澳 門 理 工 學 院
Instituto Politécnico de Macau
Macao Polytechnic Institute

School of Applied Sciences (B.Sc. In Computing)

COMP 225: Network and System Administration Notes #3: Utility

K. L. Eddie Law, PhD
Associate Professor
Computing, MPI

Academic Year 2nd Semester, 2019-2020

Topics

- Typical kernel starting up process
- Bash
- More commands
- Shell scripting

Objectives

- Understand many of the UNIX/Linux utilities that are available and how they are classified
- Use the dd utility to copy and convert files
- Make a bootable removable disk
- 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

Understanding UNIX/Linux Utilities

- UNIX/Linux utilities let you
  - Create and manage files
  - Run programs
  - Produce reports
  - Monitor and maintain the system
  - Recover from a range of errors
- New utilities are continually being updated/edited/added in order to make UNIX/Linux run more efficiently

Understanding UNIX/Linux Utilities (cont'd)

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

File Processing Utilities

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

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

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
```

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

The `finger` and `sleep` command

- The `finger` command can be 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 seconds

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

Network Utility commands

- Some commands
  - `dig` – gets IP address from a domain name address
    - `$ dig www.google.com`
  - `ifconfig` – sets up a wired network interface card (obsolete in Debian)
    - Can be used to troubleshooting networking
  - `ip` – sets up network interfaces
    - `$ ip link`
    - `$ ip addr`
  - `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
    - Install with `$ sudo apt install net-tools`

Newer 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`

Communications Utilities

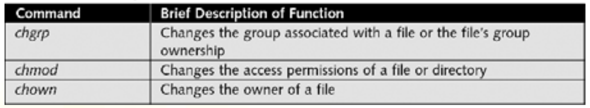
- wall – sends a message to all logged-in users.
- mesg n – denies any real-time messages
- mesg y – accepts any real-time messages
- write – sends a message to a user
- mail – sends e-mail
- talk – allows users to simultaneously ‘chat’ with other logged in users

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

Security Utilities

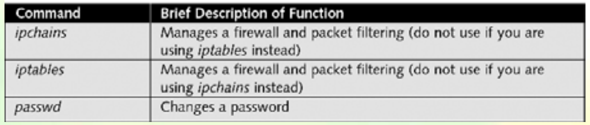
- Obsolete: `ipchains`
- Up-and-coming but not ready: `bpfilter`

Table 8-5 Security utilities



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

Table 8-5 Security utilities (continued)

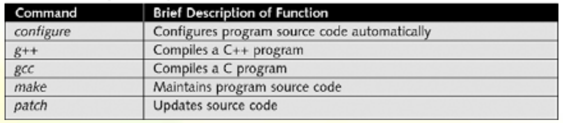


Command	Brief Description of Function
ipchains	Manages a firewall and packet filtering (do not use if you are using *iptables* instead)
iptables	Manages a firewall and packet filtering (do not use if you are using *ipchains* instead)
passwd	Changes a password

Programming and Source Code Management Utilities

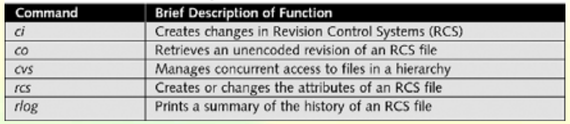
- Linux is written in C programming language
- Get compilers with `$ sudo apt install build-essential`

Table 8-6 Programming utilities



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

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



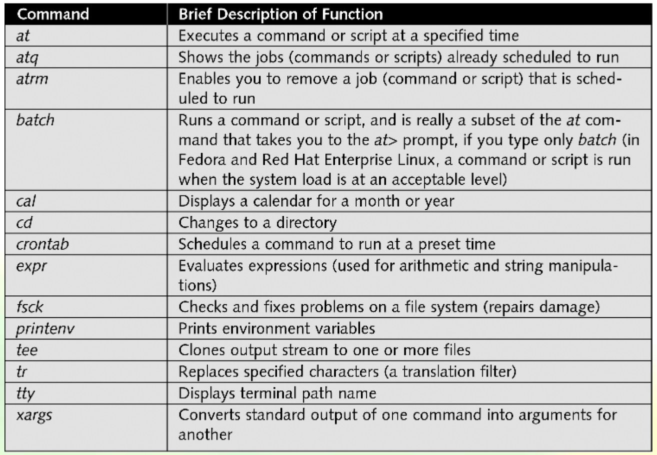
Command	Brief Description of Function
ci	Creates changes in Revision Control Systems (RCS)
co	Retrieves an unencoded revision of an RCS file
cvs	Manages concurrent access to files in a hierarchy
rcs	Creates or changes the attributes of an RCS file
rlog	Prints a summary of the history of an RCS file

Security Utilities on Permission Management

- Not the encryption security, discussed here
- `chgrp` – changes the default group associated with a file
- `chmod` – changes the access permissions of a file or directory
- `chown` – changes the owner of a file or directory

Miscellaneous Utilities

Table 8-8 Miscellaneous utilities



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

Using the dd Command

- Allows you to copy a file and change the format of the destination file
- Has a rich set of 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

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=~/partition.img (backup a partition)
  - # dd if=/dev/hda of=~/hdadisk.img (create a hard disk image)
  - # dd if=hdadisk.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)
- Commonly used options:

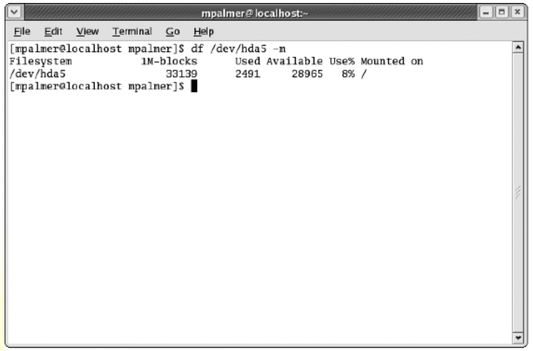
  `if=input_file`
  `of=output_file`
  `conv=ascii` (converts destination format to ASCII)
  `conv=lcase` (converts uppercase to lower)

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

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

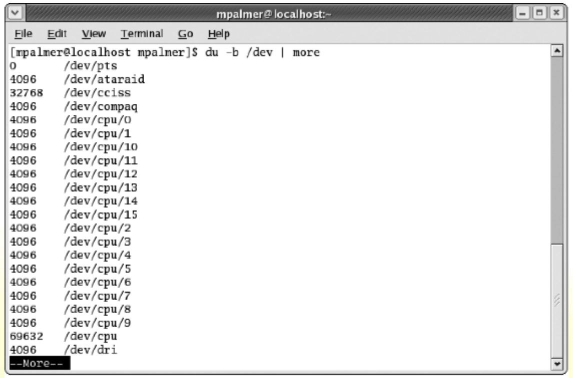


```
[mpalmer@localhost mpalmer]$ df /dev/hda5 -m
Filesystem           1M-blocks      Used Available Use% Mounted on
/dev/hda5                33139      2491     28965   8% /
[mpalmer@localhost mpalmer]$ █
```

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

Using the `du` (disk usage) Utility

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



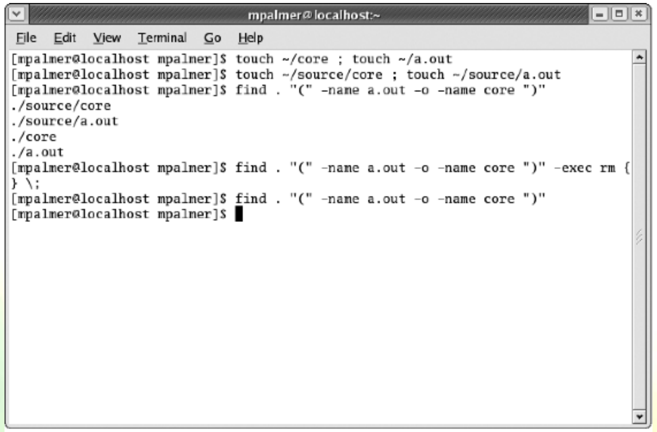
```
[mpalmer@localhost mpalmer]$ du -b /dev | more
0       /dev/pts
4096    /dev/ataraid
32768   /dev/cciss
4096    /dev/compaq
4096    /dev/cpu/0
4096    /dev/cpu/1
4096    /dev/cpu/10
4096    /dev/cpu/11
4096    /dev/cpu/12
4096    /dev/cpu/13
4096    /dev/cpu/14
4096    /dev/cpu/15
4096    /dev/cpu/2
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

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)



```
[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]$ █
```

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

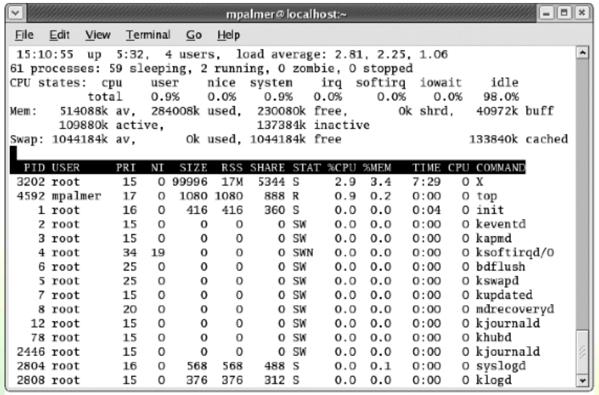
Using System Status Utilities

- 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

Using the `top` Command (cont'd)



```
 15:10:55  up  5:32,  4 users,  load average: 2.81, 2.25, 1.06
61 processes: 59 sleeping, 2 running, 0 zombie, 0 stopped
CPU states:  cpu    user    nice  system    irq  softirq  iowait    idle
           total    0.9%    0.0%    0.9%   0.0%     0.0%    0.0%   98.0%
Mem:   514088k av,  284008k used,  230080k free,       0k shrd,   40972k buff
       109880k active,             137384k inactive
Swap: 1044184k av,       0k used, 1044184k free                  133840k cached

  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 bdflush
    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 khubd
 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
```

Figure 8-11 Sample *top* display

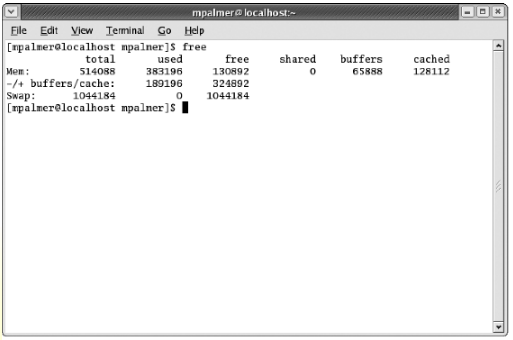
The top utility run without any options specified

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

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]$ 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

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
```

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

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.,

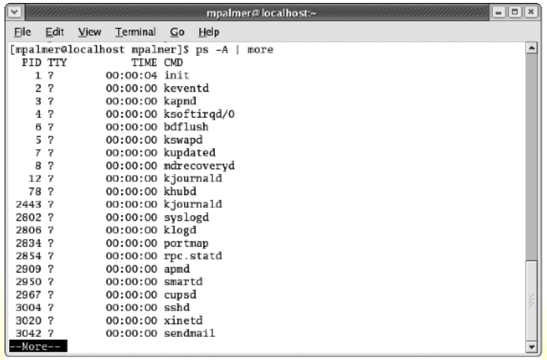
```
$ top&
```

Monitoring Processes

- The `ps` command with the `-A` option shows a list of all system processes currently running

```
$ ps -gaux
```

- is used to display all of the processes running on the system



```
[mpalmer@localhost mpalmer]$ ps -A | more
  PID TTY          TIME CMD
    1 ?        00:00:04 init
    2 ?        00:00:00 keventd
    3 ?        00:00:00 kapmd
    4 ?        00:00:00 ksoftirqd/0
    6 ?        00:00:00 bdflush
    5 ?        00:00:00 kswapd
    7 ?        00:00:00 kupdated
    8 ?        00:00:00 mdrecoveryd
   12 ?        00:00:00 kjournald
   78 ?        00:00:00 khubd
 2443 ?        00:00:00 kjournald
 2802 ?        00:00:00 syslogd
 2806 ?        00:00:00 klogd
 2834 ?        00:00:00 portmap
 2854 ?        00:00:00 rpc.statd
 2909 ?        00:00:00 apmd
 2950 ?        00:00:00 smartd
 2967 ?        00:00:00 cupsd
 3004 ?        00:00:00 sshd
 3020 ?        00:00:00 xinetd
 3042 ?        00:00:00 sendmail
--More--
```

Figure 8-5 Viewing all of the running processes

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:

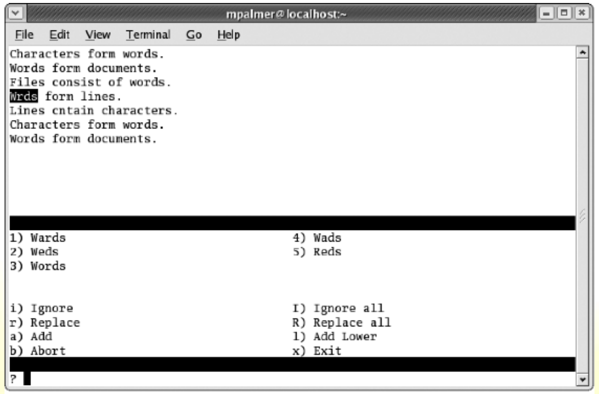
  $ ps

  (Assume the PID number of process p1 is 608)

  $ kill 608

Checking the Spelling of a Document

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



```
Characters form words.
Words form documents.
Files consist of words.
Wrds forn lines.
Lines cntain characters.
Characters form words.
Words form documents.

1) Wards                                 4) Wads
2) Weds                                  5) Reds
3) Words


i) Ignore                                I) Ignore all
r) Replace                               R) Replace all
a) Add                                   l) Add Lower
b) Abort                                 x) Exit
?
```

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

Install ispell with

```
$ sudo apt install ispell
```

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
- If there are no differences, the `cmp` command displays nothing

Summary

- UNIX/Linux utilities are classified into eight major functional areas
- Utility programs are called commands: executed by entering names on the command line
- dd command options allow it to handle copies when other copying methods fail

Summary

- UNIX/Linux utilities are classified into eight major functional areas
- Utility programs are “commands”: executed by entering names on the command line
- 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
- `top` and `free` provide views of the “internals” of the system that can be redirected to a file for system tune-up

Summary (cont’d)

- Run a program in the background by appending & to the end of a command
- `ps` displays all running processes
- `kill` terminates a specific process
- `ispell` scans for spelling errors

Remark

- We have done a lot of works on shell scripting today!!