May 30, 2017

Creating, deleting and managing directories

* mkdir, cd, pwd, rmdir
* cd: change directory
* pwd: present work directory. Find out the current directory
* mkdir: make directory
* rmdir: remove directory
* mkdir test: create test directory in current directory
* rmdir: directory must be empty (use -r to overwrite it)

RM (remove directory)

* Syntax: mkdir [options] directory-names
* Purpose: create directory or directories specified by directory-names
* Output: new directory or directories
* Common options:
  + -m: MODE create a directory with a given access mode
  + -p create parent directories that don’t exist
* Inode is a data structure contained information of the files.

CD (change directory)

* Syntax: cd [directory]
* Purpose: changing the current directory to the specified dir or return to the home direcroty when there is no argument
* Output: the new directory
* Common Options: directory

PWD (parent working directory)

* Syntax: pwd
* Purpose: to find out the current dir
* Output:
* Common options:

RM (remove directory)

* Syntax: rmdir [options] directory-names
* Purpose: removes the empty directories specified
* Output: removed directories
* Common options: -p: also removes empty parent directories
* Syntax: rm -r: delete non-empty directory (recursively descends down into the subdirectory and delete any files in it before actually deleting the directory itself. Be careful with these command.
* The rm -r command first delete the files and then the empty directory and stops at current directory.

Disk Utilities

* du: disk usage
* Estimate the file space usage on the disk
* It produces a list containing the usage of each subdirectory of its argument and finally produces a summary
* $ du /home/user1
* who: know the users
* Displays the users currently logged in the system
* $ who
* whoami: Show the owner of this account
* $ whoami
* w: Tell you who is logging in and doing what!
* $ w
* AT&T and BSD Linux. Major branches of linux. TCP/IP was developed in Berkley. Collection of protocols. Transmission Control Protocol and Internet Protocol.

Process Utilities

* ps: process status
* Display some processes attributes
* $ ps
* ps represents a snapshot of the process table
* PID / TTY / TIME / CMD
* Ps with -r option displays a fuller listing that includes the PPID.
* Ps with -u option followed by user-id displays the processes owned by the user-id
* Ps with -e option displays the system processes
* System processes: created by the kernel in the OS.

Accessing the Programmers Manual

* The UNIX and Linux systems include an extensive collection of powerful utility programs, system features, application languages and support libraries
* The UNIX programmer’s manual provides the information needed to employ the exact syntax of a particular option or command format for a utility
* The manualcontains a detailed documentation on the uses and functions of utility programs, application programs and libraries
* The manual also contains information on UNIX system libraries
* The command requires one argument – the name of the utility or the command
* The man command can be used to provide an online manual entry for a utility or a command
* The ‘man -k’ commando can be used to search the manual pages descriptions for keywords
* An important utilityv available on UNIXsystems is theonline help heature provided via the man commaned
  + The syntax is: man [options][-s section] command-list
  + Man -k keyword-list
* Purpose …
* Common options:
  + -k keyword-list: search for summaries of keywords in ‘keyword-list’ in a database and display them
  + -s sec-num: search section number ‘sec-num’ for manual pages and display them
* The output from UNIX is one Reference Manual Page, or ‘man’
* The pages are organized into sections, depending on the topics described and the topics that are applicable to the partivular system
* Most users find the pages they need in section 1 (user commands)
* Software developers mostly use library (3) and system… check other numbers
* Section 1: most useful to us
* 1:user commands
* 2:System calls
* 3: Language Library calls (C, etc)
* 4:Devices and network interfaces
* 5: File formats
* 6:Games and demonstrations
* 7: Environments, tables, and macros for troff
* 7:System maintenance-related commands
* Comprise multipage, specially formatted, descriptive documentation for every command, system call and library call in UNIX
* This format consists of seven general parts:
  + Name
  + Synopsis
  + Description
  + List of files
  + Related information
  + Errors
  + Warning and known bugs
* On the top left corner of the page has the command name with the section it belong to in (), as in LS(1)
* If more than one section of man pages has information o n the same word and we are interested in the man page for a particular section, we can use the -s option
* $ man -s2 read
* This command displays the man page for the ‘read system call’ and **not** the man page for the shell command ‘read’
* Syntax: man[options ] [-s section] command-list
* To exit the man pages simply press ‘q’
* Different versions of UNIX/Linux can have different command options

Getting help on UNIX

* Man: display entries from UNIX online documentation
* Whatis, apropos
* Manual entries organization:
  + 1. Commands
  + 2. System calls
  + 3. Subroutines
  + 4. Special files
  + 5. File format and conventions
  + 6. Games

Other methods of obtaining help

* Short description of any particular UNIX command, we use ‘whatis’ command
* Syntax: whatis keyword keyword …
* Purpose: search the ‘whatis’ database for a short description

Utility Commands

* Exammining System Setups:
  + Whereis:allow you to search along certain prescribed paths to locate utility programs and command, such as shell programs
  + Syntax: whereis [options] filename
  + Purpose: locate the binary, source and man-pages for a command
  + Options: -b and? (there were two)
* To view or display information about your ‘userid’, we use the whoami command
* We use the ‘hostname’ command to find out what host computer we re logged on to

East coast and West coast (AT&T and BSD Linux). We use BSD.

Examining and Managing Files

Printing a file

* The ‘lp’ (AT&T – System-V) or the ‘lpr’ (BSD) command can be used to print a file
* The command requires one argument – the name of the file to be printed
* The “-P” option with the ‘lpr’ command, and the “-d” option with the ‘lp’ command, can be used to specify the printer name

Printing and general utility program

* Syntax:lpr [options] filename
* Purpuse: send files to the printer
* Output: files sent to the printer queue as print jobs
* Common options:
  + -P printer: send output to the named printer
  + # copies: produce the number of copies indicated for each named file

Pipe(|): man ls | lpr -P hp1

* Two processes. They communicate via pipe (|). When the ls process is created we have default input and output. The | redirects the output of the first process to the input of the second proces
* This command prints the man pages describing the ‘ls’ command at the printer named hp1
* ‘lpr’ command is used on a BSD compliant system sycg as FreeBSD or SunOS
* Under SVR4 we use the ‘lp’ command instead of ‘lpr’ command

General utility commands

cal ( Calendar)

* $ cal [month] [year]
* Purpose: to display the calendar of the month or year

write

* Used to send a message to another user who is currently logged on
* Syntax: write username [terminal]
* Purpose: to write on the terminal screen of the user with log-in name ‘username’
* Output: message on another user’s console window
* The prerequisite fot executing the write command is execution of the ‘mesg y’ (check it)

Mesg (messages)

* Enables or disables real-time one-way messages and chat-requests from other users with the ‘write’ or ‘talk’ command
* We can turn off the permission by executig the ‘mesg -n’ command
* Example: $ write bobk ttyc2
* Bobk computer: ‘mesg y’
* pts = pseudo terminal. ttyc2 ? ttyp0?

Nicknames for commands

* Alias or alternate name can be used for commands that are hard to remember
* The ‘alias’ command can be used to provide a list of all the current aliases
* The command “unalias alias-name” can be used to remove an alias
* The alias command is used to create psedonyms (nicknames) for the command and custoimize them
* Syntax: alias [name [=string]…] Bourne, Korn and bash
* Alias [name [string]] C shell
* Output: pseudonyms that can be used for commands
* The C shell allows you to create aliases from the command line, but the others shells
* Pro: really long commands
* Con: No man pages, but more importantly it is time consuming (it has to process the command). If you use too many aliases the system can slow down
* Alias alias name utility name csh and tch
* Alias alias name=utility name bash andd ksh
* Command aliases can be placed in the:
  + .profile file for SRV4
  + .login file for BSD
* But they are typically placed in the .bashrc (bash) and the .cshrc file in the for C shell
* The .profile and .login files executes when we log on and the .cshrc or .bashrc file executes every time we start a C or Bash shell
* When we use the ‘alias’ command without any argument, it lists all the aliases currently
* bashrc means bash run command
* Environment is different for dirrent shells. One environemnt when you log in the pcomputer and one environment when you log into a shell