### Some Useful UNIX Utilities

- Search for files with find
- "pipe" and "filter"
- "sort" and "uniq"
- "cut"
- "diff" and "patch"
- msic

### How to Search for a File?

You have to go through every directory, non practical

#### which <program>

- Searches for a command you typed through the paths defined in **PATH** only
- \$ whereis <program>
  - Search file among a predefined (hard-coded) set of paths
     /{bin,sbin,etc},
     /usr/{lib, bin,include,src,man,sbin},etc
- find Search for files in a directory hierarchy according to some specifications, such as:
  - Pattern of file names (can use shell meta-characters)
  - Modification/access
  - File type, size, mode (permissions), etc.

### find (p113)

- \$ Syntax : find path <expression>
- Expression (options) (table 4.4 on p115)
  - -name/-iname: find . -name "Ab\*.txt"
    - Start from the current directory, look for files starting with Ab and ending with .txt
    - -iname: Not case sensitive
  - ◆ -type: find .. -type TYPE, the TYPE can be:
    - d: directory
    - f: regular file
    - 1: symbolic file, etc, (p, c, s, b)
  - -user or -group
  - ♦ -size: find / -size <+/->N

Start at the root directory to find all files whose size is

- +/-N: more/less than N block
- The size by default is in 512-byte block, i.e. N=+2 for file size > 1024 bytes
- You can specify the unit with N{cbkMG}
  - c for bye
  - b for block (512-bytes, this is the default)
  - k: kilobyte, M for megabyte and G for gigabyte

### **More Selection Criteria**

- ❖ -perm: find . -perm 644
  - Find files (include directories) with permission 644
- Date/Time
  - ♦ -mtime: find . -mtime [+/-]N
    - Find files modified N days ago
  - ♦ -ctime: find . -ctime [+/-]N
    - · Find files created N days ago
  - ♦ -atime: find . -atime [+/-]N
    - Files accessed N days ago
  - ◆ +/-N:
    - +: More than N days ago
    - -: Less than N days ago

Note: if "time" can be substituted with "min" to refer to N minutes on LINUX system

### Actions after 'find'

#### On UNIX system

- "path" is a must
- -print: prints selected files on standard output
- ♦ -ls: executes "ls -lids" on the selected files
- -exec cmd: Executes cmd followed by {} \;

```
find . -name al* -exec ls -l {} \;
```

- Linux system
  - "path" can be dropped if search starting from the current directory
  - Action: Using pipe to xargs

find . -name a1\* | xargs ls -1
find . -name \*.o | xargs rm -f

### The xargs Command

- Build & execute command lines from standard input, such as through a pipe
- Commonly paired with find to perform action upon the found results redirected through pipe "|"

Examples

- find . -name ``\*.txt" | xargs ls -l
- find ~ -type d -size 0 |xargs rmdir
- find . -size +10000 -mtime +90 | xargs tar czvf test.tar.gz
  - · Back up files not modified for more than 90 days
- find ~/cs390 -name \*.o | xargs rm -f
- find ~ -type d -name CVS | xargs rm -rf
- find ~ -type d -name .svn | xargs rm -rf

\*\*Note: you need use **xargs** for **"rm"**, **"ls"** that takes argument (not data stream from the pipe on the left)

## **Pipe and Filter**

- UNIX pipe is a mechanism whereby the output of one process (program) is sent as input to another program
  - The data in pipe can only flow at one direction
  - You can combine many commands with pipes on a single command line, i.e.:

cat file.txt |wc |mail -s "word count" <u>alin@itsc.uah.edu</u>

- Filter is a program designed to process a stream of input data and yield a stream of output data.
  - Filters are often used between two programs/tools
  - A filter processes another program's output, altering it in some manner. The filter's output then can become input to another program, transform/filter the data it receives via pipe

### **Pipe and Filter Examples**

sort sort lines of text files

cat file.txt | sort | mail -s "sorted file" linh@uah.edu

- uniq omit repeated lines
  - Filter adjacent matching lines from INPUT and write to OUTPUT
  - If the file is sorted, uniq ensures that no two lines that it displays are the same

cat file.txt | sort | uniq > uniq.txt
sort file.txt| uniq > uniq.txt

- tee read from stdin (or pipe), write to a file and stdout at the same time
  - who | tee who.txt | grep alex

tr UNIX "translate" utility program

- cat datafile.txt | tr NW nw
- ◆ tr [a-z] [A-Z] < abstract ⇔ tr [:lower:] [:upper:]</pre>

Question: how to remove duplicate lines from a text file?

# The "cut" Utility

Removes sections from each line of text

- -d option specifies the field delimiter, default delimiter is a tab
- -f option specifies the index of fields to be displayed
- -c option specifies the characters to be displayed
- ✤ Examples: date Fri Jan 18 17:35:10 CST 2019
  - ♦ date | cut -f 2-4 →?
  - ◆ Date | cut -d `` ″ -f 2-4 →?
  - ♦ date | cut -c 1-3 → Fri
  - ♦ date | cut -d `` " -f 1-3 → Fri Jan 18
  - \$ cut -d: -f1-5 /etc/passwd
    - Displays the fields from 1 to 5, fields separated by ":"

\*\*There is "paste", use "man paste" to see what it does

## "diff" and "patch"

### diff

 Display the differences between two files (or directories) normally used to compare the modified text file (directory) to its original file (directory) to check the change

### patch

- Apply the diff file (also called patch file or patch) to update the original file
- Updating files with patches are often referred to as patching files
- diff is a UNIX command, but patch is not
  - patch written by Larry Wall in 1985, later became part of the GNU project

### "diff" Two Text Files

### diff -u ../v1.0/test.txt test.txt

- ♦ -u : unified format...(check man page for diff)
- ♦ -w: ignore white/empty space
- ◆ -i: ignore case difference
- ♦ -b: ignore blank line

```
--- ../v1.0/test.txt 2008-08-23 21:12:40.000000000 -0500
+++ test.txt 2008-08-23 21:13:13.00000000 -0500
@@ -1,5 +1,5 @@
Hello!
-This is my first version of the applications.
+This is my improved version of the applications.
Bye
Bob
```

### "diff" Two Directories

#### diff -uNr v1.0 v1.1

```
--- v1.0/new.txt 1969-12-31 18:00:00.00000000 -0600
+++ v1.1/new.txt 2008-08-23 21:13:45.00000000 -0500
00 -0, 0 +1, 2 00
+Hi, I have added a new file to this application.
+
diff -uNr v1.0/test.txt v1.1/test.txt
--- v1.0//test.txt 2008-08-23 21:12:40.00000000 -0500
+++ v1.1/test.txt 2008-08-23 21:13:13.000000000 -0500
00 -1,5 +1,5 00
Hello!
-This is my first version of the applications.
+This is my improved version of the applications.
Bye
```

Bob

"patch"

### --apply a diff file to an original

#### patch < patchfile.txt</pre>

- the file patchfile.txt was created using diff (??)
- "<" : read the input from a patch file</p>
- The names of the files to be patched are normally determined from the patch file
- Apply the "patch" to the original to get files updated
  - Change to the directory where the old version file is
  - Run: patch < diff.txt
    </pre>
    - "patch will look for file (described in the diff.txt file) in the current directory and apply the differences to it

# "patch" Continued

Some useful options

### **\*** "-pNUM" ⇔ --strip=NUM

 Depends on how the patch file was created, you might need to use this option so the patch program can find the to-be-patched files

NUM represents the number of "/" to be stripped off

- -p0: keep whatever in the patch file
- -p1: strip one "/" off
- -p2: strip two "/" off
- Ex: in diff: ../cs390/ex1/text1.txt
- p0, p1, p2, or without "pnum" (basename of the files)
- ✤ -b: Back up the original files

--dry-run

 Print the results of applying the patches w/o actually changing any files

## A Few More Useful Commands

hostname, dnsdomainname, etc

users

Shows a list of users currently log on the system

who

 Shows the list of users currently logon and when they logged on

#### which <cmd>

Display the available "cmd" including the locations "path"

✤ date

Display the current date : date +%F or date "+%F %T"

✤ WC

Count the lines, words and bytes of STDIN

### System Information

- "uname -< options>"
  - -s: kernel name
  - -n: node name, such as "lightning"
  - -r: kernel release
  - -a: display all the information
  - Using "man uname" to learn more ...
- System information files under /proc/cpuinfo
  - more /proc/meminfo

# **Disk/Space Usage**

### \*du

- No option, lists disk usage for all the directories from current place
- "-s": gives the sum of the usage of the current directory
- "-a": lists all, including each file's size
- "-h": human readable
- "-m": display size in MB
- "-k": display size in KB
- df Report disk file system (partitions) usage

### dos2unix & unix2dos

DOS/Mac /UNIX text file format converter

- Why needs this? Discrepancy of text line breaker on different platforms
  - DOS (Windows)
    - Combination of two characters \r\n
      - CR: Carriage Return (CR) followed by a Line Feed (LF)
  - Mac: single character (CR): \r
  - Unix/Linux : single character (LF) \n
- https://www.oreilly.com/library/view/mac-osx/0596004605/ch01s06.html

"...To make matters still more interesting, until OS X came along, OS-specific line breaks stayed in their own environment and didn't play nicely with others. Windows understood only its brethren, Unix cackled madly at anything else, and the Mac just grinned knowingly. OS X, however, understands both the original Mac line break and Unix line breaks..."