
tfido           60177/tcp                       # fidonet EMSI over telnet
fido            60179/tcp                       # fidonet EMSI over TCP


john@oho:~$ tail -5 /etc/services ← Displays last 5 lines of file.
dircproxy       57000/tcp                       # Detachable IRC Proxy
tfido           60177/tcp                       # fidonet EMSI over telnet
fido            60179/tcp                       # fidonet EMSI over TCP

# Local services
```

cp – copy files and directories

% **cp a.txt b.txt** ← Copy file a.txt to b.txt.

% **cp –p a.out /pos/bin** ← -p keeps the date/time of the file.

% **cp * ../new.dir** ← Copy all files to ../new.dir directory.

% **cp –Rp dir1 dir2** ← Recursively copies files and directories in dir1 to dir2 and preserves date/time of files and directories.

mkdir – make directory

% **mkdir newdir** ← Make directory newdir.

% **mkdir /home/john/mydir** ← Make directory /home/john/mydir.

% **mkdir –p /home/john/LAB11/TASK1** ← Creates all directories in path.

rmdir – remove directory

% **rmdir mydir** ← Remove mydir directory.

% **rmdir ../temp.d** ← Remove ../temp.d directory.

% **rmdir –p a/b/c** ← Remove directories a/b/c, a/b, then a

rm – remove files or directories

% **rm myfile.txt** ← Remove myfile.txt

% **rm *** ← **Watch out with one.** Removes all files in your current directory.

% **rm –i [a-f]*** ← Interactively prompt user to remove files starting with ‘a’ through ‘f’. Use to remove filenames with special characters,

% **rm –r dir1 dir2** ← Recursively remove directories dir1 and dir2.

% **rm –f myfile.txt** ← Force removal of file myfile.txt.

% **/bin/rm –rf old.dir** ← Force remove all files and directories in old.dir. Be careful!

d rwx rwx rwx

rwx	rwx	rwx
user	group	world

d rwx rwx rwx

File Mode Bits	
Bit	Meaning
d	Directory
r	Read Access
w	Write Access
x	Execute

RWX Groupings	
First rwx	User Group Access (owner of file)
Second rwx	Group Access (if in group, have access)
Third rwx	World Access (everyone on system)

% ls –l

```
drwxr-xr-x 1 john staff  4096 Dec 24 14:15 RESERVED_WORDS
-rwxr-x--- 1 john staff 16728 Dec 25 15:51 a.out
-rw-r--r-- 1 john staff   232 Dec 24 14:17 continue.c
```

RESERVED_WORDS is a directory with rwx user, r-x group, and r-x world access.

You need r-x access to cd into a directory.

a.out is rwx user, r-x group, no world access.

continue.c is rw user, r group, r world.

d rwx rwx rwx

File Mode Bits	
Bit	Meaning
d	Directory
r	Read Access
w	Write Access
x	Execute

RWX Groupings	
First rwx	User Group Access (owner of file)
Second rwx	Group Access (if in group, have access)
Third rwx	World Access (everyone on system)

For Files:

r – You can read the file
w – You can modify the file
x – You run execute (run) file

For Directories:

r – You can see the file name in the directory.
w – You can add, remove, and rename the file in the directory.
x – You can use the directory name in a file path and change into directory.

chmod – change file mode bits

% **chmod 644 myfile.txt** ← Sets myfile.txt to rw-r--r— 6 = 110

% **chmod 755 a.out** ← Sets a.out to rwxr-xr-x 7 = 111

% **chmod 775 a.out** ← Sets a.out to rwxrwxr-x 5 = 101
4 = 100

% **chmod 400 id_rsa** ← Sets file id_rsa to be readonly, r--------

chmod – change file mode bits

Can also use …

chmod ugo +-= rwx filename/directory

where

ugo specifies user, group, other
+-= specifies add (+), subtract (-), set (=)
rwx specifies read, write, execute

% **chmod g+w myfile** ← Adds group write access to file myfile.

% **chmod o-w myfile** ← Remove other (world) write access from myfile.

% **chmod g=rwx myfile** ← Sets group to rwx for file myfile.

diff – compare files line by line

% **diff calc.c calc.c.ORIG** ← Compare calc.c with calc.c.ORIG files.

% **diff –y calc.c calc.c.ORIG** ← Compares files side by side.

% **diff dir1 dir2** ← Compare all files in directory dir1 and dir2.

Diff output will use:

< Less than indicates difference found in first file, e.g., calc.c

> Greater than indicates difference found in second file, e.g., calc.c.ORIG

wc – print newline, word, byte counts for file

% **wc /etc/passwd**
  31   43 1635 /etc/passwd

% **wc -l /etc/passwd** ← Number of lines in file /etc/passwd
31 /etc/passwd

% **wc -w /etc/passwd** ← Number of words
43 /etc/passwd

% **wc -c /etc/passwd** ← Number of characters
1635 /etc/passwd

grep – print lines that match patterns

% **grep john /etc/passwd**

john:x:1000:1000:,,,:/home/john:/bin/bash

% **grep –v john /etc/passwd** ← Displays all lines that **do not** contain “john”.

% **grep "Network Management" /etc/passwd**

systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin

grep example

```
root@comp232:~# ls
COMP232   LAB11   alberto   andrew    brittney   connor   joseph   madeline      omar        roby
EXTRA     LAB5    alex      anne.txt  caleb      george   julian   matthew       owen        snap
JOHN      LAB7    anders    antonio   castro     jasmine  kamaran  miguel        quote.txt
LAB10     LAB9    andre     betsy     chris      jose     lilang   noaccess.txt  rafael

root@comp232:~# ls -1 | grep j
jasmine
jose
joseph
julian

root@comp232:/home# ls -1 | grep j | grep -v h
jasmine
jose
julian
```

Listing the names in one column, grepping for name with the letter j, but then excluding names with the letter h.