LECTURE 1: INTRODUCTION TO *NIX (UNIX,LINUX) AND SHELL

A few words about this class

- Mostly getting familiar with Unix, C, C++, and programming under Unix and Unix variant

- Not much of theory or profound principles.
  - Well there will be some but we don’t have to be too philosophical

Setting Up the Environment

- Natively running a Ubuntu

- Setting up a Ubuntu over VirtualBox (for any other native OS)
  - Installation will be covered in the next lecture

- Last-resort solution
  - On MacOS, run terminal there
  - Web terminal: www.webminal.org/terminal
  - With caveat:
    - There will be some differences in syntax and other stuff.
    - It is your responsibility to make it work for assignments and class activities.

Class Activities (i.e., Exercises)

- You are expected to do the required activities on your own.
  - But TA will help you a little bit.
  - You need to be persistent, poking around, try out things, you can do web search if you want.
- Each activity will give you a fixed amount of time. You must finish it within that time.

Outline

- **Unix operating systems**
- Terminal and shell
- Unix file systems

Unix

- The term Unix refers to any Unix-like systems such as Linux, Solaris, Mac OS, BSD, etc.
- 1969: UNiplexed Information and Computing Service
- 1971: V1 (60 commands)
- 1973: V3 (Pipe, C language)
- 1976: V6 (rewritten in C, base for BSD)
- 1979: V7 (Licensed, portable)

Linux

- Written in 1991 by Linus Torvalds
- 2001: Linux Kernel v. 2.4
- Most popular Unix variant
- Free with GNU license
- Ported to many different hardware platforms

Free BSD

- 1993: FreeBSD (focuses on PCs)
- 1993: NetBSD (focuses on portability)
- 1996: OpenBSD (focuses on security)
- Free with BSD license

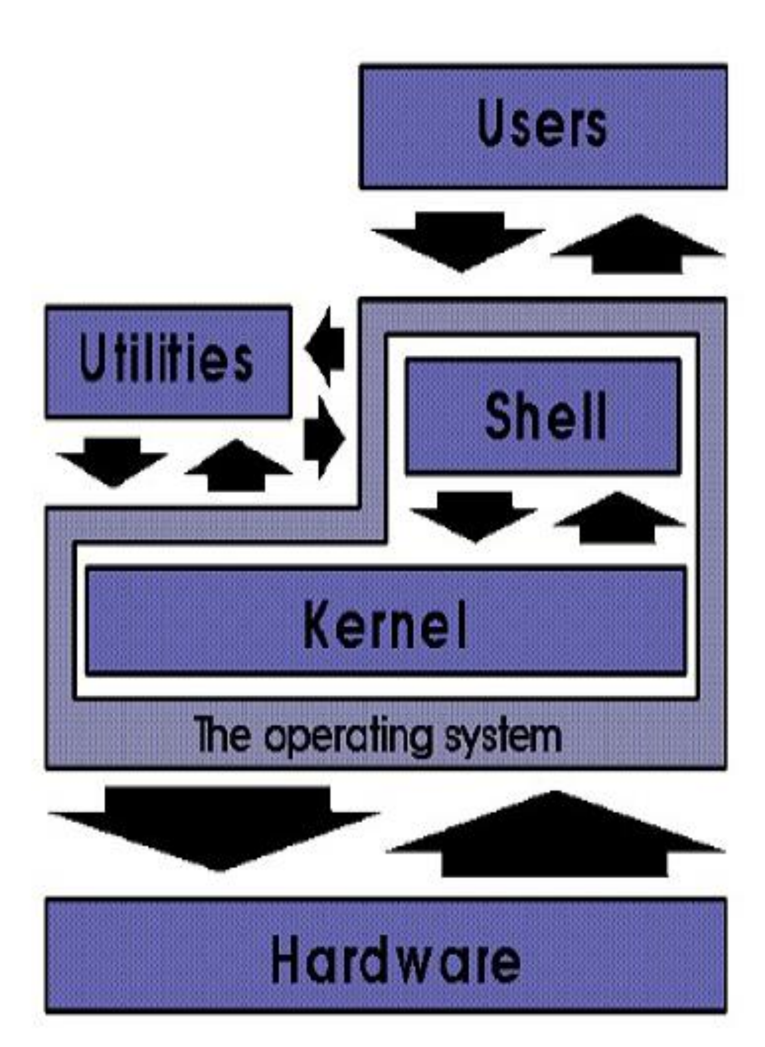
Mac OS X

- Based on 4.4 BSD-Lite
- Built its own GUI on top of BSD base
- You know the rest!

Unix Philosophy

- Small is beautiful
  - Easy to understand
  - Easy to maintain
  - Efficient
  - Reuse
- Make each program (commands) simple and do one thing well
  - Complex functionality can be achieved by combining commands.

A brief overview of OS



Shell: interpreter between the kernel and the user

Utilities: compilers, commands, others

Outline

- Unix operating systems
- **Terminal and shell**
- Unix file systems

Terminal and Shell

- Open a terminal
  - Ctrl+Alt+T

- Terminal gives you a shell
- What is a shell program?
  - That’s the one which prints out the prompt and wait for a user input in terminal
  - A command interpreter that translates the user’s commands for the Kernel to carry out the user’s commands.

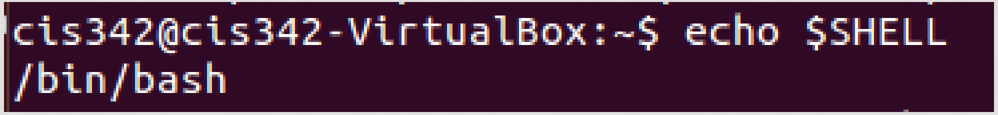
Why would I want to use terminal

- We want to learn how to program using only keyboard
- It gives us access to more advanced features
- A GUI is after all a “wrapper” for command line
- In overall, it makes you a more advanced programmer, system administrator, scientist, engineering, etc.

Types of Shell

- There are varieties of shell programs.
  - C shell (csh)
    - Developed by BSD Unix
  - Bourne shell (sh)
    - Developed by Unix System V; compact and simple
  - Korn shell (ksh)
    - Developed by David Korn; extended Bourne shell
  - Turbo shell (tsh)
    - Enhanced csh
  - GNU shell (bash)
    - Most advanced and default for Ubuntu users
  - And more…..

- Which Shell I am using:
  - Type

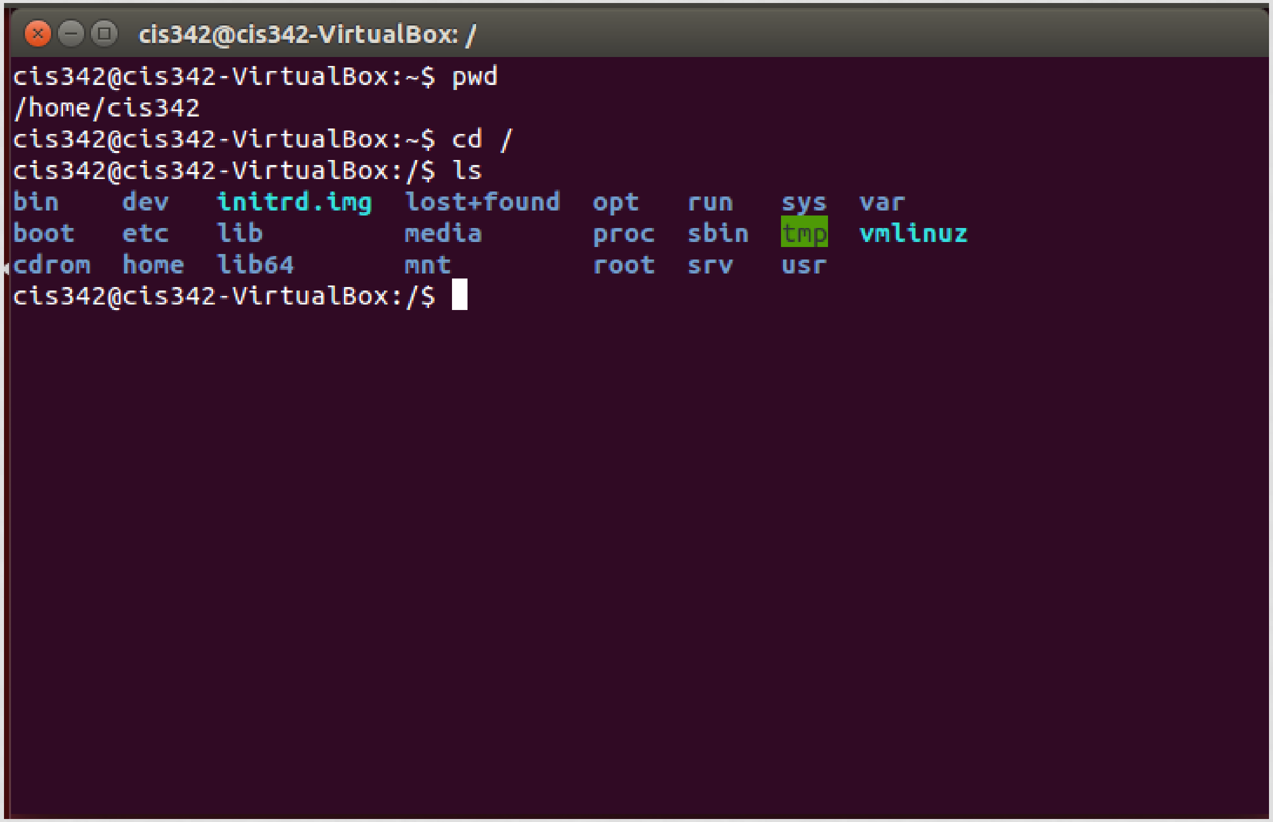


```
cis342@cis342-VirtualBox:~$ echo $SHELL
/bin/bash
```

  - The echo command will display what it was told to.

Shell Commands

- Type “ls”



```
cis342@cis342-VirtualBox:~$ pwd
/home/cis342
cis342@cis342-VirtualBox:~$ cd /
cis342@cis342-VirtualBox:/$ ls
bin     dev    initrd.img  lost+found  opt   run   sys  var
boot    etc    lib         media       proc  sbin  tmp  vmlinuz
cdrom   home   lib64       mnt         root  srv   usr
cis342@cis342-VirtualBox:/$ 
```

ls: list directory

Some special characters

- /: path separator
  - /home/cis342
- \: remove the special meaning of the character following
  - cd cis342/Good\ Student
- Wildcard characters
  - ls *: display all files
  - ls c*: display all files starts with c
  - ls *c*: display all files that has a c
    - Note that * can be empty character
  - ls ?c?: display all files start with any single character, then c, and ends with any single character.
  - ls c[ih]: match any one character within [..]
  - ls c[ih]*: match any one character within […] and any following chracter(s).

Some special characters

- ; : command separator
  - ls –F; cd Documents
  - ls –F; cd Documents; ls; cd ..
- More later
  - (): command group
  - >, <, >> : redirections.
  - | : pipe

Shell Commands: Format

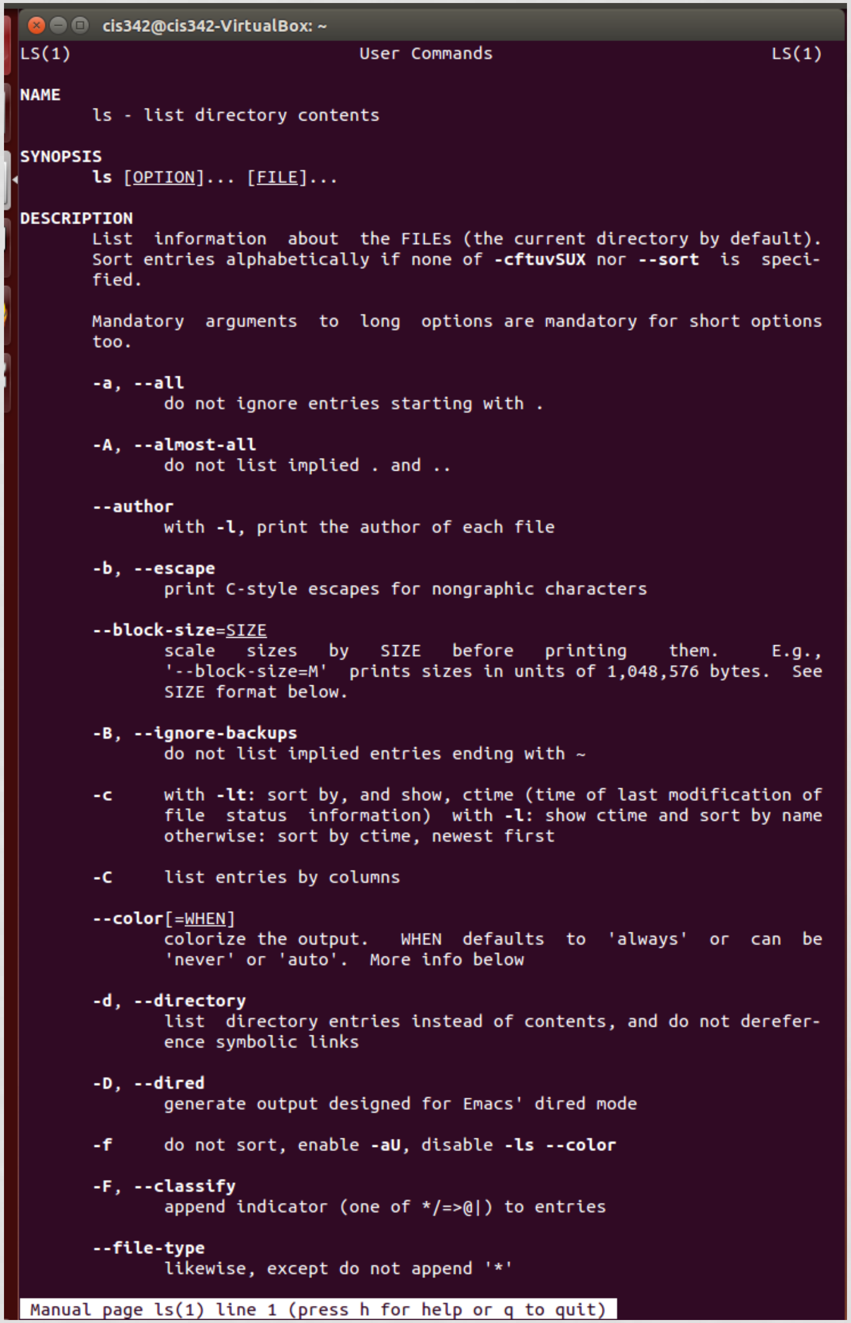
- All commands in terminal has
  - The command itself, and
  - Parameters
- cd ~
- cd /
- cd /home/cis342

Shell Commands: Getting Help

- Some commands can take a parameter
  - "-h" or "--help"

- For example,
  - gcc –help

- You can also get the **manual** page for a command
  - Try "man ls"



```
LS(1)                          User Commands                         LS(1)

NAME
       ls - list directory contents

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

DESCRIPTION
       List  information  about  the FILEs (the current directory by default).
       Sort entries alphabetically if none of -cftuvSUX nor --sort  is  speci-
       fied.

       Mandatory  arguments  to  long  options are mandatory for short options
       too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file

       -b, --escape
              print C-style escapes for nongraphic characters

       --block-size=SIZE
              scale   sizes   by   SIZE   before   printing    them.    E.g.,
              '--block-size=M'  prints sizes in units of 1,048,576 bytes.  See
              SIZE format below.

       -B, --ignore-backups
              do not list implied entries ending with ~

       -c     with -lt: sort by, and show, ctime (time of last modification of
              file  status  information)  with -l: show ctime and sort by name
              otherwise: sort by ctime, newest first

       -C     list entries by columns

       --color[=WHEN]
              colorize the output.   WHEN  defaults  to  'always'  or  can  be
              'never' or 'auto'.  More info below

       -d, --directory
              list  directory entries instead of contents, and do not derefer-
              ence symbolic links

       -D, --dired
              generate output designed for Emacs' dired mode

       -f     do not sort, enable -aU, disable -ls --color

       -F, --classify
              append indicator (one of */=>@|) to entries

       --file-type
              likewise, except do not append '*'

 Manual page ls(1) line 1 (press h for help or q to quit)
```

Read the man page carefully

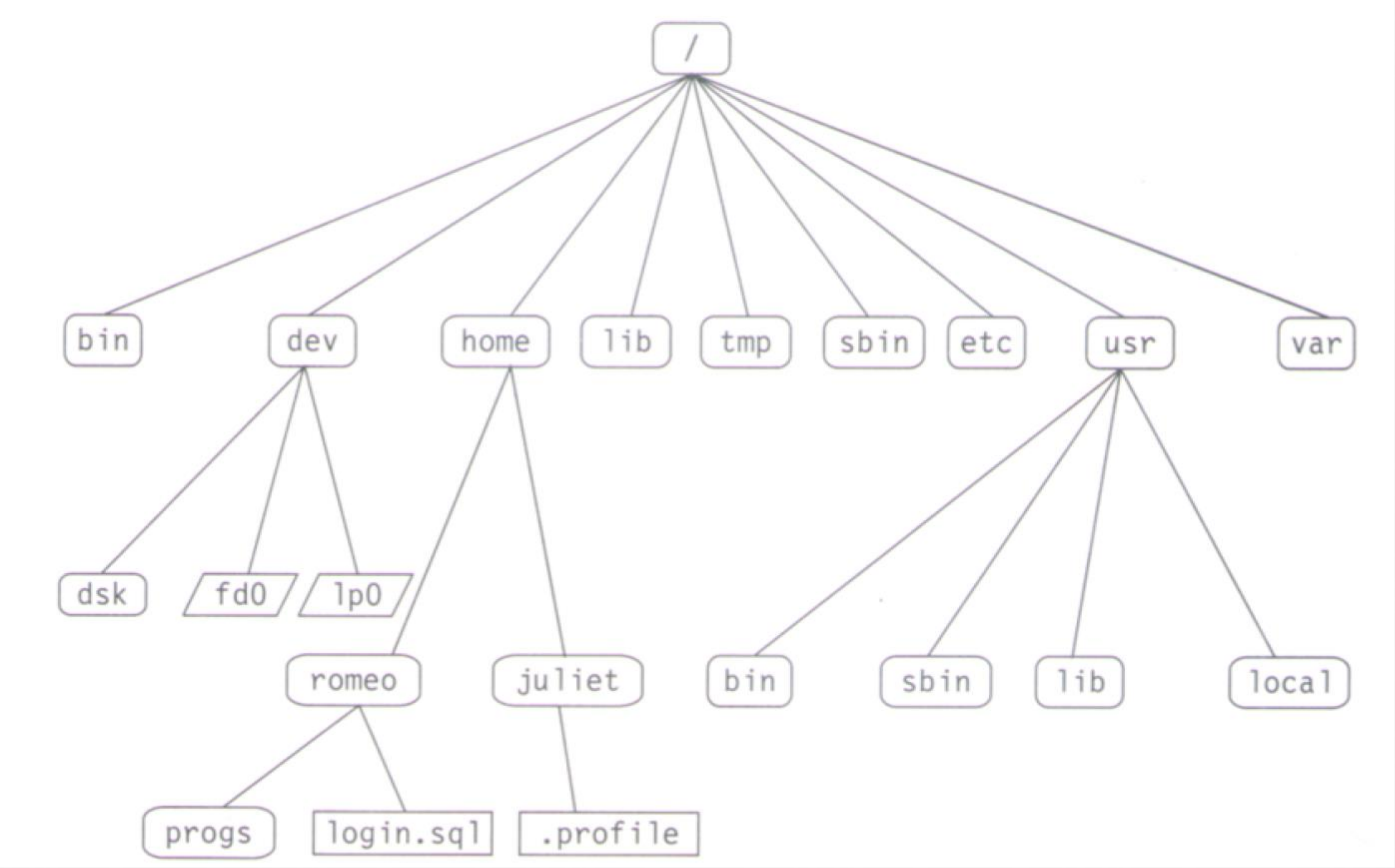
Outline

- Unix operating systems

- Terminal and shell

- **Unix file systems**

File System View

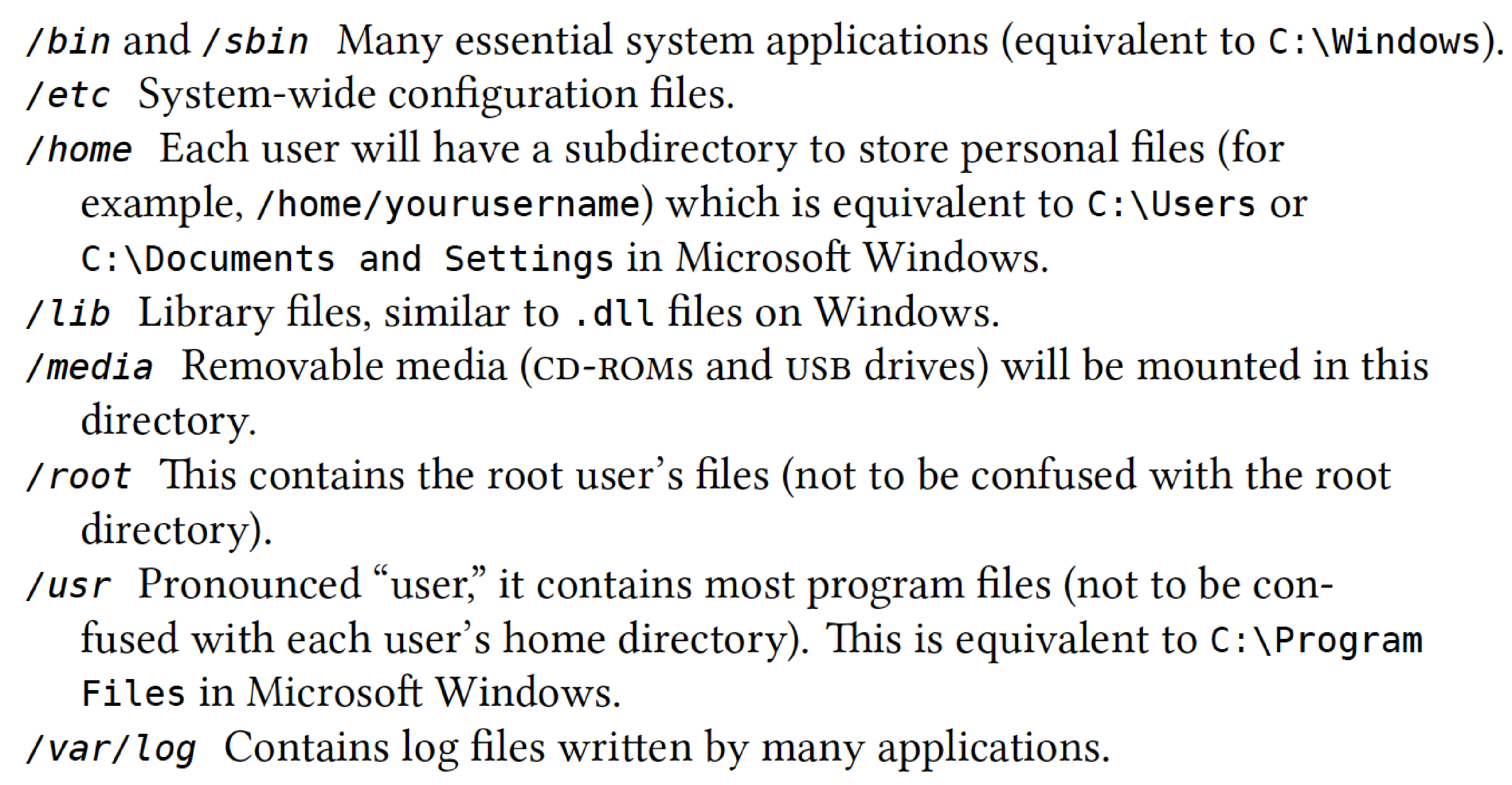


Ubuntu File System Directory (More detail)

- Directories and files starting with a period are hidden.
  - This means “ls” would not do it
  - ls –a would display these
  - These files are usually used by a program, a command, a shell, etc.

- We will discuss these later.

Ubuntu File System Directory (More detail)



/bin and */sbin* Many essential system applications (equivalent to C:\Windows).

/etc System-wide configuration files.

/home Each user will have a subdirectory to store personal files (for example, /home/yourusername) which is equivalent to C:\Users or C:\Documents and Settings in Microsoft Windows.

/lib Library files, similar to .dll files on Windows.

/media Removable media (CD-ROMs and USB drives) will be mounted in this directory.

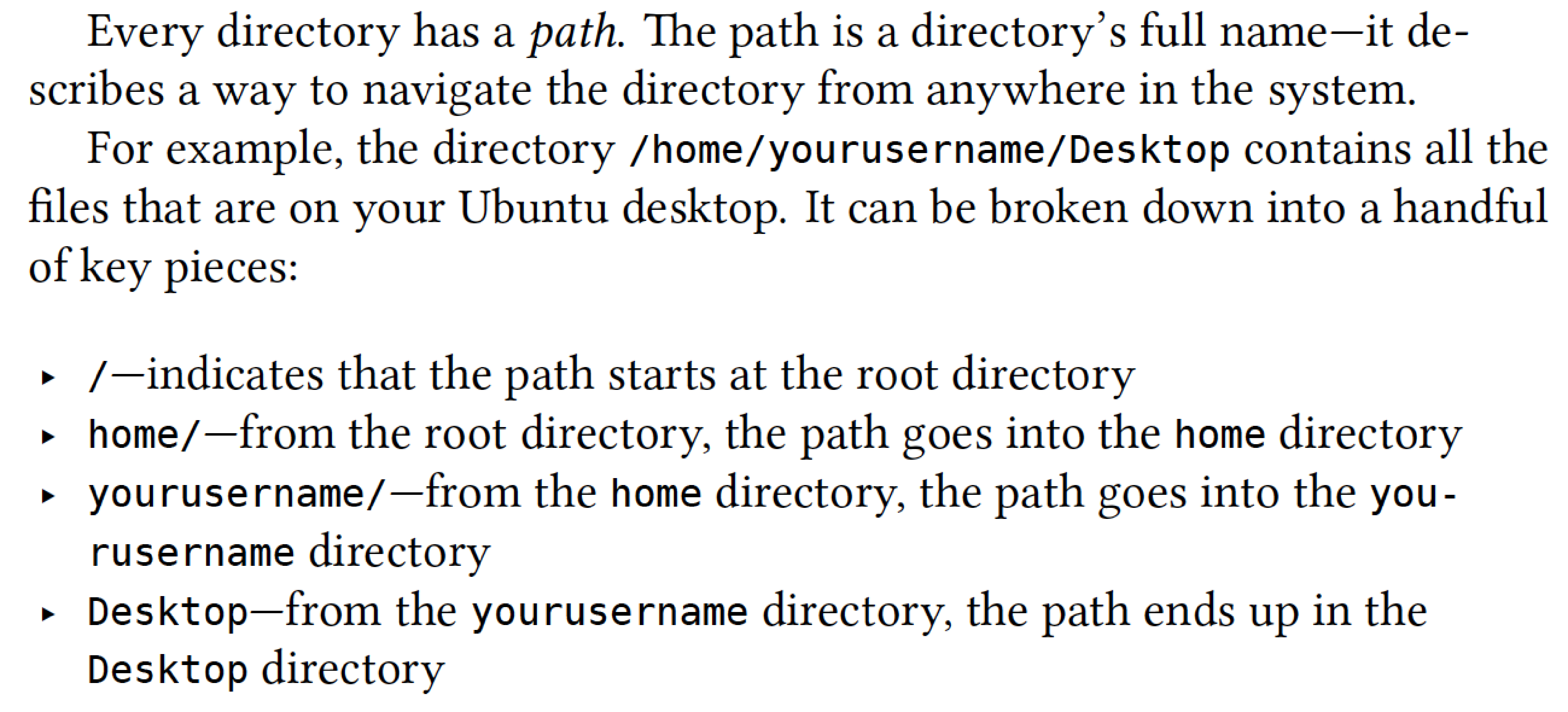
/root This contains the root user's files (not to be confused with the root directory).

/usr Pronounced "user," it contains most program files (not to be confused with each user's home directory). This is equivalent to C:\Program Files in Microsoft Windows.

/var/log Contains log files written by many applications.

From Ubuntu manual

Ubuntu File System Directory (More detail)



Every directory has a *path*. The path is a directory's full name—it describes a way to navigate the directory from anywhere in the system.

For example, the directory `/home/yourusername/Desktop` contains all the files that are on your Ubuntu desktop. It can be broken down into a handful of key pieces:

- `/`—indicates that the path starts at the root directory
- `home/`—from the root directory, the path goes into the `home` directory
- `yourusername/`—from the `home` directory, the path goes into the `yourusername` directory
- `Desktop`—from the `yourusername` directory, the path ends up in the `Desktop` directory

From Ubuntu manual