Mangesh Raut (mbr63)

HW-4-Assignment-3-C-UNIX

Part 1-a Using the man command

1. The man (short for manual page) tool uses the less program to display information about a particular command. The usage is very simple; you just type man, followed by the command you want to find out more about. For example, to get more information about the ls command,

you can type man ls -h or man ls --HELP.

ls - list directory contents

ls [OPTION]... [FILE]...

--help display this help and exit

h : print this help message

1. Using the ls Command

The ls commands display a list of files on a specific directory. The following ls options are useful for obtaining information about file sizes on the system.

-l - displays a list of files and directories in long format and shows the sizes in bytes.

-h - scales file sizes and directory sizes into KB, MB, GB, or TB when the file or directory size is larger than 1024 bytes.

-s - displays a list of the files and directories and shows the sizes in blocks.

cd Downloads/

ls -sh

1. date

(base) mangeshraut@n3-89-82 Downloads % date

Tue Apr 26 20:28:10 EDT 2022

1. w command is used to show logged-in usernames and what they are doing. The information will be read from /var/run/utmp file.

Following options can be used for the w command:

* -h Ignore the header information
* -u Display the load average (uptime output)
* -s Remove the JCPU, PCPU, and login time.

$ who

who command is used to get the list of the usernames who are currently logged in. Output of the who command contains the following columns: user name, tty number, date and time, machine address.

$ users

users command is used to print the user name who are all currently logged in the current host. It is one of the command don’t have any option other than help and version. If the user using, ‘n’ number of terminals, the user name will shown in ‘n’ number of time in the output.

$ last

last command will give login history for a specific username. If we don’t give any argument for this command, it will list login history for all users. By default this information will read from /var/log/wtmp file.

Part 1-b Dealing with files

1. When I enter cd it shows the home directory. To enter in cs503 I use cd cs503 command.
2. ls -lh
3. ls
4. The files are created are baseball, bat and chili other two commands occur errors

(base) mangeshraut@n3-89-82 CS 503 % touch baseball bat

(base) mangeshraut@n3-89-82 CS 503 % touch -bigfile-

touch: illegal option -- b

usage: touch [-A [-][[hh]mm]SS] [-achm] [-r file] [-t [[CC]YY]MMDDhhmm[.SS]]

[-d YYYY-MM-DDThh:mm:SS[.frac][tz]] file ...

(base) mangeshraut@n3-89-82 CS 503 % touch chili&beans

[8] 7170

[8] done touch chili

zsh: command not found: beans

(base) mangeshraut@n3-89-82 CS 503 %

1. The cat command is all well and good for small files. But, if the file is large, the contents will zoom past and we'll only see the last screen worth of content. One way to overcome this is by using the more command. The more command displays the contents of the file one screen at a time for large files.
2. (base) mangeshraut@n3-89-82 CS 503 % grep -o "tcsh" big|wc -l

4

I create file Big and write the string tcsh 4 times.

1. Quota is set on your Home Directory. To view the current status on Home directory

(base) mangeshraut@n3-89-82 ~ % df -H ~/



I am not even close to half of the memory also.

1. The du command stands for disk usage. This command is included by default in most Linux distributions.

You can display the size of your current directory by typing du in the command line:

du

The system should display a list of the contents of your home directory, with a number to the left. That number is the size of the object in kilobytes.

You can add the -h option to make the output more readable:

du -h

(base) mangeshraut@n3-89-82 ~ % du -h Desktop/Spring\ 2022/CS\ 503/

56K Desktop/Spring 2022/CS 503//Week8

72K Desktop/Spring 2022/CS 503//Week6

108K Desktop/Spring 2022/CS 503//Week7

68K Desktop/Spring 2022/CS 503//Week9

1.9M Desktop/Spring 2022/CS 503//Week2/mbr63\_HW2\_Project\_1

4.3M Desktop/Spring 2022/CS 503//Week2

32K Desktop/Spring 2022/CS 503//Week5

8.0K Desktop/Spring 2022/CS 503//Week4

12K Desktop/Spring 2022/CS 503//Week3/mbr63-HW3-Project2

148K Desktop/Spring 2022/CS 503//Week3

46M Desktop/Spring 2022/CS 503/

Part 1-c The file system

1. You can return to your home directory from anywhere in the directory structure by typing cd on its own.

cd

1. The pwd command displays the full, absolute path of the current, or working, directory. pwd
2. mkdir command can create new directory whenever you want in your system. [mkdir](http://nbcs.rutgers.edu/~edseries/UNIXcmds.html#mkdir) Letters Programs Misc
3. ls /bin/\*sh
4. [ls](http://nbcs.rutgers.edu/~edseries/UNIXcmds.html#ls) -d [A-Z]\*
5. [cp](http://nbcs.rutgers.edu/~edseries/UNIXcmds.html#cp) \*let\* Letters
6. [cp](http://nbcs.rutgers.edu/~edseries/UNIXcmds.html#cp) \*.[ch] Programs
7. [cp](http://nbcs.rutgers.edu/~edseries/UNIXcmds.html#cp) \*notes\* \*misc\* Misc

|  |  |  |
| --- | --- | --- |
| File | Permissions | Symbolic Mode Command |
| pp1 | rwxrwxrwx | chmod a+rwx pp1 |
| pp2 | rwxrwxr-x | chmod ugo=rwx,o-w pp2 |
| pp3 | rwxr-xr-x | chmod u+rwx,go=rx pp3 |
| pp4 | r-x------ | chmod u=rx pp4 |
| pp5 | r--r----- | chmod ug=r pp5 |
| pp6 | rw-r--r-- | chmod u=rw,go=r pp6 |
| pp7 | r--r--r-- | chmod ugo=r pp7 |
| pp8 | rw-rw-rw- | chmod ugo=rw pp8 |
| pp9 | rwx------ | chmod u+rwx pp9 |

Part 1-d Redirection

1. (base) mangeshraut@n3-89-82 CS 503 % ls > output.1

(base) mangeshraut@n3-89-82 CS 503 % more output.1

CS422-Computer-Architecture-ComputerOrganizationAndDesign5thEdition2014.pdf

HP\_AppA.pdf

Randal E. Bryant, David R. O’Hallaron - Computer Systems. A Programmer’s Perspective [3rd ed.] (2016, Pearson) (1).pdf

The Linux Programming Interface.pdf

Week2

Week3

Week4

Week5

Week6

Week7

Week8

Week9

output.1

output.2

output.3

(END)

1. (base) mangeshraut@n3-89-82 CS 503 % ls > output.2

(base) mangeshraut@n3-89-82 CS 503 % who > output.2

(base) mangeshraut@n3-89-82 CS 503 % ps > output.2

(base) mangeshraut@n3-89-82 CS 503 % more output.2

PID TTY TIME CMD

15388 ttys000 0:00.09 -zsh

15480 ttys000 0:00.01 more output.2

output.2 (END)

first I run the ls command then who and last ps so the ps which is last command update the output.2 file with the cs command which enables you to check the status of active processes on a system that data update in output.2 file.

1. (base) mangeshraut@n3-89-82 CS 503 % ls > output.3

(base) mangeshraut@n3-89-82 CS 503 % who >> output.3

(base) mangeshraut@n3-89-82 CS 503 % ps >> output.3

(base) mangeshraut@n3-89-82 CS 503 % more output.3

CS422-Computer-Architecture-ComputerOrganizationAndDesign5thEdition2014.pdf

HP\_AppA.pdf

Randal E. Bryant, David R. O’Hallaron - Computer Systems. A Programmer’s Perspective [3rd ed.] (2016, Pearson) (1).pdf

The Linux Programming Interface.pdf

Week2

Week3

Week4

Week5

Week6

Week7

Week8

Week9

output.1

output.2

output.3

mangeshraut console Apr 24 23:57

mangeshraut ttys000 Apr 27 22:17

PID TTY TIME CMD

15388 ttys000 0:00.14 -zsh

15480 ttys000 0:00.01 more output.2

15504 ttys000 0:00.01 more output.2

15515 ttys000 0:00.01 more output.1

1. (base) mangeshraut@n3-89-82 CS 503 % ps

PID TTY TIME CMD

15388 ttys000 0:00.16 -zsh

15480 ttys000 0:00.01 more output.2

15504 ttys000 0:00.01 more output.2

15515 ttys000 0:00.01 more output.1

15588 ttys000 0:00.01 more output.3

(base) mangeshraut@n3-89-82 CS 503 % ps -fe

UID PID PPID C STIME TTY TIME CMD

0 1 0 0 Sun11PM ?? 6:51.85 /sbin/launchd

0 301 1 0 Sun11PM ?? 3:37.05 /usr/libexec/logd

ps -fe shows full listing of every processes.

ps command is used to list the currently running processes and their PIDs along with some other information depends on different options.

1. (base) mangeshraut@n3-89-82 CS 503 % more output.1

CS422-Computer-Architecture-ComputerOrganizationAndDesign5thEdition2014.pdf

HP\_AppA.pdf

Randal E. Bryant, David R. O’Hallaron - Computer Systems. A Programmer’s Perspective [3rd ed.] (2016, Pearson) (1).pdf

The Linux Programming Interface.pdf

Week2

Week3

Week4

Week5

Week6

Week7

Week8

Week9

output.1

output.2

output.3

1. (base) mangeshraut@n3-89-82 CS 503 % ps -ef | grep root

501 15859 15388 0 10:58PM ttys000 0:00.01 grep root

501 15879 15388 0 11:00PM ttys000 0:00.00 grep root

(base) mangeshraut@n3-89-82 CS 503 % grep bash /etc/passwd

\_mbsetupuser:\*:248:248:Setup User:/var/setup:/bin/bash

(base) mangeshraut@n3-89-82 CS 503 % grep root string.txt

string root

(base) mangeshraut@n3-89-82 CS 503 % ps > string.txt

(base) mangeshraut@n3-89-82 CS 503 % grep 'root' string.txt

6557 ttys000 0:00.00 grep root

(base) mangeshraut@n3-89-82 CS 503 % cat string.txt | more

PID TTY TIME CMD

6537 ttys000 0:00.32 -zsh

6557 ttys000 0:00.00 grep root

6652 ttys000 0:00.00 grep string

6709 ttys000 0:00.01 grep foo

(END)

Redirecting Standard Output

When the notation > filename is added to the end of a command, the output of the command is written to the specified file name. The > symbol is known as the output redirection operator.

Any command that outputs its results to the screen can have its output sent to a file.

Redirecting Standard Input

When the notation < filename is added to the end of a command, the input of the command is read from the specified file name. The < symbol is known as the input redirection operator.

Note: Only commands that normally take their input from the keyboard can have their input redirected.

For example, to send the file letter1 as a message to user denise with the mail command, enter:

mail denise < letter1

1. $ sort inputfile.txt > filename.txt

$ sort -o filename.txt inputfile.txt

$ sort -r inputfile.txt

$ sort -n filename.txt

$ sort -nr filename.txt

$ sort -k filename.txt

$ sort -c filename.txt

$ sort -u filename.txt

$ sort -M filename.txt

$ sort output

1. Some of the processes are run by root which shows the active processes not all above processes are run by root like the ls > output.1 then 2 and 3rd. from 4th all the processes run by root.
2. (base) mangeshraut@n3-89-82 CS 503 % ps -ef | grep root

501 15859 15388 0 10:58PM ttys000 0:00.01 grep root

501 15983 15388 0 11:07PM ttys000 0:00.00 grep root

501 17066 15388 0 12:14AM ttys000 0:00.00 grep root

The output of the ps command can be filtered with grep . For example, to show only the process belonging to the root user.

Part 2 – GDB

1.

a.

Graphical user interface, text, application, email

Description automatically generated

2.

a. (base) mangeshraut@n3-89-82 Week3 % gcc -o etox etox.c

(base) mangeshraut@n3-89-82 Week3 % ./etox

This program calculates e^x

using sum of (x^k)/k!

Enter x, n : 2

5

x,n = 2.0000 5

e^x = inf

b. mbr63@tux2:~/cs503$ g++ -g etox.c -o etox

**etox.c:25:18:** **error:** ‘**double getvalue**’ redeclared as different kind of entity

25 | double getvalue (**x**,n)

| **^**

**etox.c:3:8:** **note:** previous declaration ‘**double getvalue(double, int)**’

3 | double **getvalue** (double, int);

| **^~~~~~~~**

**etox.c:25:18:** **error:** ‘**x**’ was not declared in this scope

25 | double getvalue (**x**,n)

| **^**

**etox.c:25:20:** **error:** ‘**n**’ was not declared in this scope

25 | double getvalue (x,**n**)

| **^**

**etox.c:28:2:** **error:** expected unqualified-id before ‘**{**’ token

28 | **{**

| **^**

**etox.c:40:22:** **error:** ‘**int factorial**’ redeclared as different kind of entity

40 | int factorial (number**)**

| **^**

**etox.c:4:5:** **note:** previous declaration ‘**int factorial(int)**’

4 | int **factorial** (int);

| **^~~~~~~~~**

**etox.c:40:16:** **error:** ‘**number**’ was not declared in this scope

40 | int factorial (**number**)

| **^~~~~~**

**etox.c:42:2:** **error:** expected unqualified-id before ‘**{**’ token

42 | **{**

| **^**

mbr63@tux2:~/cs503$ nano etox.c

Use "fg" to return to nano.

[5]+ Stopped nano etox.c

mbr63@tux2:~/cs503$ g++ -g etox.c -o etox

**etox.c:** In function ‘**int main()**’:

**etox.c:16:12:** **error:** ‘**getvalue**’ was not declared in this scope

16 | series = **getvalue**(x,n);

| **^~~~~~~~**

**etox.c:** At global scope:

**etox.c:22:18:** **error:** ‘**x**’ was not declared in this scope

22 | double getvalue (**x**,n)

| **^**

**etox.c:22:20:** **error:** ‘**n**’ was not declared in this scope

22 | double getvalue (x,**n**)

| **^**

**etox.c:22:21:** **error:** expression list treated as compound expression in initializer [**-fpermissive**]

22 | double getvalue (x,n**)**

| **^**

**etox.c:35:16:** **error:** ‘**number**’ was not declared in this scope

35 | int factorial (**number**)

| **^~~~~~**

mbr63@tux2:~/cs503$ nano etox.c

Use "fg" to return to nano.

[6]+ Stopped nano etox.c

mbr63@tux2:~/cs503$ g++ -g etox.c -o etox

**etox.c:** In function ‘**int main()**’:

**etox.c:16:12:** **error:** ‘**getvalue**’ was not declared in this scope

16 | series = **getvalue**(x,n);

| **^~~~~~~~**

**etox.c:** In function ‘**double getvalue(double, int)**’:

**etox.c:29:21:** **error:** ‘**factorial**’ was not declared in this scope

29 | value += xpow / **factorial**(k);

| **^~~~~~~~~**

mbr63@tux2:~/cs503$ nano etox.c

Use "fg" to return to nano.

[7]+ Stopped nano etox.c

mbr63@tux2:~/cs503$ g++ -g etox.c -o etox

**etox.c:** In function ‘**int main()**’:

**etox.c:18:12:** **error:** ‘**getvalue**’ was not declared in this scope; did you mean ‘**getValue**’?

18 | series = **getvalue**(x,n);

| **^~~~~~~~**

| getValue

c. mbr63@tux2:~/cs503$ nano etox.c

Use "fg" to return to nano.

[8]+ Stopped nano etox.c

d. mbr63@tux2:~/cs503$ g++ -g etox.c -o etox

mbr63@tux2:~/cs503$ ./etox

This program calculates e^x

using sum of (x^k)/k!

Enter x, n : 2

5

x,n = 2.0000 5

e^x = inf

mbr63@tux2:~/cs503$ gdb etox

**e. GNU gdb (Ubuntu 9.2-0ubuntu1~20.04.1) 9.2**

Copyright (C) 2020 Free Software Foundation, Inc.

License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software: you are free to change and redistribute it.

There is NO WARRANTY, to the extent permitted by law.

Type "show copying" and "show warranty" for details.

This GDB was configured as "x86\_64-linux-gnu".

Type "show configuration" for configuration details.

For bug reporting instructions, please see:

<http://www.gnu.org/software/gdb/bugs/>.

Find the GDB manual and other documentation resources online at:

<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".

Type "apropos word" to search for commands related to "word"...

Reading symbols from etox...

(gdb) b 43

Breakpoint 1 at 0x1314: file etox.c, line 44.

(gdb) run

Starting program: /home2/home-m/mbr63/cs503/etox

This program calculates e^x

using sum of (x^k)/k!

Enter x, n : 2

3

x,n = 2.0000 3

Breakpoint 1, factorial (number=1) at etox.c:44

44 fact = fact \* j;

(gdb) step

42 for (j = 1; j <= number; j++)

(gdb) next

47 return(fact);

(gdb) n

48 }

(gdb) s

getvalue (x=2, n=3) at etox.c:32

32 xpow = xpow \* x;

(gdb) bt

#0 getvalue (x=2, n=3) at etox.c:32

#1 0x000055555555523b in main () at etox.c:18

(gdb) n

29 for (k = 0; k <= n; k++)

(gdb) n

31 value += xpow / factorial(k);

(gdb) print value

$1 = inf

(gdb) n

Breakpoint 1, factorial (number=2) at etox.c:44

44 fact = fact \* j;

(gdb) print fact

$2 = 0

(gdb) n

42 for (j = 1; j <= number; j++)

(gdb) print number

$3 = 2

(gdb) quit

A debugging session is active.

Inferior 1 [process 97521] will be killed.

Quit anyway? (y or n) y

mbr63@tux2:~/cs503$ nano etox.c

Use "fg" to return to nano.

[9]+ Stopped nano etox.c

mbr63@tux2:~/cs503$ g++ -g etox.c -o etox

mbr63@tux2:~/cs503$ ./etox

This program calculates e^x

using sum of (x^k)/k!

Enter x, n : 2

3

x,n = 2.0000 3

e^x = 6.3333333333