CSCI 2132
Software Development

Lecture 3:

Unix Shells and Other Basic Concepts

Instructor: Vlado Keselj

Faculty of Computer Science

Dalhousie University

Previous Lecture

- **Introduction to UNIX**
- Reading: Chapter 1 of the UNIX book
- Operating System overview
  - Onion skin model
- OS functionalities
- UNIX history
- UNIX philosophy and main features

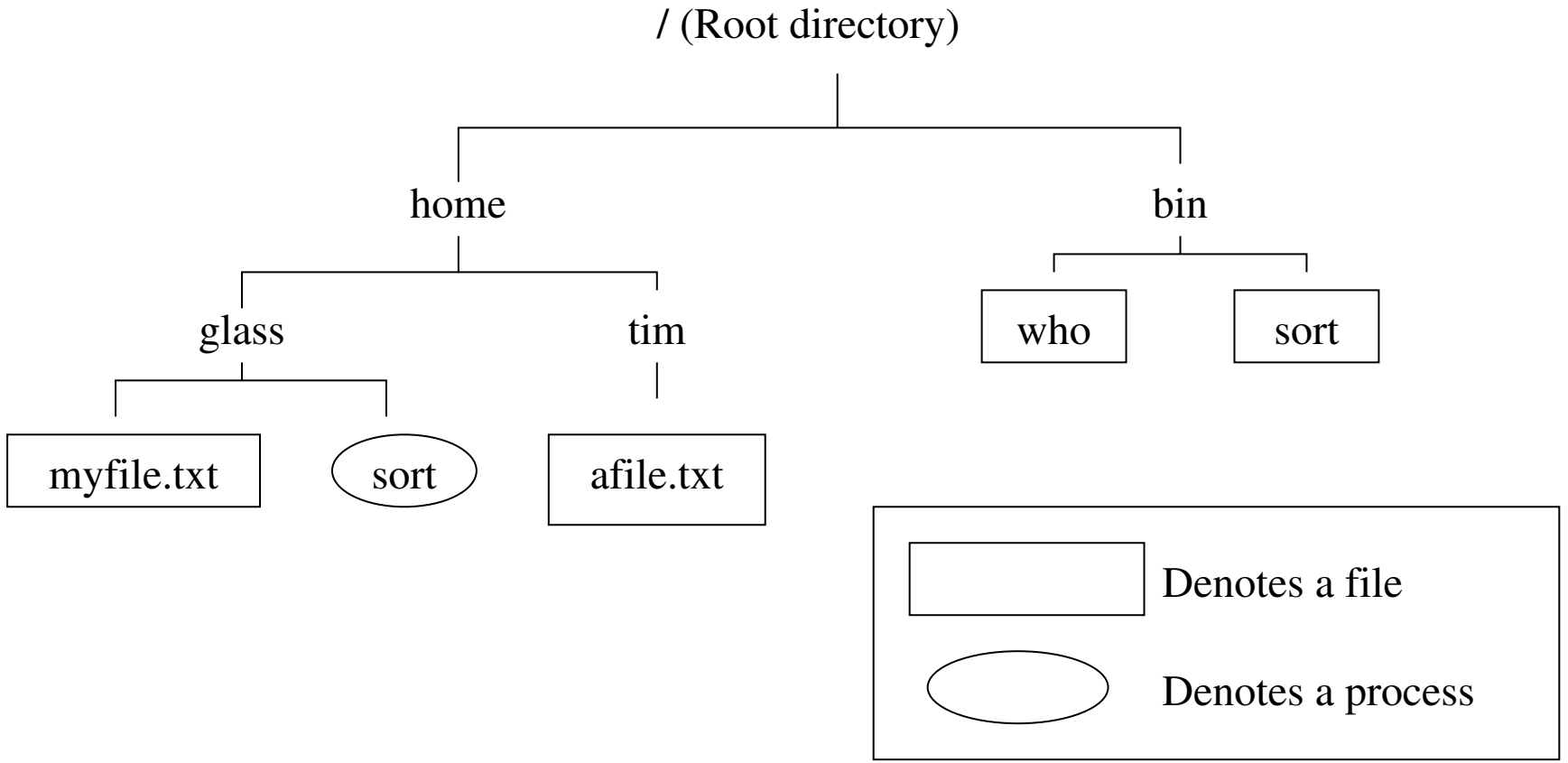
Some Hardware Concepts

- Central Processing Unit (CPU)
- Random-Access Memory (RAM)
- Read-Only Memory (ROM)
- Disk Memory (Hard Disk Drive, CD-ROM, etc.)
- Graphics card
- Network card (Ethernet card)
- Peripherals: keyboard, monitor, mouse, etc.

Some Important Unix Concepts

- **File:** collection of data (sequence of bytes) stored on disk (or other external storage)
- **Program:** a file containing machine code and data that can be loaded into memory and run
- **Process:** Program that is loaded in memory and running
- **Owner** of a File or Process: Processes and files have owners and may be protected against unauthorized access
- **Hierarchical directory structure** is suppored in Unix file system
- **Location** of a file or process: Files and processes have a location within the directory hierarchy. A process may change its location and the location of a file
- **System Calls:** Unix provides services (system calls) for creation, modification, and destruction of files and processes

Directory Hierarchy



Shells

- a program used by the user to interact with the system
- used to run other programs in the system
- built-in commands and utilities (external commands)
- used to automate many tasks
- A UNIX characteristic is existence of multiple shells (even Window managers); e.g.:
  - **Bourne Shell:** sh, bash (Bourne-Again Shell); we will focus on this one
  - Korn Shell: ksh
  - C Shell: csh, tcsh
  - Z Shell: zsh

Getting Started in Unix

- Covered in the lab
- Mandatory exercise: login to your bluenose account using ssh
- Options: putty, MobaXterm, Mac terminal, Linux terminal
- More options: use of VirtualBox
- Main learning objective:
  - be able to open one or more Terminals with ssh login to your bluenose account in each of them
- ask TAs, use Learning Center if there are any issues

Logging In

- You can choose Windows or Mac environment in some labs
- Windows: you can use the `putty` program
- On Mac: open a Terminal and type:
  `ssh` *CSID*`@bluenose.cs.dal.ca`
- Instead of *CSID* use your CS userid (CSID)
- On Linux: similarly to Mac, you open the terminal and type the same command:
  `ssh` *CSID*`@bluenose.cs.dal.ca`

Shell Prompt

- After login, we communicate with a shell program, which responds with a *prompt* string; e.g.:

  ```
  CSID@bluenose:~$
  ```

- or, some shells respond with `%` or `#`
- User can change the prompt string
- We type commands, followed by enter key, and the shell will respond
- We can edit the command line before pressing Enter

Running a Utility

- Enter a command, which can be a utility or a built-in command
- A utility is a program with the same name: shell finds it and runs it
- Example: `date`
- Output:

```
Tue Sep 10 22:35:49 ADT 2013
```

- More examples:
  - `clear`
  - `passwd`

Command-line Arguments

- Utilities can accept arguments
- For example, date utility allows us to choose format of the output, as in:

```
date +%Y-%m-%d-%H-%M-%S
```

- Output:

```
2013-09-10-22-35-49
```

- To explore usage of ‘date’, type:
  `man date`
  (use ‘q’ to exit)

The ‘man’ Utility

- `man` comes from ‘manual pages’
- the command ‘man’ is the standard way of providing usage descriptions for various commands in Unix
- examples:
  - `man man`
    to read about the command ‘man’
  - `man -k directory`
    to search commands using keyword ‘directory’
  - `man 2 rmdir`
    to specify the section number of the man page in case of an ambiguity

Special Shell Characters (Metacharacters)

- Characters which are interpreted in a special way when typed in a Unix terminal
- They are usually control characters obtained by pressing Ctrl key and then typing another key while Ctrl is pressed, e.g., Ctrl-C
- We will learn two of them:
  - Control-C (`^C`) is used to terminate a process
  - Control-D (`^D`) is used to signal end of file
- The command `stty -a` can be used to show many special characters

Input, Output, and Error Channels

- Remember that there are three default I/O channels for each UNIX program, i.e., three **standard I/O streams:**
  - stdin (standard input)
  - stdout (standard output)
  - stderr (standard error)
- Commands so far mostly used stdout only
- Command ‘cat’ frequently uses stdin and stdout

‘cat’ example

- Consider the following example (`^D` is Control-D):

```
$ cat > hamlet.txt
To be or not to be
that is the question
^D
$ cat hamlet.txt
```

Editors

- The ‘cat’ example was a way to create a textual file
- A more convenient way is to use a textual editor
- Unix offers several editors, for example:
  - emacs — to be used as main editor in this course,
  - vi or vim — also a major Unix editor,
  - pico, nano, and some others
- Editors are covered more in the labs

Logging Out

- On some shells typing Control-D is sufficient to log us out
- Sometimes it is disabled since a user can easily type it by mistake
- Commands ‘logout’ and ‘exit’ are available