UNIX Reference

> Childhood Cancer Data Lab, Alex's Lemonade Stand Foundation

Table of Contents *generated with DocToc*

UNIX Shortcuts and symbols

Shortcut/symbol	Description
`Ctrl+C`	Kills current process/command.
`tab` (tab key)	Autocomplete.
↑ (up arrow)	Scroll back through previous commands in an interactive shell.
~ (tidle)	Represents your home directory.
`.`	Represents the current directory you are in
`..`	Represents one directory backwards. For example, the relative path `..` refers to "one directory back," the relative path `../..` refers to "two directories back," and so on.
`*`	Wildcard symbol.
|	Pipe symbol (located on the backslash key). Join a string of UNIX commands together into a "pipeline."
> and >>	Redirect ("send") output from the command line, for example to a file. > will *overwrite* the destination file, and >> will *append to* the destination file.

UNIX commands

Navigating directories

- **`cd`**: **C**hange **d**irectory

```
# Change directory to the given path
cd path/where/I/want/to/go

# Go one directory back in the hierarchy
cd ..

# Two ways to go to your home directory:
cd
cd ~
```

- **`ls`**: **L**i**s**t files and directories

```
# List contents of working directory
ls

# List contents of a different directory
ls path/to/directory/I/want/to/list

# List files in long format with the -l flag`
ls -l

# Include hidden files/directories with the -a flag
ls -a

# Show "human-readable" file sizes with the -h flag
ls -h

# Flags can be combined, in any order, for example to
# show all files in long format with human-readable sizes
ls -lah
```

- **`pwd`**: **P**rint the **w**orking **d**irectory.
  - Display the full path of the current/working directory

```
# print the current directory
pwd
```

Manipulating files and directories

- **`cp`**: **C**o**p**y a file or directory.
  - The `cp` command does not change the original file. It is similar to "copy and paste."

```
# Make a copy of oldfile.txt called newfile.txt
cp oldfile.txt newfile.txt

# Make a copy of the directory directory_original to be called
↪  directory_copy
# Copying directories requires the -r flag
cp -r directory_original directory_copy
```

- **`mkdir`**: **M**a**k**e a new **dir**ectory

```
# Create a new directory called new_directory
mkdir new_directory

# Use the -p flag to avoid errors if the directory already exists
mkdir -p new_directory_that_might_already_have_been_created
```

- **`mv`**: **M**o**v**e or rename a file or directory.
  - The `mv` command *removes* the original file. It is similar to "cut and paste."

```
# Rename a file from oldname.txt to newname.txt
mv oldfile.txt newfile.txt

# Move file.txt into a new directory
mv file.txt path/to/new/directory/file.txt
```

- **`rm`**: **R**e**m**ove a file or directory
  - **This is *permanent.*** You cannot recover files deleted with `rm`

```
# Permanently delete a file
rm file_I_dont_need.txt

# Permanently delete a directory
# Removing directories requires the -r flag
rm -r directory_I_dont_need/
```

- **`touch`**: Create an new (empty) file

  - When run on existing files, the `touch` command updates its "last modified by" date (it "touches" the file!)

```
# Create a new file
touch name_of_the_new_file_you_want_to_create.txt

# Update modification date of existing file
touch existing_file_to_update_modification.txt
```

Working with files

- **head (`tail`): Display the first (last) ten lines of a file

```
# See first 10 lines of a file
head file_I_want_to_peek_at.txt

# See last 10 lines of a file
tail file_I_want_to_peek_at.txt

# Use the -n flag to specify a different number of lines,
#  for example 20:
head -n 20 file_I_want_to_peek_at.txt
```

- **`less`**: View the contents of a file in the terminal
  - This command was named `less` named because it is an improved version of the related command `more`, which also lets you view file contents, and…"less is more."
  - You can scroll through the view with up and down arrows
  - You can exit the view by typing q

```
# View a file's contents
less file_I_want_to_look_at.txt
```

- **`gunzip`**: Decompress a `.gz`-compressed file
  - The `gunzip` command removes the original file and creates a new file of the same name but *without* the added extension `.gz`

```
# Decompress very_big_file.txt.gz and create very_big_file.txt
gunzip very_big_file.txt.gz
```

```
# Use the -c flag and redirection to retain the original file
gunzip -c very_big_file.txt.gz > very_big_file.txt
```

- **`gzip`**: Compress a file to `.gz` format
  - The `gzip` command removes the original file and a creates a new file of the same name but with the added `.gz` extension

```
# Compress very_big_file.txt and create very_big_file.txt.gz
gzip very_big_file.txt

# Use the -c flag and redirection to retain the original file
gzip -c very_big_file.txt > very_big_file.txt.gz
```

- **`wc`**: **W**ord **c**ount. Count the number of words, lines, characters, and/or bytes in a file

```
# Count all values (words, lines, characters, and bytes)
wc file_I_want_to_count.txt

# Only count the number of lines with the -l flag
wc -l file_whose_lines_I_want_to_count.txt

# Only count the number of words with the -w flag
wc -w
```

Other tasks

- **`echo`**: Print

```
# Print a message from a script
echo "Running Step 2"
```

- **`curl`**:C**lient** URL**. Download the contents of a website
  - By default, the downloaded website text gets sent to the command line directly

```
# Download and save an internet file, using redirection
curl https://database.com/interesting_data.csv > interesting_data.csv

# Use the -o flag to specify the output file, similar to redirection
# The destination filename must directly follow the -o flag
curl https://database.com/interesting_data.csv -o interesting_data.csv
```

```
# Use the -O flag to automatically save a file as the same name as it
#  appears on the internet (here, interesting_data.csv)
curl -O https://database.com/interesting_data.csv
```

- **man**: See a command's documentation in the **man**ual
  - *Caution*: The so-called "man pages" are not always very clear, but they are still often helpful, in particular for seeing different flags you can use with a command
  - You can scroll through the documentation with up and down arrows
  - You can exit the documentation by typing q

```
# See the documentation for the echo command
man echo
```