

# Linux Commands Review



**CS 60**

**Discussion – I**

**Sudipto Das**

**12<sup>th</sup> January 2007**

# Linux Commands



- **cd: Change Directory**
  - Usage: *cd [dir]*
- **exit: Allows you to exit from a program, shell or log you out of a UNIX network**
  - Usage: *exit*
- **echo: Send (echo) the input *string* to standard output**
  - Usage: *echo [string]*
- **logout: Exit a login shell**
  - Usage: *logout*
- **pwd: Print the full pathname of the current working directory**

# Linux Commands

## *cat*



- Usage: *cat [options] [files]*
- Read (con**cat**enate) one or more *files* and print them on standard output.
- The `>` operator can be used to combine several files into a new file, or `>>` to append files to an existing file.
- When appending to an existing file, use Ctrl-D, the end-of-file symbol, to end the session.

# Linux Commands

## *chmod*



- Usage:

*chmod [options] mode files*

*chmod [options] --reference=filename files*

- Change the access *mode* (permissions) of one or more *files*
- Only the owner of a file or a privileged user may change the mode
- Who: u=user g=Group o=Other a=All
- Opcodes: + Add, - Remove, = Assign Permission
- r: Read, w: Write, x: Execute
- Alternatively by using three-digit octal numbers to specify permissions

# Linux Commands

## *chmod*



- **Examples**
- Add execute-by-user permission to *file*:
  - *chmod u+x file*
- Either of the following will assign read/write/execute permission by owner (7), read/execute permission by group (5), and execute-only permission by others (1) to *file*:
  - *chmod 751 file*
  - *chmod u=rwx,g=rx,o=x file*
- Any one of the following will assign read-only permission to *file* for everyone:
  - *chmod =r file*
  - *chmod 444 file*
  - *chmod a-wx,a+r file*

# Linux Commands

***cp***



- Usage:
  - *cp [options] file1 file2*
  - *cp [options] files directory*
- Copy *file1* to *file2*, or copy one or more *files* to the same names under *directory*
- If the destination is an existing file, the file is overwritten; if the destination is an existing directory, the file is copied into the directory (the directory is *not* overwritten)

# Linux Commands

## *ls*



- *Usage: ls [options] [names]*
- List contents of directories. If no *names* are given, list the files in the current directory
- Most important options are:
  - -1: display in 1 column format
  - -a: list all files, including hidden files
  - -l: long listing format, provides details
  - -h: print the file sizes in human readable format (1K, 4M)
  - -R: recursive listing

# Linux Commands

## *man*



- Usage: *man [options] [section] [title]*
- Display information from the reference manuals
- **man** locates and prints the manual named *title* from the designated reference *section*
- Traditionally, man pages are divided into 9 sections
  - **1** Executable programs or shell commands.
  - **2** System calls (functions provided by the kernel).
  - **3** Library calls (functions within system libraries).
  - **4** Special files (usually found in */dev*).
  - **5** File formats and conventions (e.g., */etc/passwd*).
  - **6** Games.
  - **7** Macro packages and conventions.
  - **8** System administration commands (usually only for a privileged user).
  - **9** Kernel routines (nonstandard).

# Linux Commands

## *mkdir*



- Usage: *mkdir [options] directories*
- Create one or more *directories*
- One must have write permission in the parent directory in order to create a directory
- *Examples:*
  - **mkdir dir1 dir2 dir3**
  - **mkdir -m 444 personal**
  - **mkdir -p work/junk/questions**

# Linux Commands

## *more*



- Usage: *more [options] [files]*
- Display the named *files* on a terminal, one screen at a time
- Page through *file* in "clear" mode, and display prompts:
  - **more -cd** *file*

# Linux Commands

## *mv*



- Usage: *mv [option] sources target*
- Move or rename files and directories
- The source (first column) and target (second column) determine the result (third column)

Source	Target	Result
File	Target non-existent	Rename the source file
File	Existing File	Overwrite existing file with source file
Directory	<i>name</i> (nonexistent)	Rename directory to <i>name</i>
Directory	Existing directory	Move directory to be a subdirectory of existing directory
One or more files	Existing directory	Move files to directory

# Linux Commands

## *rm*



- Usage: *rm [options] files*
- Delete one or more *files*
- To remove a file, you must have write permission in the directory that contains the file, but you need not have permission on the file itself
- **rm** is often aliased to **rm -i**, especially for the root user, to protect against inadvertently deleting files

# Linux Commands

## *rm*dir



- Usage: *rm*dir [*options*] *directories*
- Delete the named *directories* (not the contents)
- *directories* are deleted from the parent directory and must be empty
- A useful option is
  - **-p, --parents** Remove *directories* and any intervening parent directories that become empty as a result. Useful for removing subdirectory trees

# Linux Commands

*w*



- Usage: *w [options] [user]*
- Print summaries of system usage, currently logged-in users, and what those users are doing
- **w** is essentially a combination of **uptime**, **who**, and **ps -a**.
- Display output for one user by specifying *user*.

# Linux Commands

## *who*



- Usage:
  - `who [options]`
  - `who am i`
- Show who is logged into the system
- With no options, list the names of users currently logged in, their terminal, the time they have been logged in, and the name of the host from which they have logged in

# Something beyond Commands



- **Linux Wildcards**
  - \*: Matches one or more characters
  - ?: Matches a single character
  - [abcde]: Matches exactly one character listed
- **Shortcuts in Linux**
- **Linux File system – An overview**



<http://www.oreillyn.com/linux/cmd/>