UNIX

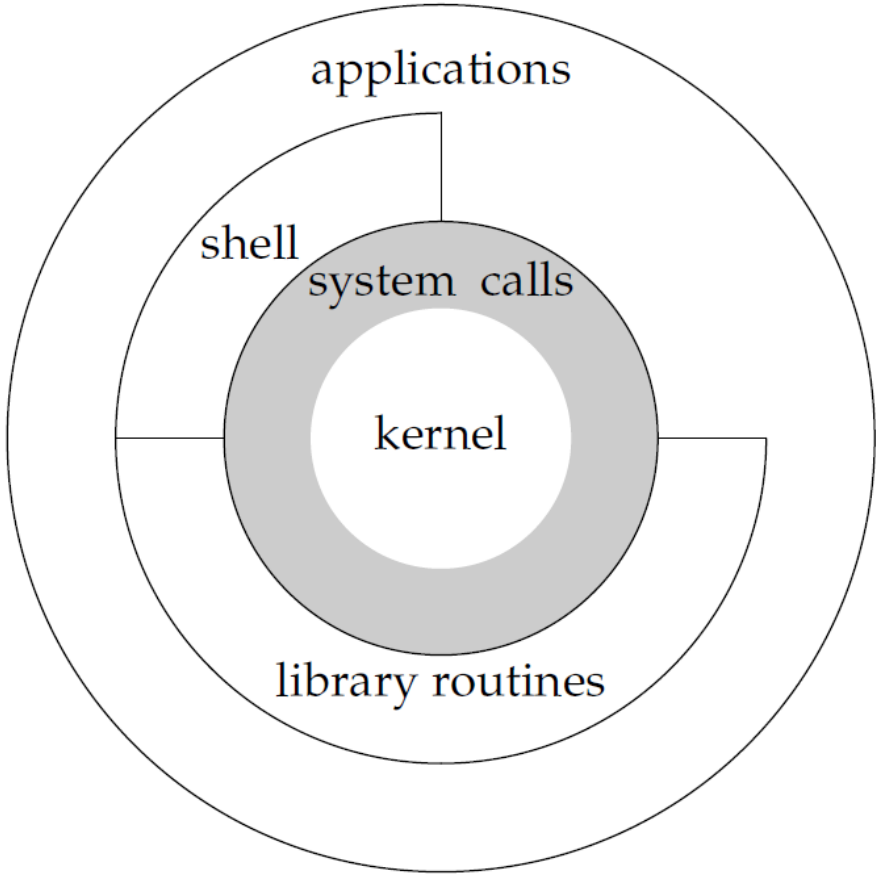
MA658 – UNIX and Network Programming

Overview of UNIX

- An **operating system** can be defined as the software that controls the hardware resources of the computer and provides an environment under which programs can run.
- This software is also called as the **kernel**. It resides at the core of the environment.
- UNIX was originally developed in 1970 by a group of AT&T employees Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna at Bell Labs.

UNIX System Architecture

- The interface to the kernel is a layer of software called the system calls.
- Libraries of common functions are built on top of the system call interface, but applications are free to use both.
- The shell is a special application that provides an interface for running other applications.

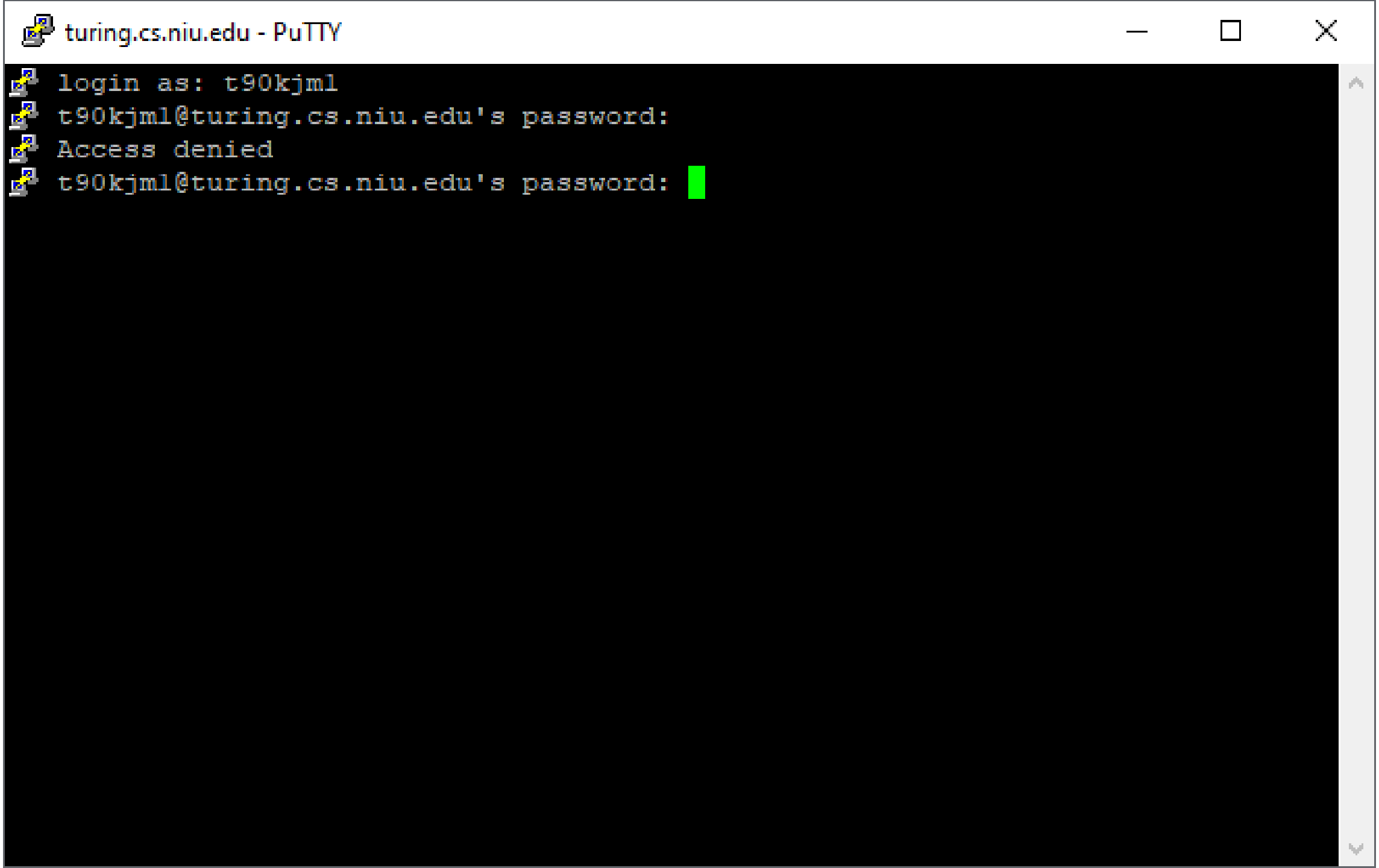


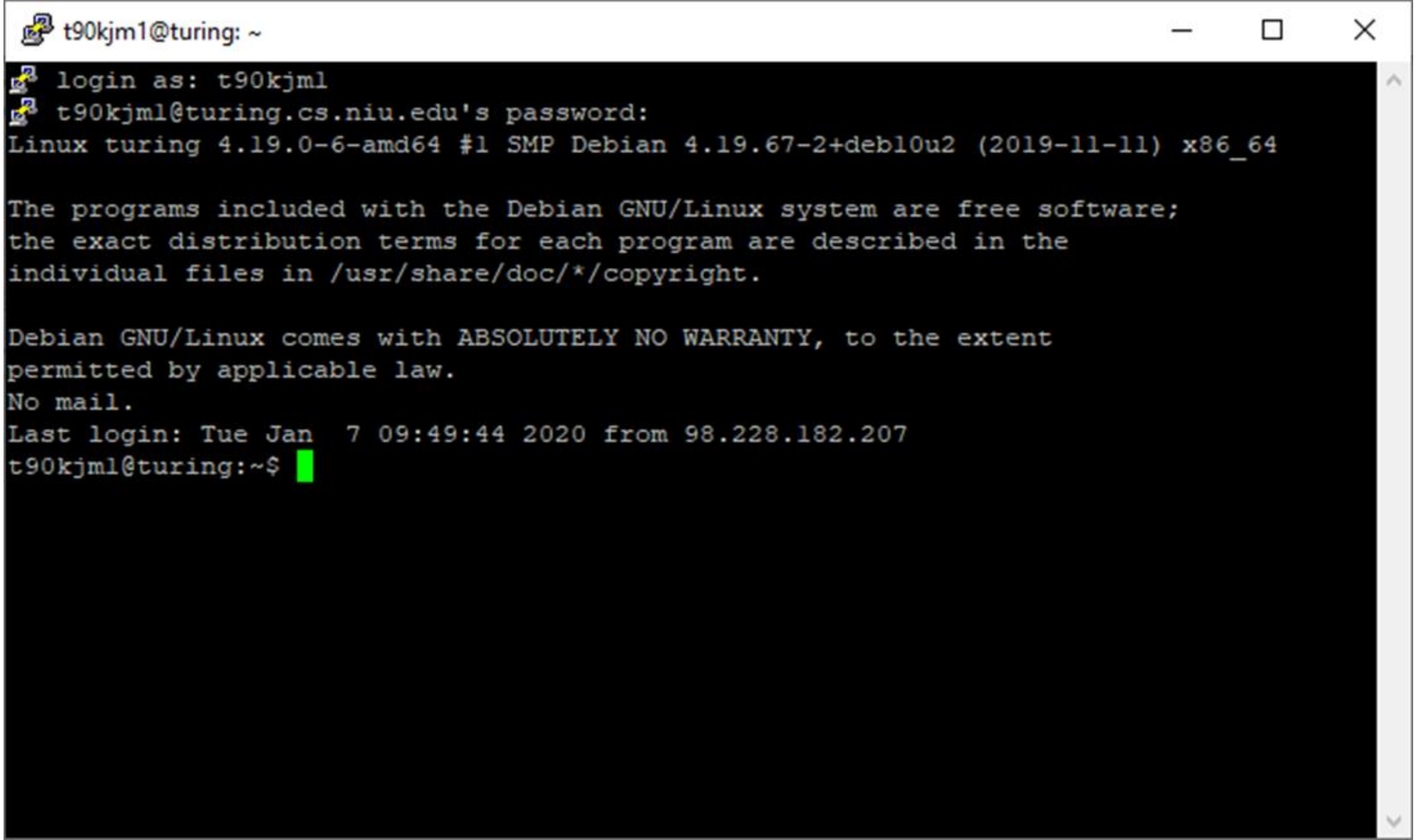
- A **shell** is a command-line interpreter that reads user input and executes commands.
- The user input to a shell is normally from the terminal (an interactive shell) or sometimes from a file (called a shell script).

Logging in…

- When you turn on the UNIX system the prompt will appear like this

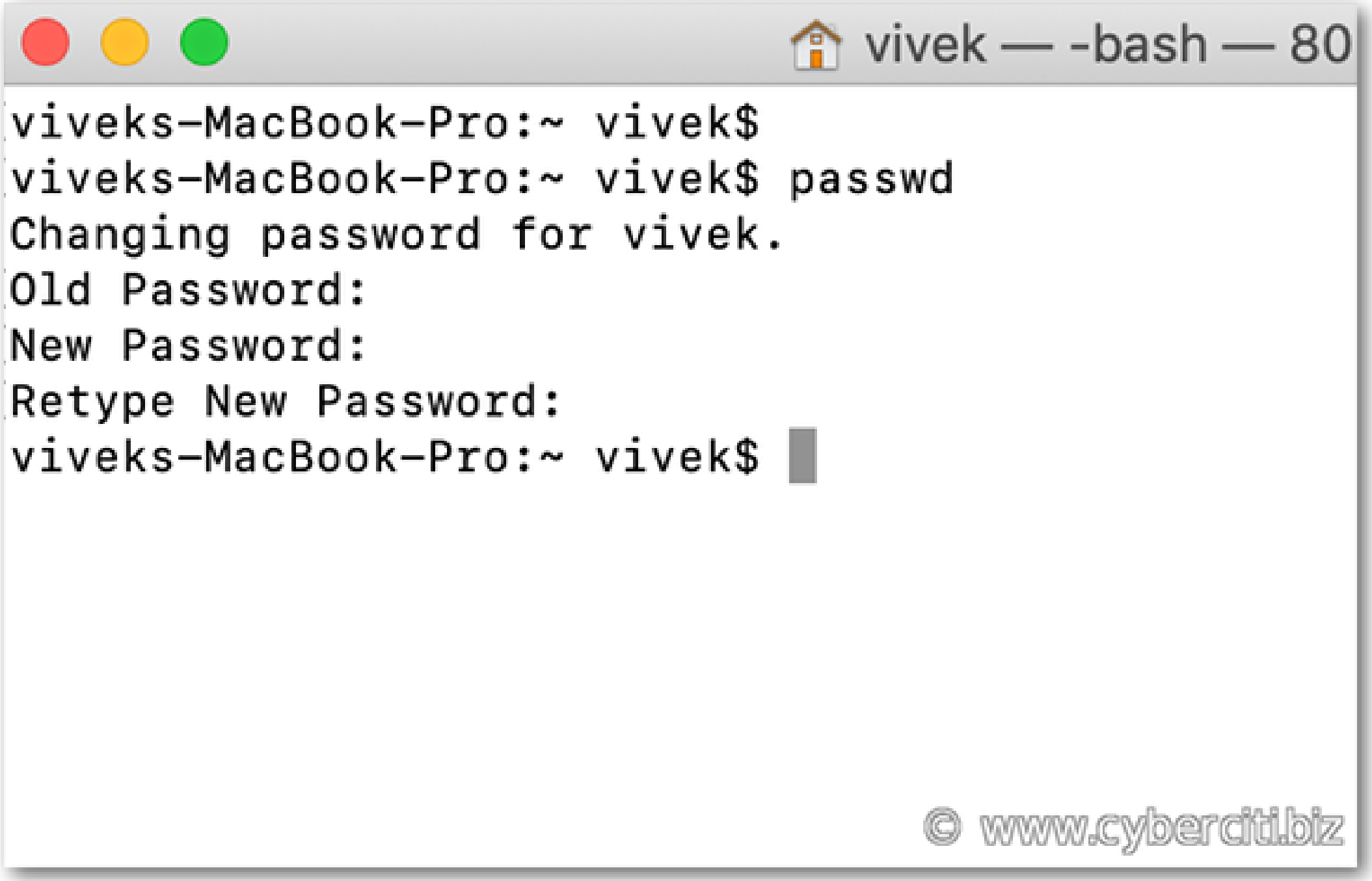

turing.cs.niu.edu - PuTTY
login as: t90kjml
t90kjml@turing.cs.niu.edu's password:


turing.cs.niu.edu - PuTTY
login as: t90kjml
t90kjml@turing.cs.niu.edu's password:
Access denied
t90kjml@turing.cs.niu.edu's password:


t90kjm1@turing: ~
login as: t90kjml
t90kjml@turing.cs.niu.edu's password:
Linux turing 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u2 (2019-11-11) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
No mail.
Last login: Tue Jan 7 09:49:44 2020 from 98.228.182.207
t90kjml@turing:~$

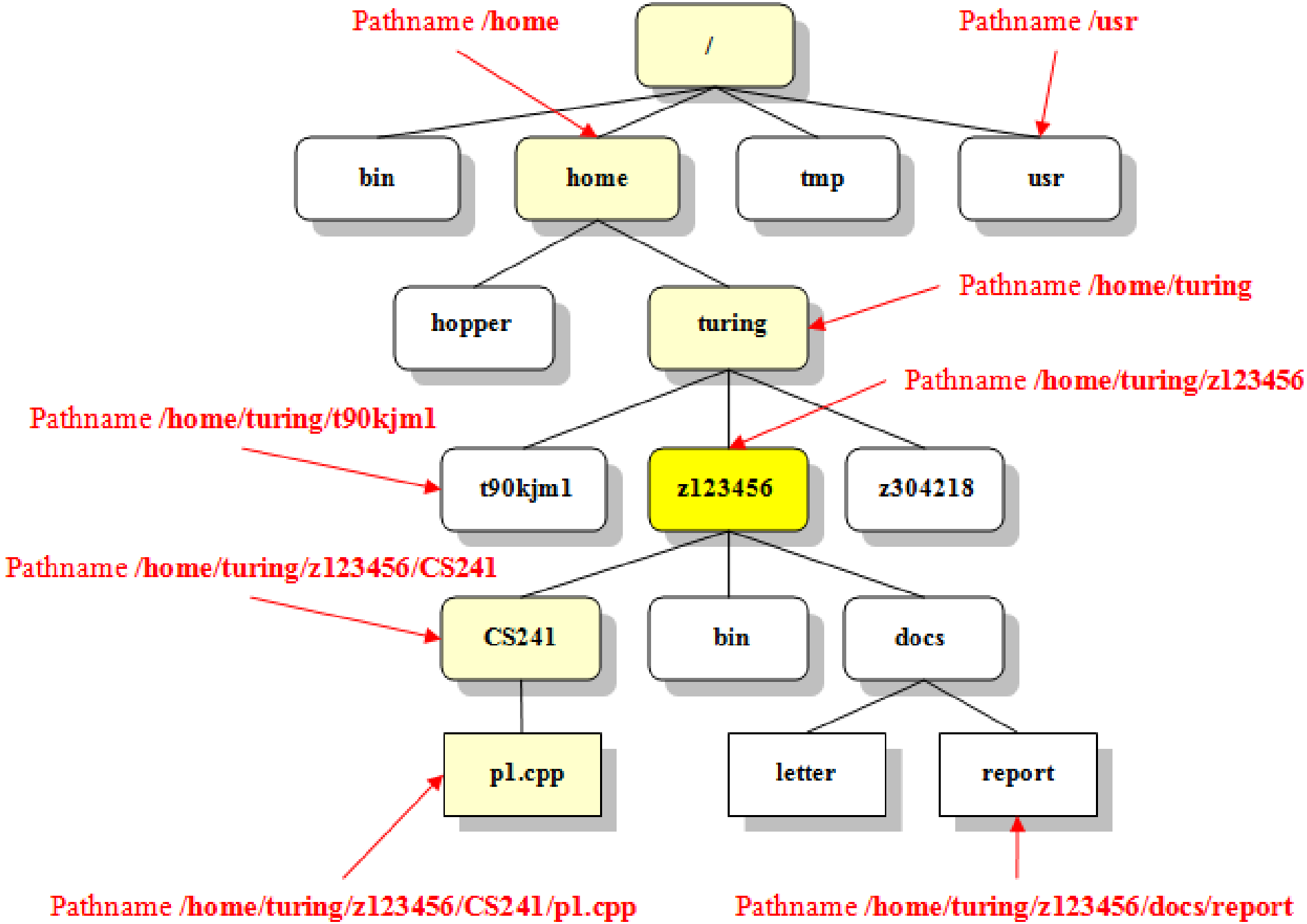
Change Password

- Step 1: To start, type **passwd** at the command prompt.
- Step 2: Enter your old password, the one you're currently using.
- Step 3: Type in your new password.
- Step 4: You must verify the password by typing it again.


vivek — -bash — 80
viveks-MacBook-Pro:~ vivek$
viveks-MacBook-Pro:~ vivek$ passwd
Changing password for vivek.
Old Password:
New Password:
Retype New Password:
viveks-MacBook-Pro:~ vivek$
© www.cyberciti.biz

Files and Directories

- The UNIX file system is a hierarchical arrangement of directories and files.
- Everything starts in the directory called **root**, whose name is the single character /.
- A **directory** is a file that contains directory entries.
- The names in a directory are called **filenames**.
- A sequence of one or more filenames, separated by slashes and optionally starting with a slash, forms a **pathname**.


Pathname /home
/
Pathname /usr
bin
home
tmp
usr
hopper
turing
Pathname /home/turing
Pathname /home/turing/z123456
Pathname /home/turing/t90kjm1
t90kjm1
z123456
z304218
Pathname /home/turing/z123456/CS241
CS241
bin
docs
p1.cpp
letter
report
Pathname /home/turing/z123456/CS241/p1.cpp
Pathname /home/turing/z123456/docs/report

Filenames and Pathnames

- The only two characters that cannot appear in a filename are the slash character (/) and the null character.
- Characters that can be used: letters (a-z, A-Z), numbers (0-9), period (.), dash (-), and underscore (_).
- The slash separates the filenames that form a pathname (described next) and the null character terminates a pathname.

ls command

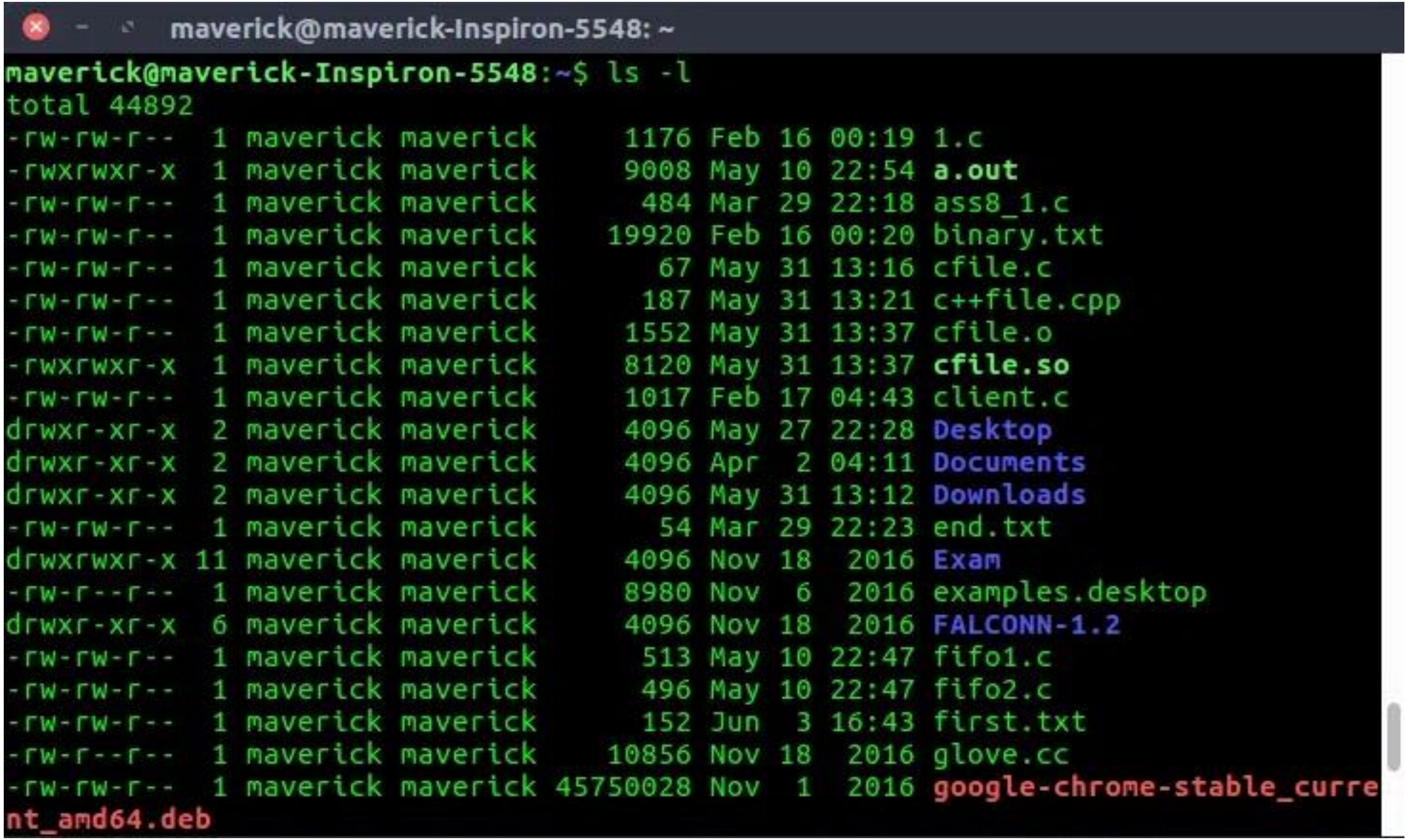
- To list all the files in current working directory command used is **ls**.
- Syntax of `ls` command

  ls [option] [file/directory]

- There are different options that are commonly used with **ls** command.

- -l known as a long format that displays detailed information about files and directories.
- -a Represent all files Include hidden files and directories in the listing.
- -t Sort files and directories by their last modification time, displaying the most recently modified ones first.
- -r known as reverse order which is used to reverse the default order of listing.
- -S Sort files and directories by their sizes, listing the largest ones first.

- -R List files and directories recursively, including subdirectories.
- -i known as inode which displays the index number (inode) of each file and directory.
- -g known as group which displays the group ownership of files and directories instead of the owner.
- -h Print file sizes in human-readable format (e.g., 1K, 234M, 2G).
- -d List directories themselves, rather than their contents.


maverick@maverick-Inspiron-5548: ~
maverick@maverick-Inspiron-5548:~$ ls -l
total 44892
-rw-rw-r-- 1 maverick maverick 1176 Feb 16 00:19 1.c
-rwxrwxr-x 1 maverick maverick 9008 May 10 22:54 a.out
-rw-rw-r-- 1 maverick maverick 484 Mar 29 22:18 ass8_1.c
-rw-rw-r-- 1 maverick maverick 19920 Feb 16 00:20 binary.txt
-rw-rw-r-- 1 maverick maverick 67 May 31 13:16 cfile.c
-rw-rw-r-- 1 maverick maverick 187 May 31 13:21 c++file.cpp
-rw-rw-r-- 1 maverick maverick 1552 May 31 13:37 cfile.o
-rwxrwxr-x 1 maverick maverick 8120 May 31 13:37 cfile.so
-rw-rw-r-- 1 maverick maverick 1017 Feb 17 04:43 client.c
drwxr-xr-x 2 maverick maverick 4096 May 27 22:28 Desktop
drwxr-xr-x 2 maverick maverick 4096 Apr 2 04:11 Documents
drwxr-xr-x 2 maverick maverick 4096 May 31 13:12 Downloads
-rw-rw-r-- 1 maverick maverick 54 Mar 29 22:23 end.txt
drwxrwxr-x 11 maverick maverick 4096 Nov 18 2016 Exam
-rw-r--r-- 1 maverick maverick 8980 Nov 6 2016 examples.desktop
drwxr-xr-x 6 maverick maverick 4096 Nov 18 2016 FALCONN-1.2
-rw-rw-r-- 1 maverick maverick 513 May 10 22:47 fifo1.c
-rw-rw-r-- 1 maverick maverick 496 May 10 22:47 fifo2.c
-rw-rw-r-- 1 maverick maverick 152 Jun 3 16:43 first.txt
-rw-r--r-- 1 maverick maverick 10856 Nov 18 2016 glove.cc
-rw-rw-r-- 1 maverick maverick 45750028 Nov 1 2016 google-chrome-stable_curre
nt_amd64.deb

- – normal file
- d : directory
- s : socket file
- l : link file
- r : read
- w : write
- x : execute

- First Column: Represents the file type and the permission given on the file. Below is the description of all type of files.
- Second Column: Second field specifies the number of links for that file.
- Third Column: Represents the owner of the file. This is the Unix user who created this file.
- Fourth Column: Represents the group of the owner. Every Unix user will have an associated group.
- Fifth Column: Represents the file size in bytes.
- Sixth Column: Represents the date and the time when this file was created or modified for the last time.
- Seventh Column: Represents the file or the directory name.

- Two filenames are automatically created whenever a new directory is created: . (called dot) and .. (called dot-dot).
- Dot refers to the current directory, and dot-dot refers to the parent directory. In the root directory, dot-dot is the same as dot.
- Every process has a working directory, sometimes called the current working directory. (command used **pwd**)
- This is the directory from which all relative pathnames are interpreted.
- A process can change its working directory with the **chdir** function.

Absolute Path Vs Relative Path

- An absolute path is defined as specifying the location of a file or directory from the root directory(/). In other words we can say absolute path is a complete path from start of actual filesystem from / directory.
- Relative path is defined as path related to the present working directory(**pwd**). Suppose I am located in /home/user1 and I want to change directory to /home/user1/Documents. I can use relative path concept to change directory(**cd**) to Documents.

Absolute Path

/Home/user/document/srv.Txt

/Root/data/dev.Jpg

/Var/log/messages

Relative path

$ pwd

/home/user1

$cd Documents/ (using relative path)

$pwd

/home/user1/Documents

or

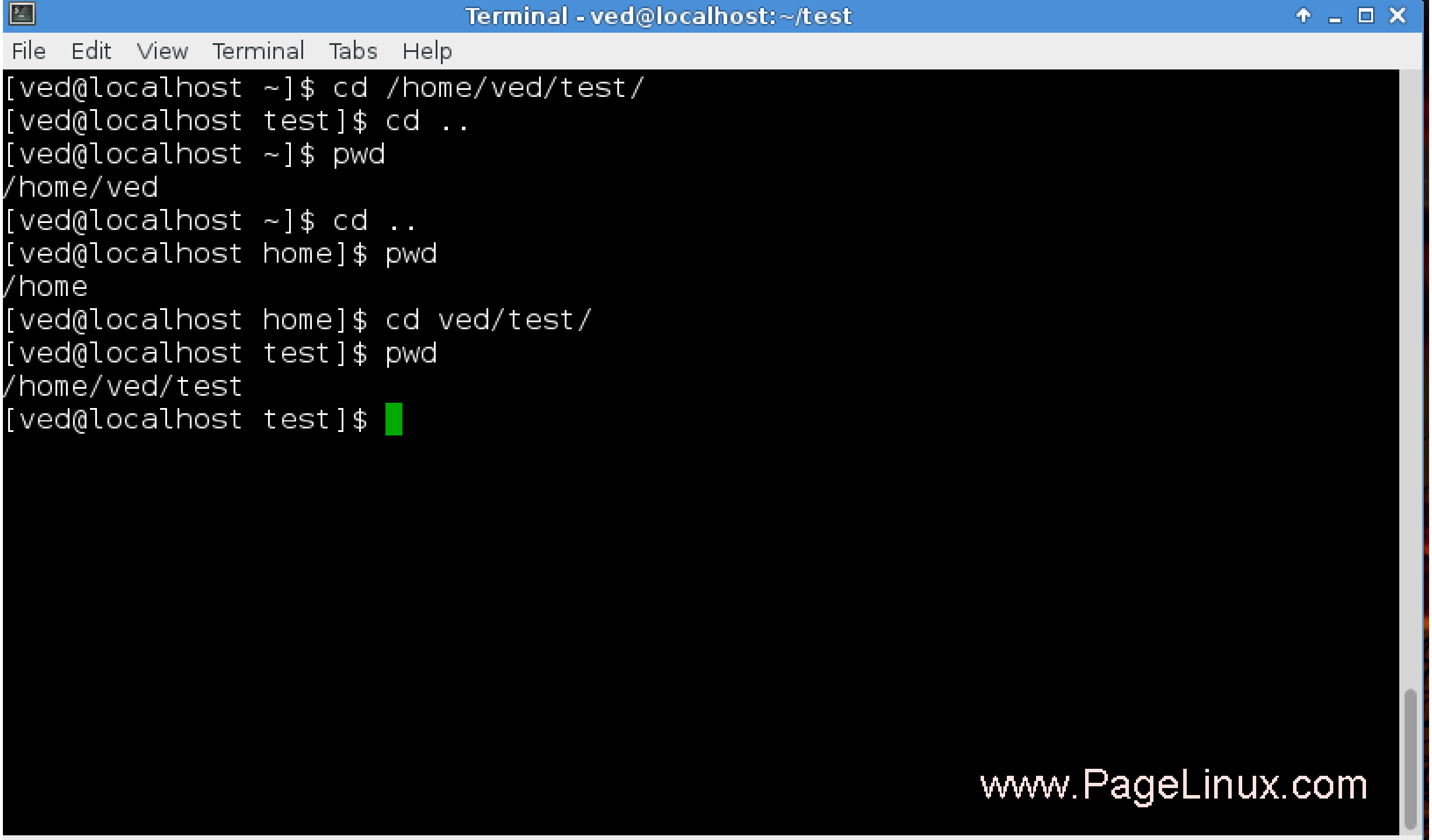
$ pwd

/home/user1

$cd /home/user1/Documents/ (using absolute path)

$ pwd

/home/user1/Documents


Terminal - ved@localhost:~/test
File Edit View Terminal Tabs Help
[ved@localhost ~]$ cd /home/ved/test/
[ved@localhost test]$ cd ..
[ved@localhost ~]$ pwd
/home/ved
[ved@localhost ~]$ cd ..
[ved@localhost home]$ pwd
/home
[ved@localhost home]$ cd ved/test/
[ved@localhost test]$ pwd
/home/ved/test
[ved@localhost test]$
www.PageLinux.com

File I/O

- Most file I/O on a UNIX system can be performed using only five functions: open, read, write, lseek, and close.
- A **file descriptor** is a non-negative integer that the kernel uses to identify the files accessed by a process.
- Whenever it opens an existing file or creates a new file, the kernel returns a file descriptor that we use when we want to read or write the file.

- When reading or writing a file, the file will be identified with the file descriptor that was returned by open or create as an argument to either read or write.
- By convention, all shells open three descriptors whenever a new program is run: standard input, standard output, and standard error.
- UNIX System shells associate file descriptor 0 with the standard input of a process, file descriptor 1 with the standard output, and file descriptor 2 with the standard error.

Input and Output

File Descriptors

- File descriptors are normally small non-negative integers that the kernel uses to identify the files accessed by a process. Whenever it opens an existing file or creates a new file, the kernel returns a file descriptor that we use when we want to read or write the file.