

## Basic Command Examples

Extract a field from a delimited file:

```
cut -d <delim> -f <field_num>
```

Extract a range characters from a file

```
cut -c <start>-<finish>
```

Find files that were modified today:

```
find . -type f -mtime 0
```

Perform an action on a subset of files:

```
find . -type f | head -n 10  
| xargs -I {} <cmd>
```

Determine if a file is being used:

```
lsof -V <filename>
```

Grep with line numbers and context:

```
grep -nA <num_lines> <pattern> *
```

Show all ports in use: netstat -alp

Show port statistics: netstat -s

Character substitution: tr '<from>' '<to>'

Delete characters: tr '<char>' -d

Show processes as a tree:

```
pstree -ap | grep -C <lines> <proc>
```

Get the current time as a unix timestamp:

```
date +%s
```

Get the current time in ISO format:

```
date +%F\ %T
```

Execute cmd with watch and mark changes:

```
watch -d '<cmd>'
```

Merge two files together (columns):

```
paste <file1> <file2>
```

Execute a command as a different user:

```
su -c <cmd>
```

## Sed Examples

Basic substitution:

```
sed -e 's/<old>/<new>/g' <input>  
> <output>
```

Delete lines: sed -e '/<pattern>/d' <input>

Multiple substitutions: sed -e

```
's/<old>/<new>/g;s/<old2>/<new2>/g'  
<input>
```

Modify a file in-place:

```
sed -i 's/<old>/<new>/g' <input>
```

## tr Classes

**alnum** Alphanumeric Characters

**alpha** Alphabetic Characters

**blank** Whitespace Characters

**cntrl** Control Characters

**digit** Numeric Characters

**graph** Graphic Characters

**lower** Lower-Case Alphabetic Characters

**print** Printable Characters

**punct** Punctuation Characters

**space** Space Characters

**upper** Upper-Case Alphabetic Characters

**xdigit** Hexadecimal Characters

## Basic FTP Commands

**ascii** Set the transfer type to ascii.

**bell** Sound an alarm when transfer is finished.

**binary** Set the transfer type to binary.

**cd <directory>** Change remote directory.

**exit** Exit the FTP client.

**get** Download a file from the server.

**lcd** Change local directory.

**put** Upload a file to the server.

**tick** Display the number of bytes transferred.

## Bash Tricks

One-line for loop:

```
for file in *; do <cmd>; done
```

Use the output of one command in another:

```
echo 'foo' > 'date +%s'.txt
```

Same, but using pipes:

```
diff <(find /dir1) <(find /dir2)  
tar cvf >(gzip -c > dir.tgz) dir
```

Fix a mistake in the previous command:

```
^old^new
```

Use the last arg. from the last command:

```
rm !$
```

Run command 2 if command 1 succeeds:

```
<cmd1> && <cmd2>
```

Repeat command for every item in braces:

```
touch {1..10}.txt ; rm foo.{obj,bin,bar} ;  
cp <file>{,.bak}
```

Start/end of line: Ctrl-A ; Ctrl-E

Follow the execution of your shell script:

```
#!/bin/bash -x
```

Time multiple commands:

```
time sh -c '<commands>'
```

## Vim Specific Tips

Open a remote file using vim and ssh:

```
vim scp://user@host//path/to/file
```

Open many files that match a pattern:

```
args **/*.java
```

Modify the contents of the matched files:

```
argdo <command> | w
```

Modify the contents of the opened buffers:

```
bufdo <command> | w
```

Autocomplete Word: Ctrl-n Ctrl-p

Indent/Outdent (Insert Mode):

```
Ctrl-t, Ctrl-d
```

Paste to command line: Ctrl-r <reg>

Replace with register: :s/string/\=@<reg>/

## Redirection/Processes

Create a named pipe: `mkfifo <name>`

Process Substitution (Ex):

```
diff <(cat foo.txt) <(cat bar.txt)
```

Use piping in find + exec:

```
find . -exec sh -c  
'grep "$1" > "$1.out"' -- {} \;
```

Use piping with xargs:

```
<cmd> | xargs -I {} sh -c  
'grep "$1" > "$1.out"' -- {} \;
```

## SSH Tips

Copy IDs:

```
ssh-copy-id -i <pub_key> user@host
```

## File/Dir/Disk Tips

Find largest files/dirs:

```
du -hsx * | sort -rh | head -10
```

Same, using find:

```
find <dir> -print '%s %p\n'  
| sort -nr | head -10
```