

PDS Lab
Section 16
Autumn-2018

Tutorial 1

Unix Commands

pwd

The *pwd* command displays the full pathname of the current directory.

Syntax:

```
pwd
```

mkdir

The *mkdir* command creates a single directory or multiple directories.

Syntax:

```
mkdir [options] <directory(ies)_name(s)>
```

Options:

- m mode Sets the access mode for the new directory.
- p if the parent directories don't exist, this command creates them.

Examples:

```
mkdir student
mkdir -m 444 tech
mkdir -p tech/net/faqs
```

Practice 1:

- (a) Create a new directory say, "lab1" under your home directory.
- (b) Create another directory "temp" under the same home directory
- (c) Type the command "pwd"

ls

The *ls* command lists all files/ directories in the directory that match the *name*. If name is left blank, it will list all of the files in the directory.

Syntax:

```
ls [options] [name]
```

Examples:

```
ls -la
ls
```

Practice 2:

- (a) Type the command "ls".
- (b) Type the command "ls -al"
- (c) Type the command "ls -la"

cd

The change directory (*cd*) command changes the current working directory. The *cd* command can be used to either change to a directory that is relative to the location of the current working directory or to an absolute location in the filesystem.

Syntax:

```
cd <directory>
```

Example:

```
cd test
cd /home/user
```

Practice 3:

- (a) Type the command "ls"
- (b) Type the command "ls -al"
- (c) Type the command "cd lab1"
- (d) Type the command "pwd"
- (e) Type the command "ls"
- (f) Type the command "cd temp"
- (g) Type the command "cd.."
- (h) Type the command "pwd"
- (i) Type the command "cd temp"
- (j) Type the command "pwd"

edit

This command invokes an editor to write a text file (e.g., a program in C).

Syntax:

```
edit <fileName> &
```

Example:

```
edit test.c
```

Practice 4:

- (a) Type `edit test.c &`
- (b) Write the following:
Programming and data structures lab.
Indian Institute of Technology Kharagpur
- (c) Save the file.
- (d) Type `ls -al`
- (e) Type `edit test.c &`
- (f) Add the following at the end of the file:
My name is <Your name> and Roll No. is <Your Roll
No.>
- (g) Save the file.
- (h) Type another file say "test.txt" using `edit` command and then write something in it. Save the file and quit

cat

The *cat* command reads one or more files and prints their contents to standard output (i.e., computer display unit). Files are read and output in the order they appear in the command arguments.

Syntax:

```
cat <file(s)>
```

Examples:

```
cat file1  
cat myfile file2 file3
```

Practice 5:

- (a) `cat test.c`
- (b) `cat test.txt`
- (c) `cat test.txt test.c`

cp

The *cp* command allows you to copy one file to another file or copy multiple files to a directory.

Syntax:

```
cp [options] <source> <target(s)>
```

Options:

- f Forces the move.
- i Prompt for a confirmation before overwriting any files.

Examples:

```
cp file1 file2 file3
cp file4 file5
```

Practice 6:

- (a) Type `pwd`
- (b) `cp test.c temp1.c`
- (c) `ls -al`
- (d) `cp -i test.c test.txt`
- (e) `cp test.txt test1.c`
- (f) `ls -al`
- (g) `cat temp1.c test1.c`

touch

The *touch* command creates one or more blank files in the current directory.

Syntax:

```
touch <file(s)>
```

Examples:

```
touch myFile
touch file1 file2 file3
```

Practice 7:

- (a) `ls -al`
- (b) `touch myFile`
- (c) `cat myFile`

mv

The *mv* command allows you to move and rename files and directories.

Syntax:

```
mv [options] <source> <target>
```

Options:

- f Forces the move.
- i Prompt for a confirmation before overwriting any files.

Examples:

```
mv file1 tech/net/faqs
mv -f tech /usr
```

Practice 8:

- (a) Type pwd
- (b) ls -al
- (c) cd temp
- (d) mv test.c /home/.../lab1
- (e) cd..
- (f) cd lab1
- (g) ls -al
- (h) cat myFile
- (i) edit test.c
- (j) Save as test2.c
- (k) Ls -al
- (l) cat test2.c

rm

This command will remove (destroy) one or more files.

Syntax:

```
rm [options] <file(s)>
```

Options:

- i prompt before every removal
- r remove directories and their contents recursively
- d remove empty directories

© D. Samanta, IIT
Examples:

```
rm file1 file2
```

```
rm -i file2 file3
rm -d myfile
rm -r tech
```

Practice 9:

Assume that you are in lab1 directory and there exist a file say test2.c. If not make the file there.

- (a) Type pwd
- (b) ls -al
- (c) cp test2.c test3.c test4.c
- (d) ls -al
- (e) cat test2.c test3.c test4.c
- (f) rm test4.c
- (g) ls -al
- (h) rm -i test3.c
- (i) ls -al

rmdir

This command will remove one or more empty directory.

Syntax:

```
rmdir [options] <directory(ies)>
```

Options

- p remove directory and its ancestors;
For example, 'rmdir -p a/b/c' is similar to 'rmdir a/b/c a/b a'

Example:

```
rmdir student
rmdir tech/net/faqs
```

Practice 10:

- (a) Type pwd
- (b) mkdir lab2
- (c) ls -al
- (d) cd lab2
- (e) Create a file say myFile.txt there
- (f) ls -al
- (g) cd..
- (h) rmdir lab1

- (i) `ls -al`
- (j) `rmdir lab2`
- (k) `ls -al`

grep

The **grep** command allows you to search one file or multiple files for lines that contain a pattern. Exit status is 0 if matches were found, 1 if no matches were found, and 2 if errors occurred.

Syntax:

```
grep [options] <pattern> <file(s)>
```

Options:

- c Display the number of matched lines.
- i Ignore case sensitivity.
- v Display all lines that do NOT match.

Examples:

```
grep "quick" file1
grep -c tech file1
```

find

The *find* command lists all of the files within a directory and its subdirectories that match a set of conditions.

Syntax:

```
find [starting_point] [options] expression
```

Example:

```
find -name myfile.txt -print
find /home/user/myname/ -name myfile.txt -
print
```

cal

This command will print a calendar for a specified month and/or year.

Examples:

```
cal
cal 2008
```

cal 8 2107

date

The date command displays the current day, date, time, and year.

Syntax:

```
date
```

man

This *man* command displays the manual page for a particular command. If you are unsure how to use a command or want to find out all its options, you might want to try using *man* to view the manual page.

Syntax: `man <command>`

Examples:

```
man ls  
man mkdir
```

Important links:

A detail about all GNU commands can be had at (for experienced users)

<https://www.gnu.org/software/coreutils/manual/coreutils.html>

The most frequent unix commands are available at (for intermediate users)

http://www.omg.unb.ca/people/aluizio/unix_commands.pdf

You may consult the link at (for the beginners)

<http://www.maths.manchester.ac.uk/~pjohnson/resources/unixShort/examples-commands.pdf>

C-Programs Writing

1. Write a program to print "Hello World".

```
#include <stdio.h>

main()
{
    printf("Hello, World!\n");

    return 0;
}
```

2. Write a program to take a number as input and print the same number.

```
#include <stdio.h>

main()
{
    int n;

    scanf("%d",&n);
    printf("%d\n",n);

    return 0;
}
```

3. Write a program to take a number as input and print the square of the number.

```
#include <stdio.h>

main()
{
    int n;

    scanf("%d",&n);
    printf("%d\n",n*n);

    return 0;
}
```

4. Write a program to take a number as input and print the reciprocal of the number.

```
#include <stdio.h>

main()
{
    int n;
    scanf("%d",&n);
    printf("%f\n",1.0/n);

    return 0;
}
```

5. Write a program to take two inputs and print their sum as output.

```
#include <stdio.h>
main()
{
    int a,b;
    scanf("%d%d",&a,&b);
    printf("%d\n",a+b);

    return 0;
}
```

6. Write a program to take a number as input and print the sum of natural numbers up to that number as output.

```
#include <stdio.h>

main()
{
    int n;
    scanf("%d",&n);
    printf("%d\n",n*(n+1)/2.0);

    return 0;
}
```

7. Write a C program to input temperature in Celsius (centigrade) and convert to Fahrenheit.

```
#include <stdio.h>

int main()
{
    float celsius, fahrenheit;

    printf("Enter temperature in Celsius: ");
    scanf("%f", &celsius);
    fahrenheit = (celsius * 9.0 / 5.0) + 32;
    printf("%f Celsius = %f Fahrenheit", celsius,
fahrenheit);

    return 0;
}
```

8. Write a C program to swap the values of two given variables.

```
#include <stdio.h>

main()
{
    int a=5,b=10,temp;

    temp = a;
    a = b;
    b = a;
    printf("a=%d b=%d\n",a,b);

    return 0;
}
```

9. Write a C program to calculate the area and perimeter of a rectangle. Take the length and breadth of the rectangle as input.

```
#include<stdio.h>
main()
{
    int length, breadth, area, perimeter;

    printf("Enter the length of a rectangle : \n");
    scanf("%d", &length);
```

```

printf("Enter the width of the rectangle : \n");
scanf("%d", &breadth);

area = length * breadth;
printf("Area of the rectangle : %d\n", area);

perimeter = 2*(length + breadth);
printf("Perimeter of the rectangle : %d\n",
perimeter);

return 0;
}

```

10. Write a C program to find the square root of a number. Take the number as input.

```

#include<stdio.h>
#include<math.h>

main()
{
    int n;

    printf("Enter a number\n");
    scanf("%d", &n);

    printf("Square root of n is %f",sqrt(n));

    return 0;
}

```

11. Write a C program to solve a quadratic equation for given numbers.

```

#include<stdio.h>
#include<math.h>
main()
{
    float a=1.0, b=-5.0, c=6.0;
    float desc, root1, root2;
    desc = sqrt(b * b - 4 * a * c);
    root1 = (-b + desc) / (2.0 * a);
    root2 = (-b - desc) / (2.0 * a);
    printf("\nFirst Root : %f", root1);
    printf("\nSecond Root : %f", root2);

    return 0;
}

```