



澳門理工學院  
Instituto Politécnico de Macau  
Macao Polytechnic Institute

# COMP 225

## Network and System Administration

Notes #3: Utility

---

K. L. Eddie Law, PhD

Macao Polytechnic Institute  
School of Applied Sciences  
Academic Year 2020-2021, 2<sup>nd</sup> Semester

### Some Commands Briefly Discussed

- |                            |                         |                      |                      |
|----------------------------|-------------------------|----------------------|----------------------|
| • <code>uname -a</code>    | • <code>id</code>       | • <code>stat</code>  | • <code>mkdir</code> |
| • <code>systemctl</code>   | • <code>groups</code>   | • <code>head</code>  | • <code>rmdir</code> |
| • <code>ps aux</code>      | • <code>hostname</code> | • <code>tail</code>  | • <code>cd</code>    |
| • <code>ls -la</code>      | • <code>date</code>     | • <code>clear</code> | • <code>rm</code>    |
| • <code>cat</code>         | • <code>cal</code>      | • <code>echo</code>  | • <code>pwd</code>   |
| • <code>less   more</code> | • <code>uptime</code>   | • <code>chmod</code> | • <code>touch</code> |
| • <code>whoami</code>      | • <code>wc</code>       | • <code>chown</code> | • <code>which</code> |
| • <code>logname</code>     | • <code>file</code>     | • <code>chgrp</code> |                      |

## Some Reserved Keywords – Scripting

- |        |        |         |       |
|--------|--------|---------|-------|
| • if   | • case | • while | • for |
| • elif | • esac | • until | • !   |
| • else |        | • do    | • {   |
| • then |        | • done  | • }   |
| • fi   |        |         |       |

## Objectives

- Learn some basic UNIX/Linux utilities
- The basic of grep command
- Use the dd utility to copy and convert files
- Monitor hard disk usage
- Use system status utilities
- Monitor and manage processes
- Check the spelling of text in a document
- Use the cmp command to compare the contents of two files
- Format text to create and use a man page

# UNIX/Linux Utilities

- Utilities for
  - Creating and managing files
  - Producing reports
  - Monitoring and maintaining the system
  - Executing programs
  - Recovering from a range of errors
- New utilities are updated/edited/added daily to make Linux run more efficiently
  - Of course, some are deprecated or removed

Eddie Law

## UNIX/Linux Utilities (cont'd)

- Eight major areas:
  - File processing
  - System status
  - Networking
  - Communications
  - Security
  - Programming
  - Source code management
  - Miscellaneous

Eddie Law



# File Processing Utilities

Command	Brief Description of Function
<b>awk</b>	Processes files
<b>cat</b>	Displays files (and is used with other tools to concatenate files)
<b>cmp</b>	Compares two files
<b>comm</b>	Compares sorted files, and show differences
<b>cp</b>	Copies files
<b>cpio</b>	Copies and backups files in a archive
<b>cut</b>	Selects characters or fields from input lines
<b>dd</b>	Copies and converts input records
<b>diff</b>	Compares two text files, and shows differences
<b>file</b>	Displays the file type
<b>find</b>	Finds files within file tree
<b>fmt</b>	Formats text files for displaying
<b>grep</b>	Matches patterns in files, for line filtering, word search, etc.
<b>gzip</b>	Compresses or decompresses files
<b>ispell</b>	Checks one or more files for spelling errors
<b>ln</b>	Creates a link to a file

Command	Brief Description of Function
<b>lpr</b>	Sends a file to a printer or print device
<b>man</b>	Displays documentation for commands
<b>mkfs</b>	Builds a Linux/UNIX file system
<b>mount</b>	Mounts file systems or devices
<b>od</b>	Formats and displays data from a file in octal, hexadecimal, or ASCII format
<b>paste</b>	Concatenates file horizontally
<b>pr</b>	Formats text files for printing, and displays them
<b>sed</b>	Edits streams (non-interactive)
<b>sort</b>	Sorts or merges files
<b>tail</b>	Displays the last lines of files (default: last 10 lines)
<b>tar</b>	Copies and backs up files to a tape archive
<b>tr</b>	Translates or deletes characters from standard input and writes results to standard output
<b>uniq</b>	Displays unique lines, or reports repeated lines
<b>whereis</b>	Locates information about a specific file

Eddie Law

## The man Command

- An interface to the system reference manuals
- There might be different “section” page numbers for each command
- `$ man stat`
- And try `$ info stat`

Page Number	Description
1	User commands
2	System calls
3	C library functions
4	Devices and special files
5	File formats and conventions
6	Games
7	Miscellaneous
8	System admin tools and daemons

# The ln Command\*

- Soft link
  - `$ ln -s original newlink`
- What is the inode number?
  - `$ ls -li`
- Hard link
  - `ln original newlink`
- What are the inode and link numbers?
  - `$ ls -li`

# The tar Command

- tar stands for tape archiver
- Used as a powerful backup and restore utility
- Most Linux files are downloaded as .tar files
- E.g., extraction of a tar file

```
tar -xvf file.tar
```

  - x extracts files
  - v verbose
  - f filename
- More on it later...

## Two Files

- `$ cat file1`

Brian has a dog

Byron has 2 dogs

Ellen has a cat

Elden has a snake

Louise has two dogs

Ruby has two puppies and one cat

- `$ cat file2`

Brian has a dog

Byron has two dogs

Ellen has a cat

Elden has a snake

## Comparing Files

- Use the `cmp` utility to compare the contents of two files, and report the first difference between them
- The `cmp` command displays the position and line number of this difference
  - `$ cmp file1 file2`
- If there are no differences, the `cmp` command displays nothing



# The uniq Command

- Runs on a single file to searching for consecutive duplicate lines
- `uniq` removes any such duplicates, does not overwrite the file, but outputs a file without the duplicates, can simply redirect to a new file  
`$ uniq file.txt > filewithoutduplicates.txt`
- Since only compares adjacent lines, cannot find duplicate lines that do not appear one after the other

Eddie Law

## uniq (cont'd)

- Options:
  - Output file without duplicates, or
  - `(-D)` output the found duplicates, or
  - `(-c)` count the number of duplicates
  - Can ignore `(-i)` case or
  - `(-s #)` skip over characters

Eddie Law 14

# The diff Command\*

- Compares the differences between two files, in general
- For each line in the first file but not in second, output preceded by a '<'
- For every line in second file but not in first, output preceded by a '>'
- For each line or group of lines that differ, a summary is provided indicating how the two differ
  - If file 1 had a line, not in file 2, then the line had to be deleted to match file 2
  - If file 2 had a line that not in file 1, suggested that the line had to be added
  - If two corresponding lines between the two files did not match, suggested to change the line
  - Indicated by letters 'a' for added, 'd' for deleted, and 'c' for changed; if 3a5,6 ⇒ at line 3 of the first file, we had to add lines 5 to 6 of the second file

Eddie Law

## The diff Command (cont'd)

- If a file does not exist, it responds with an error
- If first filename is a directory, diff finds file in the directory whose name matches that of the second file
- If diff is provided with two directories, it compares all pairs of files who share the same names in both directories
- Option -i ignores upper and lower case letters; -w, -B, -E, ignore all white space, blank lines, tab expansions, respectively
- To operate on more than two files, add option --from-file=
  - If comparing file1 to all of file2, file3, file4, and file5, should use  
`$ diff --from-file=file1 file2 file3 file4 file5`
  - Output of the comparisons is separated by --- to indicate that the next file is being compared

Eddie Law



# The diff and patch Commands

- Given two file1 and file2
  - `$ diff file1 file2 > patchfile`
  - `$ path file1 < patchfile`
- Now both file1 and file2 are identical

# grep (Global Regular Expression Print)\*

- To search for words “cat” in the file “file1”

```
$ grep cat file1
$ grep cat < file1
$ cat file1 | grep cat
```

Represents a character

Beginning of a line, or not the letter after this symbol

End of a line

Zero or more of a pattern just described

- Some **metacharacters**: . ^ \$ \* etc.

- Search the metacharacter symbol, use an escape “\”
- ? have meaning in shell, not for grep

- Character classes: delineated with [ and ] symbols

- [ai] matches an a or an i, both are in lowercase
- [a-z0-9] matches the lowercase letters a through z, and 0 to 9

## grep (cont'd)

- Regular expression can be complicated
- Options
  - i ignores cases
  - v complement lines instead of selected lines, -v “^\$” removes blank lines
  - c counts number of **lines** matching
  - l shows names of files that contains the pattern, check out all files with “\*”
  - w shows results with exact word matching, no partial matching
  - b prints the byte offsets from the beginning of the file
  - n prints the line offsets from the beginning of the file
  - o only characters that match
  - E use extended regular expression (equal to use egrep)

Eddie Law

## The find Command\*

- A powerful tool, the format is
  - \$ find [directory] [options]
  - where directory is the starting point in the file system for the search to begin
- Let consider the simple case here
- E.g., search for files with .conf extensions under /etc directory
  - \$ find /etc -name “\*.conf” -print
  - Many new implementations don’t need “-print” anymore
- Option: -type [b|c|d|p|f|l|s]
  - where b (block), c (character), d (directory), p (pipe), f (regular file), l (symbolic link), s (socket)
- Test it out: \$ find ~ -type f -exec wc -l {} \;

Eddie Law

# System Status Utilities

Command	Brief Description of Function
date	Sets and displays date and time
df	Displays the amount of free space remaining on disk
du	Summarizes file space usage
file	Determines file type (e.g., script, executable, ASCII, etc.)
free	Displays amount of free and used memory in the system
kill	Terminates a running process
pgrep	Returns the process IDs that match the process name
pkill, killall	Kills a process, given its name
ps	Displays process status by process identification number and name
pstree	Visualizes processes, and displays them in tree format
renice	Changes the nice value of an already running process
sleep	Suspends process execution for a specified time
top	Dynamically displays the status of processes in real time, focusing on those processes that are using the most CPU resources
uname	Shows information about the operating system
vmstat	Shows information about virtual memory use
w	Displays detailed information about the users who are logged in
who	Displays brief information about the users who are logged in

21

## The finger and sleep command

- The **finger** command was used to find out information about users
  - **Unsafe**, usually not installed by default
  - `$ finger username`
    - Displays information about the user including username, full name, home directory, last login time, shell, etc.
- The **sleep** command:
  - Suspends the execution of the process in number of milliseconds



# Network Utilities

Command	Brief Description of Function
dig	Performs DNS lookups and displays the answers from name servers
ftp	Transfers files over a network (not safe)
ifconfig	Set up a wired network interface (old command)
ip	Set up network interface (new command)
netstat	Shows network connection information (old)
nfsstat	Shows statistics for Network File System (NFS) activity
nmap	Checks the opened port on the server
nslookup	Queries Internet domain name server (DNS)
ping	Polls another network station (using TCP/IP)
rcp	Remotely copies a file from a network computer
rlogin	Logs in to a remote computer (not safe)
route	Displays routing table information
rsh	Executes commands on a remote computer
scp	Secure transfers files between a local host and a remote host or between two remote hosts
sftp	Transfers files securely over a network connection
ssh	Enables secure connection to SSH server on a remote machine

Eddie Law 23

## Some Network Utility Commands

- **dig** – gets IP address from a domain name address, replacing the nslookup command  
\$ dig www.yahoo.com
- **ifconfig** – sets up a wired network interface card (obsolete in Debian/Ubuntu)
- **ip** – sets up network interfaces, can be used to troubleshoot networking  
\$ ip link  
\$ ip addr

Eddie Law 24

## Network Utility Commands (cont'd)

- `netplan` – backend-agnostic network configuration tool in YAML
- `netstat` – shows network connection information (not popular now)
- `ping` – establishes connectivity to a remote device (try it)
- `route` – displays routing table information (being replaced by `ip route`)
  - Can be Installed with `$ sudo apt install net-tools`

## Other Network Utility Commands

- Most of them require installations
  1. Overall bandwidth - `nload`, `bmon`, `slurm`, `bwm-ng`, `cbm`, `speedometer`, `netload`
  2. Overall bandwidth (batch style output) - `vnstat`, `ifstat`, `dstat`, `collectl`
  3. Bandwidth per socket connection - `iftop`, `iptraf`, `tcptrack`, `pktstat`, `netwatch`, `trafshow`
  4. Bandwidth per process - `nethogs`
- We will come back more on the IP networking later!

# Communications Utilities (Insecure)

- wall – sends a message to all logged-in users.
- mesg n – denies any real-time messages
- mesg y – accepts any real-time messages
- write – sends a message to a user
- mail – sends e-mail
- talk – allows users to simultaneously 'chat' with other logged in users

Command	Brief Description of Function
mail	Sends email messages
mesg	Denies (mesg n) or accept (mesg y) messages
talk	Lets users simultaneously type messages to each other (unsafe)
wall	Sends a message to all logged in users (who have permissions set to receive messages)
write	Sends a message to another use

Eddie Law 27

# Security Utilities

- Obsolete: ipchains
- Up-and-coming but not yet ready: bpfilter

Table 8-5 Security utilities

Command	Brief Description of Function
<i>chgrp</i>	Changes the group associated with a file or the file's group ownership
<i>chmod</i>	Changes the access permissions of a file or directory
<i>chown</i>	Changes the owner of a file

Table 8-5 Security utilities (continued)

Command	Brief Description of Function
<i>ipchains</i>	Manages a firewall and packet filtering (do not use if you are using <i>iptables</i> instead)
<i>iptables</i>	Manages a firewall and packet filtering (do not use if you are using <i>ipchains</i> instead)
<i>passwd</i>	Changes a password

More on it later →

Eddie Law



# Programming and Source Code Management Utilities

- Linux is written in C programming language
- Get compilers with `$ sudo apt install build-essential`

Table 8-6 Programming utilities

Command	Brief Description of Function
<i>configure</i>	Configures program source code automatically
<i>g++</i>	Compiles a C++ program
<i>gcc</i>	Compiles a C program
<i>make</i>	Maintains program source code
<i>patch</i>	Updates source code

Table 8-7 Source code management utilities (fyi only, obsolete)

Command	Brief Description of Function
<i>ci</i>	Creates changes in Revision Control Systems (RCS)
<i>co</i>	Retrieves an unencoded revision of an RCS file
<i>cvs</i>	Manages concurrent access to files in a hierarchy
<i>rcs</i>	Creates or changes the attributes of an RCS file
<i>rlog</i>	Prints a summary of the history of an RCS file

Eddie Law

## Revisit Permission Management

- Not the encryption security
- Recalling the commands ...
- `chmod` ...
  - Changes the access permissions of a file or directory
- `chown [newOwner] [file]`
  - Changes the owner of a file or directory
- `chgrp [newGroup] [file]`
  - Changes the default group associated with a file

Eddie Law

# POSIX File Permission Management

- Design issues with typical Linux file permission system
  - Only one user owner and one group owner
  - Different to change with the inheritance nature
- Try

```
$ getfacl file1
```
- Example:

```
$ setfacl -m g:root:rwX file1
$ getfacl file1
```
- Just let you get a feel of the concept of ACL (Access Control List) only

Eddie Law 31

## Miscellaneous Utilities

Table 8-8 Miscellaneous utilities

Command	Brief Description of Function
<i>at</i>	Executes a command or script at a specified time
<i>atq</i>	Shows the jobs (commands or scripts) already scheduled to run
<i>atrm</i>	Enables you to remove a job (command or script) that is scheduled to run
<i>batch</i>	Runs a command or script, and is really a subset of the <i>at</i> command that takes you to the <i>at&gt;</i> prompt, if you type only <i>batch</i> (in Fedora and Red Hat Enterprise Linux, a command or script is run when the system load is at an acceptable level)
<i>cal</i>	Displays a calendar for a month or year
<i>cd</i>	Changes to a directory
<i>crontab</i>	Schedules a command to run at a preset time
<i>expr</i>	Evaluates expressions (used for arithmetic and string manipulations)
<i>fsck</i>	Checks and fixes problems on a file system (repairs damage)
<i>printenv</i>	Prints environment variables
<i>tee</i>	Clones output stream to one or more files
<i>tr</i>	Replaces specified characters (a translation filter)
<i>tty</i>	Displays terminal path name
<i>xargs</i>	Converts standard output of one command into arguments for another

Eddie Law

# Using the dd Command

- Copy a file and change the format of the destination file
- Options to handle copies when other methods are inappropriate such as when the format of the destination file needs to be altered
  - E.g., ASCII to EBCDIC, uppercase to lowercase, etc.
- An advantage to using the dd command over cp is that all users, not just the administrator, can copy files to and from the drive without mounting it
- Beware: no ways to restore if overwritten

Eddie Law

## Using the dd Command (cont'd)

- Be careful in using dd command
  - # dd if=/dev/sda of=/dev/sdb (backup the entire harddisk)
  - # dd if=/dev/hda1 of=/dev/partition.img (backup a partition)
  - # dd if=/dev/hda of=/dev/hdimage.img (create a hard disk image)
  - # dd if=hdimage.img of=/dev/hdb (restore using a hard disk image)
  - # dd if=/dev/cdrom of=cd.iso bs=2048 (create CDROM backup, 2048 block size)
  - # dd if=ubuntu-20.10.iso of=/dev/sdb bs=4M status=progress oflag=sync
- Commonly used options:
  - if=input\_file
  - of=output\_file
  - conv=ascii (converts destination format to ASCII)
  - conv=lowercase (converts uppercase to lower)

Eddie Law



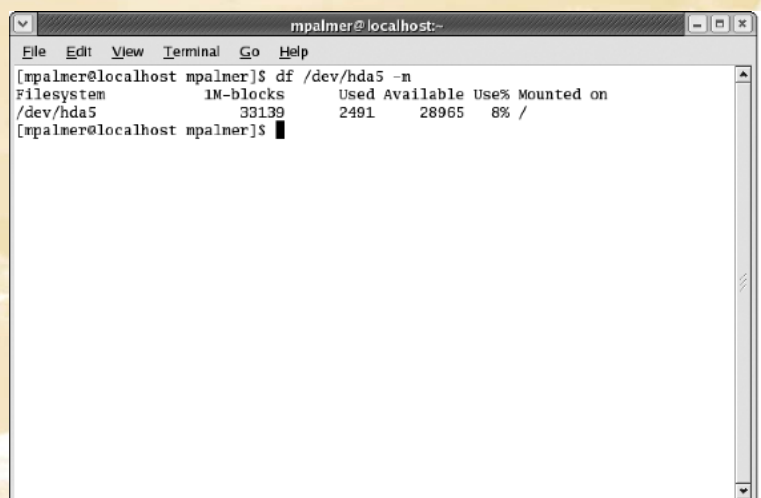
# Checking Hard Disk Usage

- To maintain adequate hard disk free space, use these strategies:
  - Be vigilant against running dangerously low on free space by using the **df** command
  - Watch for conspicuous consumption using the **du** command
  - Follow a routine schedule for “garbage” collection and removal by using the **find** and **rm** commands

Eddie Law

## Using the df (disk free) Utility

- The df utility reports on the status of 1024-byte blocks that are allocated, used, and available and the mount point
- Options:
  - h human readable form
  - k sizes in kilobytes



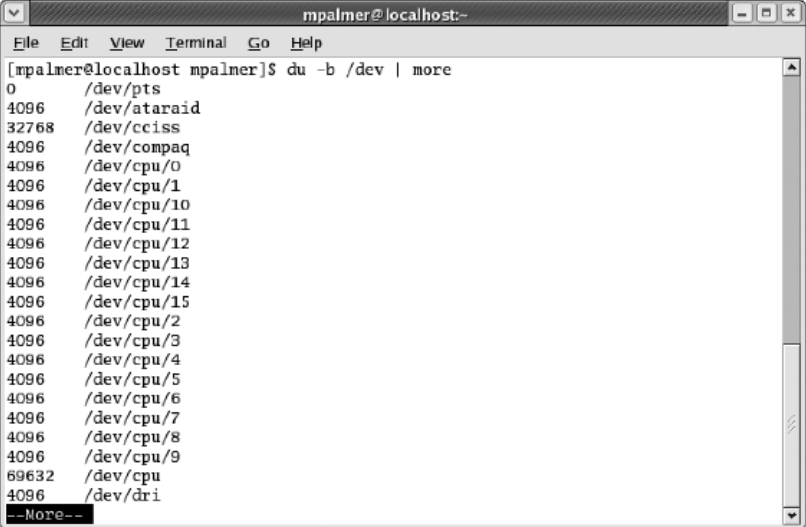
```
mpalmer@localhost:~$ df /dev/hda5 -h
Filesystem      1M-blocks      Used Available Use% Mounted on
/dev/hda5        33139        2491    28965    8% /
```

Figure 8-2 Viewing information for one file system in megabytes

Eddie Law

# Using the du (disk usage) Utility

- The du utility summarizes disk usage, expressed in 1024-byte blocks (default) or by the number of bytes (-b option)
- Remark: 512-byte block for POSIXLY-CORRECT
- Options:
  - a displays info for files/dirs
  - c creates an ending total
  - b displays in bytes
  - h human readable



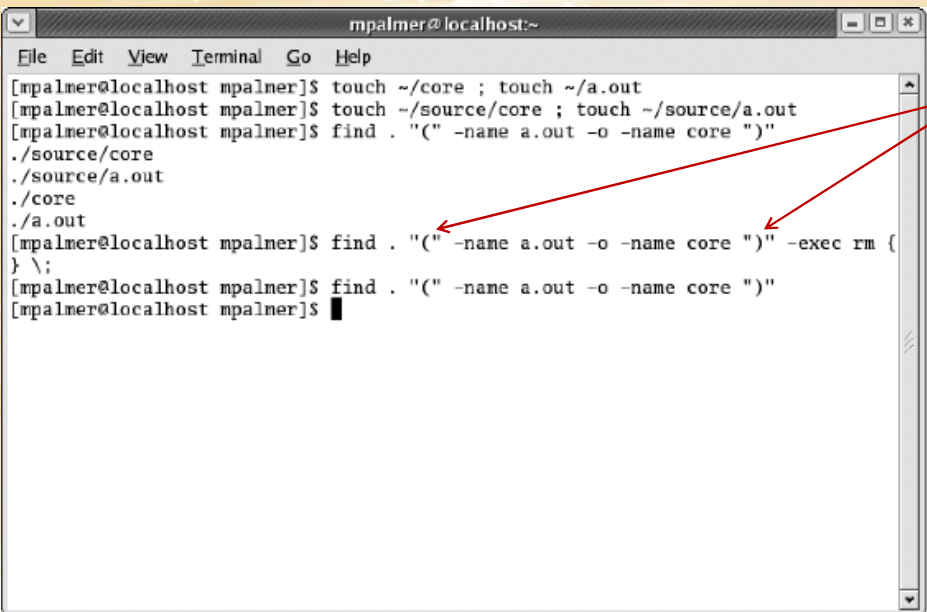
```
mpalmer@localhost:~$ du -b /dev | more
0      /dev/pts
4096   /dev/ataraid
32768  /dev/cciss
4096   /dev/compaq
4096   /dev/cpu/0
4096   /dev/cpu/1
4096   /dev/cpu/10
4096   /dev/cpu/11
4096   /dev/cpu/12
4096   /dev/cpu/13
4096   /dev/cpu/14
4096   /dev/cpu/15
4096   /dev/cpu/2
4096   /dev/cpu/3
4096   /dev/cpu/4
4096   /dev/cpu/5
4096   /dev/cpu/6
4096   /dev/cpu/7
4096   /dev/cpu/8
4096   /dev/cpu/9
69632  /dev/cpu
4096   /dev/dri
--More--
```

Figure 8-3 Viewing du information for the /dev directory  
Eddie Law

# Removing Garbage Files

- Garbage files are temporary files that lose their usefulness after several days
- Two examples of garbage files are core files (named core) and a.out files
- Use the find command to assist you in locating these files and the rm command to remove them
  - On the next slide, find is used to remove garbage files, and
  - The -exec rm {} \; option tells Linux to rm all files found {} by the command

## Removing Garbage Files (cont'd)



A terminal window titled 'mpalmer@localhost:~' showing the following commands and output:

```
[mpalmer@localhost mpalmer]$ touch ~/core ; touch ~/a.out
[mpalmer@localhost mpalmer]$ touch ~/source/core ; touch ~/source/a.out
[mpalmer@localhost mpalmer]$ find . "(" -name a.out -o -name core ")"
./source/core
./source/a.out
./core
./a.out
[mpalmer@localhost mpalmer]$ find . "(" -name a.out -o -name core ")" -exec rm {
} \;
[mpalmer@localhost mpalmer]$ find . "(" -name a.out -o -name core ")"
[mpalmer@localhost mpalmer]$
```

Red arrows point from the text 'May not be needed, depends on version' to the first and third instances of the `find` command in the terminal output.

May not be needed, depends on version

-o OR  
-a AND

Figure 8-10 Using the *find* command to delete garbage files die Law

## Using System Status Utilities

- System status commands reflect the system's performance
- System engineers primarily use the data related to system status
- Good to know how to obtain and store relevant information to send to system administrator and tune-up specialists

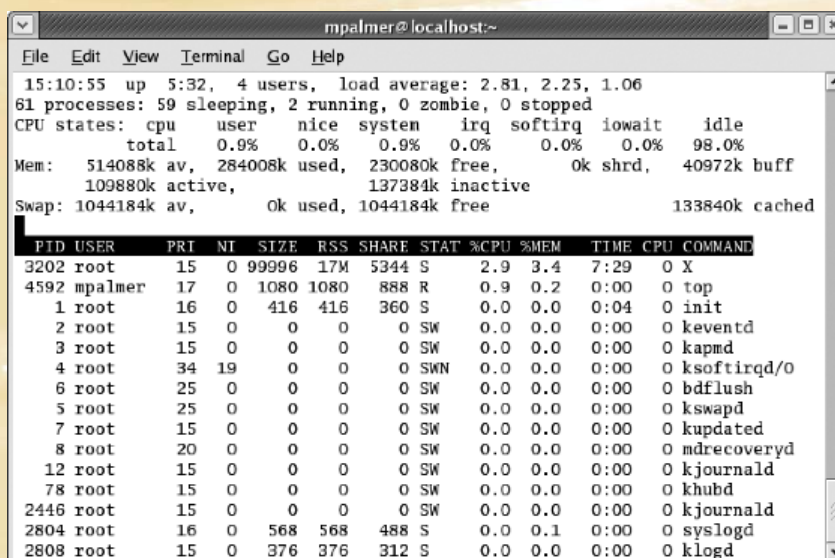


# Using the top Command

- One of the most effective utilities for auditing system performance is the top command
- The top command displays a listing of the most CPU-intensive tasks in real time
- Updates every five seconds by default
- Press “q” to quit it

Eddie Law

## Using the top Command (cont'd)



The screenshot shows a terminal window titled 'mpalmer@localhost:~'. The output of the 'top' command is displayed, showing system statistics and a list of processes. The processes are sorted by CPU usage, with 'X' (idle) at the top, followed by 'top', 'init', 'keventd', 'kapmd', 'ksoftirqd/0', 'bdfldush', 'kswapd', 'kupdated', 'mdrecoveryd', 'kjournald', 'khubb', 'kjournald', 'syslogd', and 'klogd'.

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	CPU	COMMAND
3202	root	15	0	99996	17M	5344	S	2.9	3.4	7:29	0	X
4592	mpalmer	17	0	1080	1080	888	R	0.9	0.2	0:00	0	top
1	root	16	0	416	416	360	S	0.0	0.0	0:04	0	init
2	root	15	0	0	0	0	SW	0.0	0.0	0:00	0	keventd
3	root	15	0	0	0	0	SW	0.0	0.0	0:00	0	kapmd
4	root	34	19	0	0	0	SWN	0.0	0.0	0:00	0	ksoftirqd/0
6	root	25	0	0	0	0	SW	0.0	0.0	0:00	0	bdfldush
5	root	25	0	0	0	0	SW	0.0	0.0	0:00	0	kswapd
7	root	15	0	0	0	0	SW	0.0	0.0	0:00	0	kupdated
8	root	20	0	0	0	0	SW	0.0	0.0	0:00	0	mdrecoveryd
12	root	15	0	0	0	0	SW	0.0	0.0	0:00	0	kjournald
78	root	15	0	0	0	0	SW	0.0	0.0	0:00	0	khubb
2446	root	15	0	0	0	0	SW	0.0	0.0	0:00	0	kjournald
2804	root	16	0	568	568	488	S	0.0	0.1	0:00	0	syslogd
2808	root	15	0	376	376	312	S	0.0	0.0	0:00	0	klogd

The top utility  
run without  
any options  
specified

Figure 8-11 Sample top display

Eddie Law

# Using the uptime Command

- Uptime tells you how long a system has been running since the last time it was booted
- Displays current time, how long the system has been up, number of users on the system, and the load average for 1, 5, and 15 minutes

Eddie Law

# Using the free Command (vmstat)

- The free utility displays the amount of free and used memory in the system
- Options
  - b bytes
  - m megabytes
  - g gigabytes
  - t totals



```
mpalmer@localhost:~  
[mpalmer@localhost mpalmer]$ free  
              total        used        free      shared    buffers     cached  
Mem:           514088       383196       130892           0        65888       128112  
-/+ buffers/cache: 189196       324892  
Swap:          1044184           0       1044184  
[mpalmer@localhost mpalmer]$
```

Figure 8-4 Using the free command to monitor memory and swap usage

Eddie Law

## Forwarding top and free Output

- When problems arise with performance, may need to forward top and free output to support person
- Use redirection (>) to store outputs in files  

```
$ top n 3 > topdata
```

Eddie Law

## Managing Processes

- A process is identified through a unique number called a process id (PID)
- Unix/Linux offer utilities to run, monitor, and kill processes using PIDs

Eddie Law



# Running Processes in the Background

- Run a process in the background while working with another program in the foreground
- To run a program in the background, append the & character to end of the startup command, e.g.,

`$ top &`

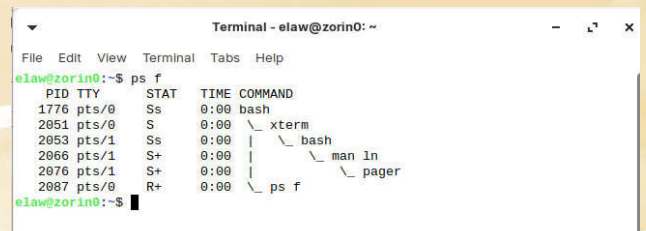
Eddie Law

# Monitoring Processes

- The `ps` command with the `-A` option shows a list of all system processes currently running
- Command `$ ps -aux` displays all processes running in the system
- Command `$ ps f` displays processes in parent-child relationships



```
Terminal - elaw@zorin0: ~
File Edit View Terminal Tabs Help
PID TTY      TIME CMD
1 ?        00:00:00 systemd
2 ?        00:00:00 kthreadd
3 ?        00:00:00 rcu_gp
4 ?        00:00:00 rcu_par_gp
6 ?        00:00:00 kworker/0:0H-kblockd
9 ?        00:00:00 mm_percpu_wq
10 ?       00:00:00 ksoftirqd/0
11 ?       00:00:00 rcu_sched
12 ?       00:00:00 migration/0
13 ?       00:00:00 idle_inject/0
14 ?       00:00:00 cpuhp/0
15 ?       00:00:00 kdevtmpfs
16 ?       00:00:00 netns
17 ?       00:00:00 rcu_tasks_kthre
18 ?       00:00:00 rcu_tasks_rude_
19 ?       00:00:00 rcu_tasks_trace
20 ?       00:00:00 kauditd
21 ?       00:00:00 khungtaskd
22 ?       00:00:00 oom_reaper
23 ?       00:00:00 writeback
24 ?       00:00:00 kcompactd0
25 ?       00:00:00 ksmd
~More~
```



```
Terminal - elaw@zorin0: ~
File Edit View Terminal Tabs Help
elaw@zorin0:~$ ps f
PID TTY      STAT TIME COMMAND
1776 pts/0    Ss   0:00 bash
2051 pts/0    S    0:00 \_ xterm
2053 pts/1    Ss   0:00 | \_ bash
2066 pts/1    S+   0:00 | \_ man ln
2076 pts/1    S+   0:00 | \_ pager
2087 pts/0    R+   0:00 \_ ps f
elaw@zorin0:~$
```

Eddie Law

# Killing Processes

- Administrator with root privileges can kill any user's processes
- User can kill own processes
- Use the `kill` command with the pid of the process
- Use `$ kill -9` (the sure kill) to stop a process that doesn't respond to an initial `kill` command
- If I have started executing a program (p1) that is running infinitely, I may kill that process with the following steps:

```
$ ps
```

(Assume the PID number of process p1 is 608)

```
$ kill 608
```

Eddie Law

# Checking the Spelling of a Document

- `ispell` scans a document, displays errors on the screen and suggests alternative spellings

```
Install ispell with  
$ sudo apt install ispell
```

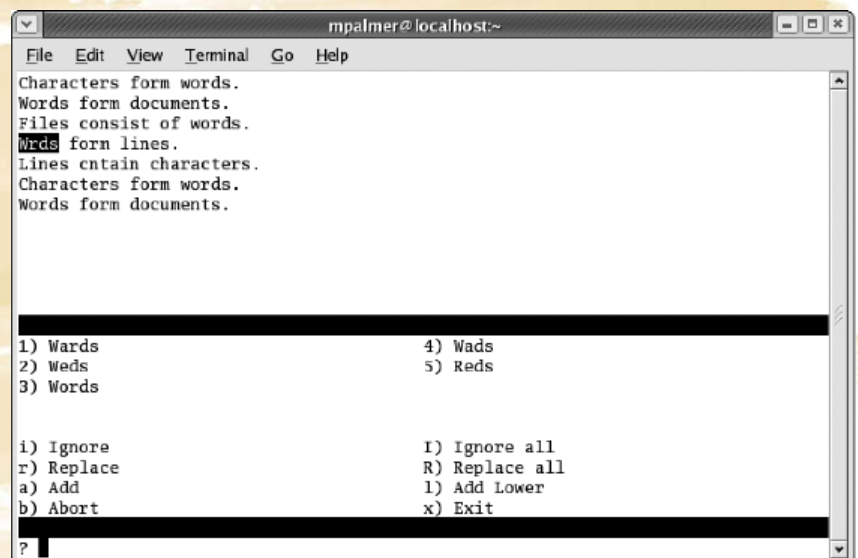


Figure 8-6 Checking the spelling in a document with *ispell*

Eddie Law

## Summary

- UNIX/Linux utilities are classified into eight major functional areas
- Utility programs are “commands”: executed by entering names on the command line
- `grep` and regular expression
- `dd` command options allow it to handle copies when other copying methods fail
- `df` checks and reports on free disk space
- `du` checks for disk usage
- Use `find` to retrieve temporary files and use `rm` to remove them

Eddie Law

## Summary (cont'd)

- Run a program in the background by appending `&` to the end of a command
- `top` and `free` provide views of the “internals” of the system that can be redirected to a file for system tune-up
- `ps` displays all running processes
- `kill` terminates a specific process
- `ispell` scans for spelling errors (requires installation)

Eddie Law



