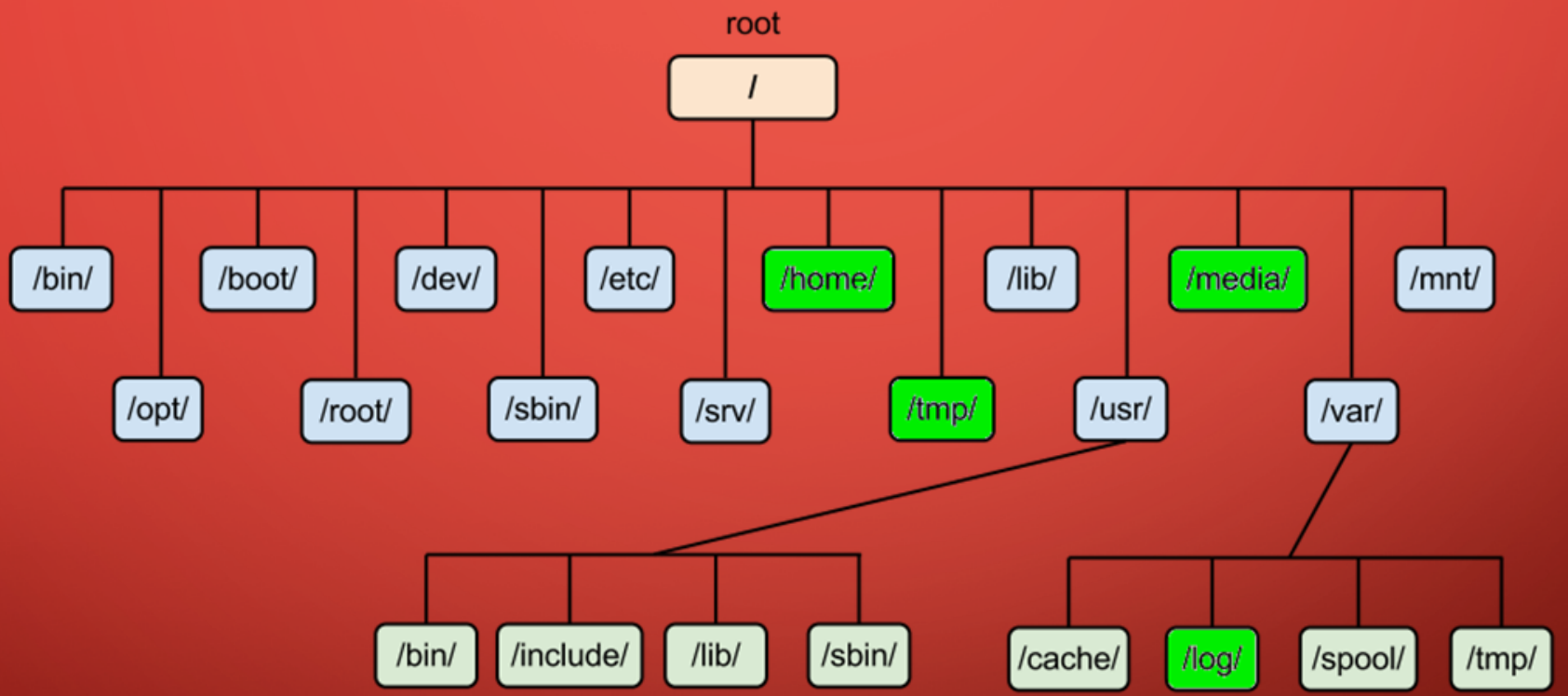
USING THE SHELL

THE LINUX/UNIX COMMAND LINE INTERFACE

Tony Ross – W7EFS
John Hays – K7VE

THE FILE SYSTEM



GETTING TO THE SHELL

- Open a Terminal
  - Automatically attaches to the shell (bash shell on your pi)
- Use ssh to login to the system remotely
  - From Linux/Unix: ssh pi@<ip address or domain name>
  - From Apple MacOS: ssh pi@<ip address or domain name>
  - From Windows, use a program like:
    - Bitvise
    - PuTTY

FIRST COMMANDS - NAVIGATION

- pwd
  - Tells you what directory you are currently in, e.g. /home/pi
- cd
  - Go to your home directory, e.g. /home/pi
  - cd .. Go to the parent directory, e.g. /home
  - cd ~bill Go to the home directory for account 'bill', e.g. /home/bill
  - cd Documents Go the directory name Documents that is a child of your current directory, e.g. if you are in /home/pi, go to /home/Documents

LEARN ABOUT YOUR SHELL

- printenv
  - Returns many lines, including:

LOGNAME=pi
TERM=xterm SSH_CONNECTION=192.168.100.145 37960 192.168.100.176 22
SHELL=/bin/bash

LOGNAME=pi

SSH_CONNECTION=192.168.100.145 37960 192.168.100.176 22

SSH_TTY=/dev/pts/1

PWD=/home/pi

USE ENVIRONMENT VARIABLES

- echo $PATH

  Returns:

  /home/pi/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

- PATH=$PATH:/foo

  Adds /foo to directories where the shell will look for commands

  /home/pi/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/foo

EXECUTE PROGRAMS / COMMANDS

- List the current directory
  pi@compass:~ $ ls
  bin descriptions.txt Desktop Documents Downloads Music n7nix oldconffiles Pictures Public python_games Templates Videos
- The program must exist in a directory in your PATH
  pi@compass:~ $ which ls
  /bin/ls
  You can also use the full path name, e.g. Execute /bin/ls

STANDARD I/O

- The programs and commands use standard I/O streams for input and output. You can redirect the streams:

  Command > filename.txt (send output to a file)
  Command >> filename.txt (append output to a file)
  Command < filename.txt (take input from a file)
  Command1 | Command2 (take the output of command 1 as input to command 2 - pipe)
  Command 2> errorfile.txt (send errors to a file)
  Command &> both.txt (send output and errors to a file)
  Command1 < input.txt | Command2 > output.txt &> error.txt (combine as needed

HELPFUL COMMANDS AND TRICKS

- Editors: pico and vi (vim)
- Help with commands: man command (manual) or command –h
- Command and filename completion: type a few characters and hit tab
- sudo command (run a command with super user permissions)
- chmod and chown (set permissions and ownership of files and directories)
- sudo find <path> -name “filename” (locate files in the filesystem)

HELPFUL COMMANDS AND TRICKS

- touch filename (create an empty file)
- ps –aux (see what processes are running, ‘kill’ to kill a process)
- top (see what is using your cpu and memory, hit 1 to see individual cpus)
- rm filename (remove a file, ‘rm –r’ to remove recursively)
- who (see who is logged in, use ‘last’ to see previous logins)
- And many more

LEARN REGULAR EXPRESSIONS

- https://linux.die.net/Bash-Beginners-Guide/chap_04.html
- Powerful in tools like editiors and commands:
  Use inside of vi/vim
  sed (stream editor, e.g. ls | sed 's/Do/Whoopie/')
  grep (find strings in a file, e.g. grep 'error|warning' file.log
  tail (look at the end of a file, or follow it, e.g. tail –f /var/log/syslog)
  …

QUESTIONS AND ANSWERS