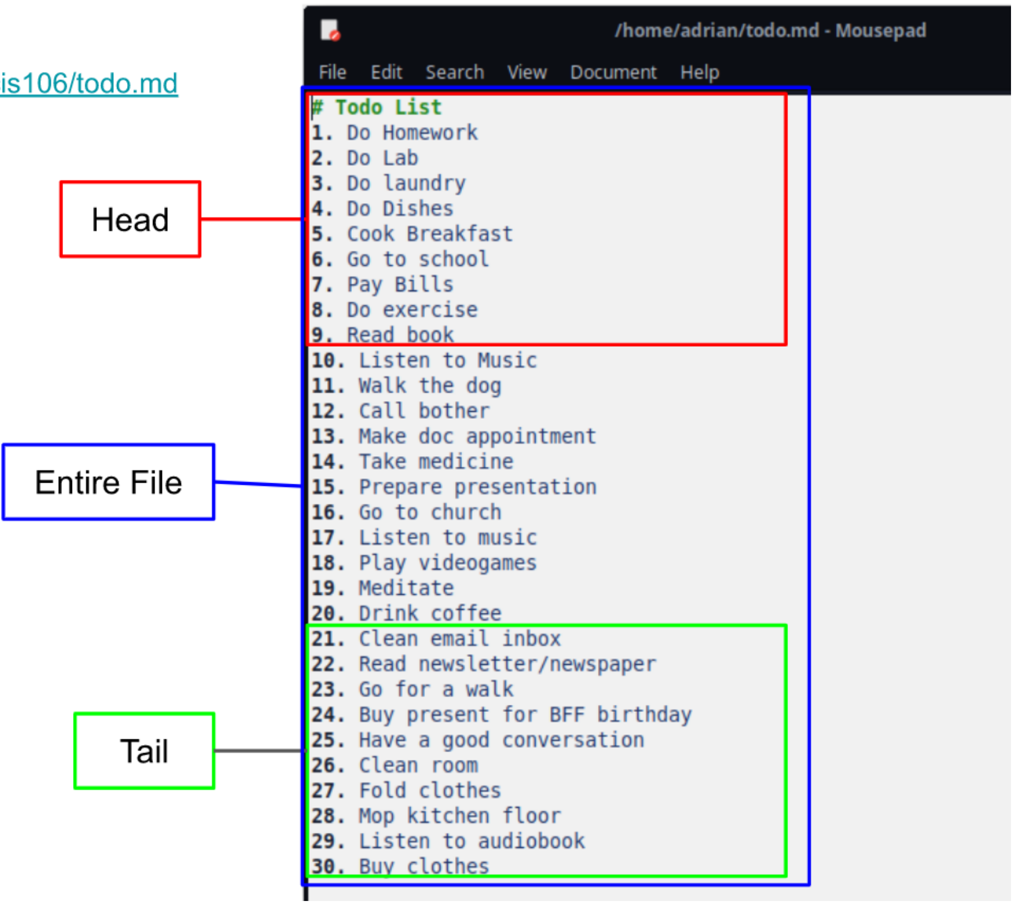
Handling Text Files

Handling text files

- Linux offers a lot of command line tools for handling text
  - cat
  - tac
  - more
  - less
  - head
  - diff
  - tail
  - cut
  - past
  - sort
  - wc
  - tr
  - grep

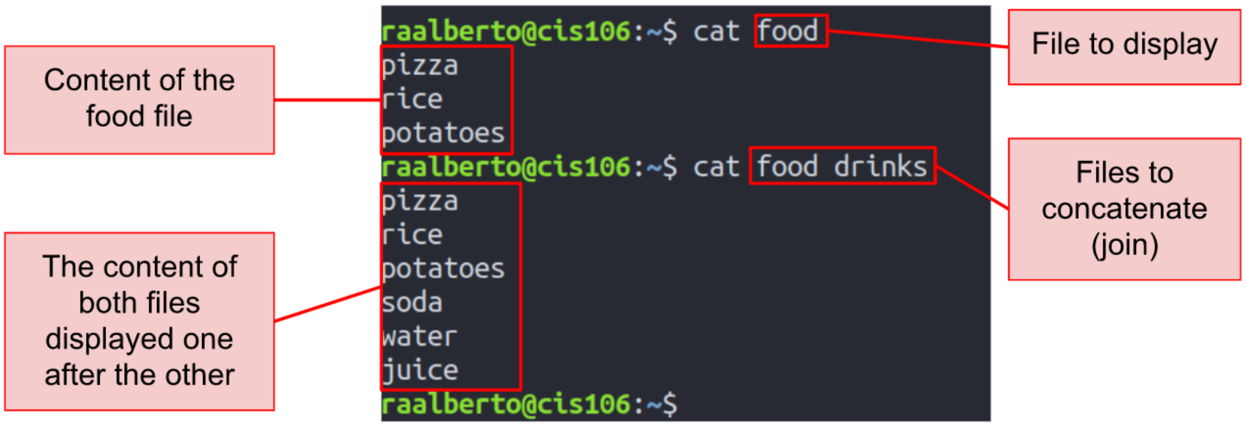
To download this file:
https://robertalberto.com/cis106/todo.md


/home/adrian/todo.md - Mousepad
File Edit Search View Document Help
Head
Entire File
Tail

```
# Todo List
1. Do Homework
2. Do Lab
3. Do laundry
4. Do Dishes
5. Cook Breakfast
6. Go to school
7. Pay Bills
8. Do exercise
9. Read book
10. Listen to Music
11. Walk the dog
12. Call bother
13. Make doc appointment
14. Take medicine
15. Prepare presentation
16. Go to church
17. Listen to music
18. Play videogames
19. Meditate
20. Drink coffee
21. Clean email inbox
22. Read newsletter/newspaper
23. Go for a walk
24. Buy present for BFF birthday
25. Have a good conversation
26. Clean room
27. Fold clothes
28. Mop kitchen floor
29. Listen to audiobook
30. Buy clothes
```

Cat

- The cat command is used for **displaying the content of a file**.
- Cat is short for **concatenate** which is the the command intended use.
- Concatenation means joining two strings together.
- Usage:
  - **cat + file to display**
  - **cat + file 1 + file 2**

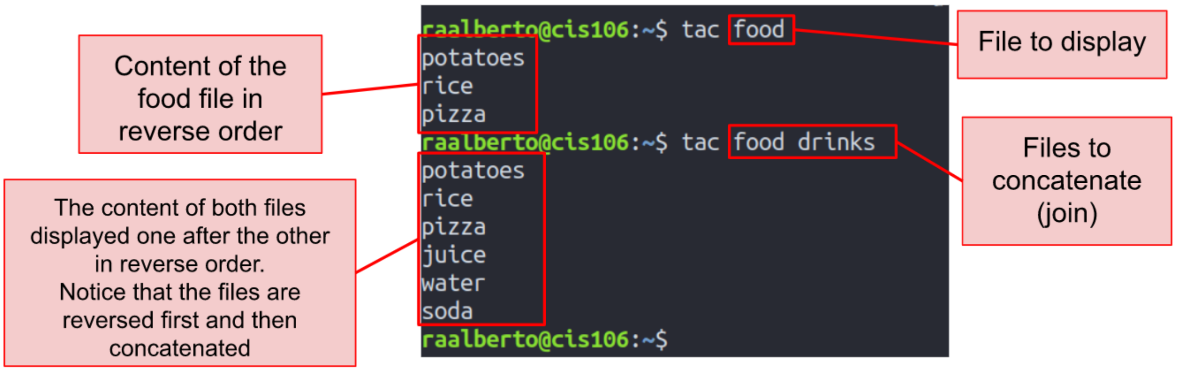


```
raalberto@cis106:~$ cat food
pizza
rice
potatoes
raalberto@cis106:~$ cat food drinks
pizza
rice
potatoes
soda
water
juice
raalberto@cis106:~$
```

File to display
Content of the food file
Files to concatenate (join)
The content of both files displayed one after the other

Tac

- The tac command is used for displaying the content of a file in **reverse order** in a line by line basis
- The tac command can also **concatenate files** in reverse order
- Usage:
  - **tac + file to display**
  - **tac + file 1 + file 2**



```
raalberto@cis106:~$ tac food
potatoes
rice
pizza
raalberto@cis106:~$ tac food drinks
potatoes
rice
pizza
juice
water
soda
raalberto@cis106:~$
```

File to display
Content of the food file in reverse order
Files to concatenate (join)
The content of both files displayed one after the other in reverse order. Notice that the files are reversed first and then concatenated

More

- The **more** command is a pager program used for displaying the content of text file one page at a time.
- Usage:
  - > **more** + file to view.
  - > *For more information, read the man page of the more command.*

Less

- The **less** command is another pager program that displays the content of file 1 page at time.
- **Less** is faster than more when dealing with large files since it loads 1 page at time.
  - Usage:
    - > **less** + file to view
    - > *For more information, read the man page of the more command.*

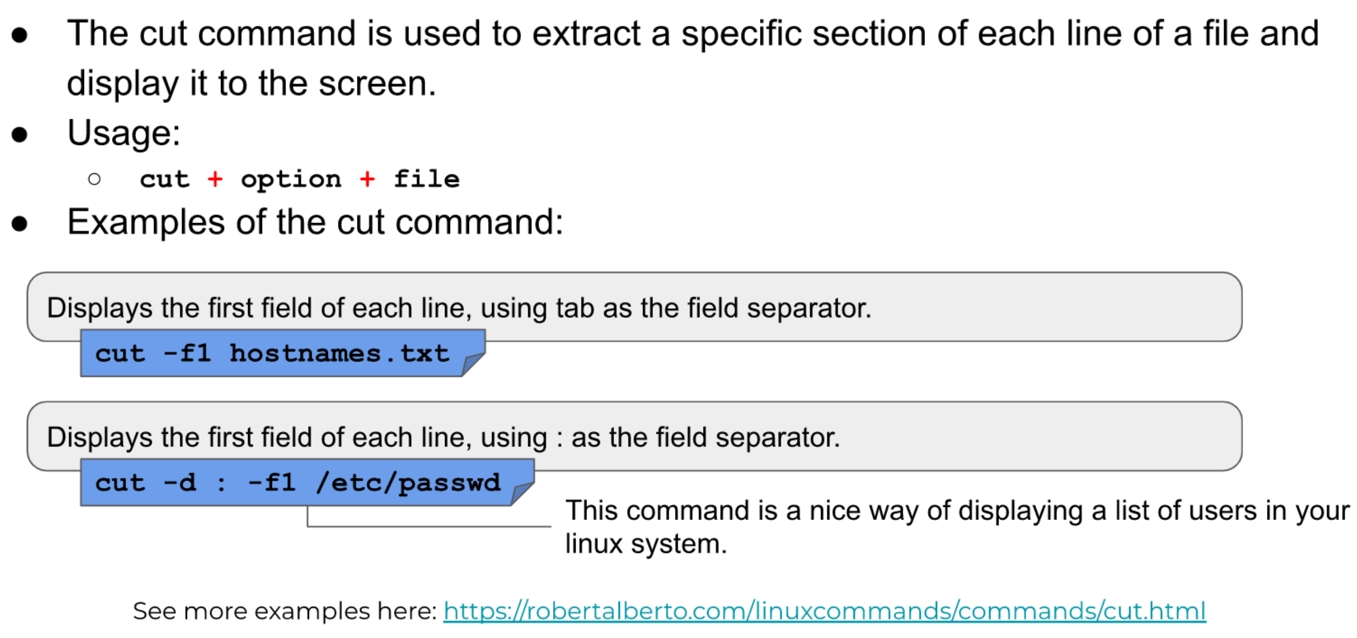
Head

- The **head** command displays the top **N** number of lines of a given file.
- By default, it prints the first 10 lines.
  - Usage:
    - > **head** + **option** + **file**
    - > *For more information, read the man page of the more command.*

Tail

- The **tail** command displays the las **N** number of lines a given file.
- By default, it prints the last 10 lines.
  - Usage:
    - > **tail** + **option** + **file**
    - > *For more information, read the man page of the more command.*

cut



- The cut command is used to extract a specific section of each line of a file and display it to the screen.
- Usage:
  - `cut + option + file`
- Examples of the cut command:

Displays the first field of each line, using tab as the field separator.

```
cut -f1 hostnames.txt
```

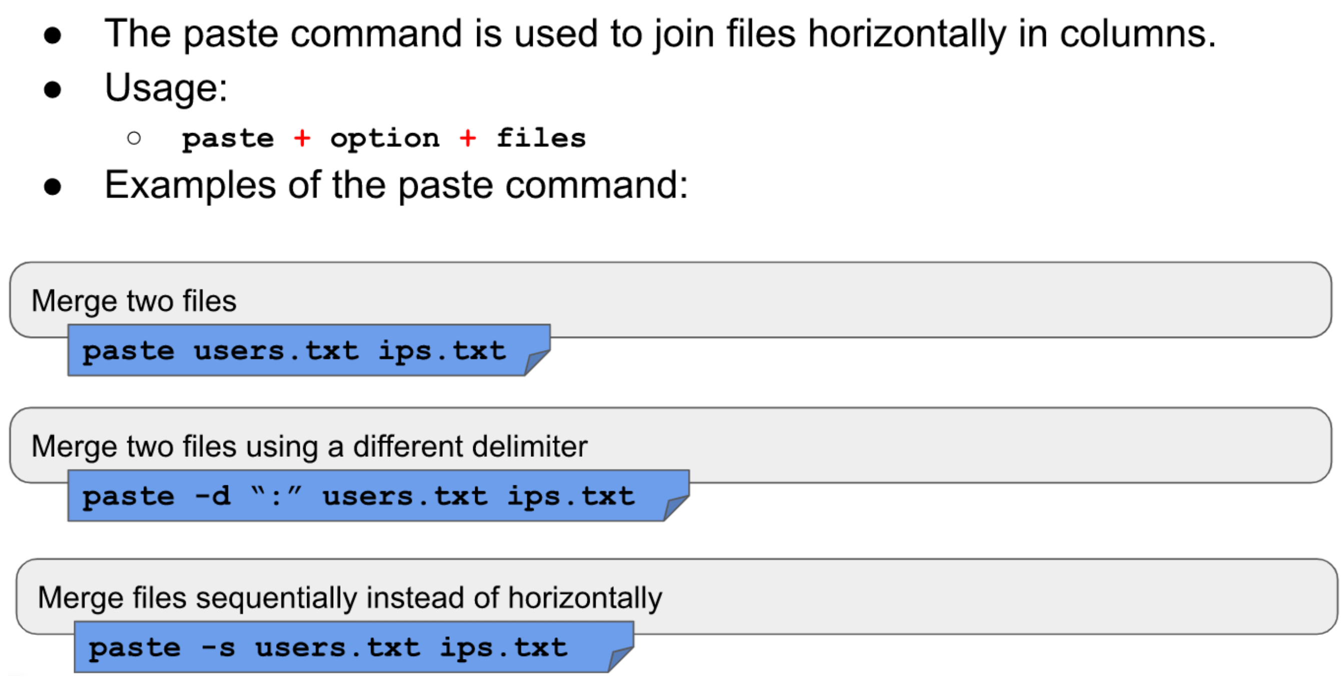
Displays the first field of each line, using : as the field separator.

```
cut -d : -f1 /etc/passwd
```

This command is a nice way of displaying a list of users in your linux system.

See more examples here: https://robertalberto.com/linuxcommands/commands/cut.html

Paste



- The paste command is used to join files horizontally in columns.
- Usage:
  - `paste + option + files`
- Examples of the paste command:

Merge two files

```
paste users.txt ips.txt
```

Merge two files using a different delimiter

```
paste -d ":" users.txt ips.txt
```

Merge files sequentially instead of horizontally

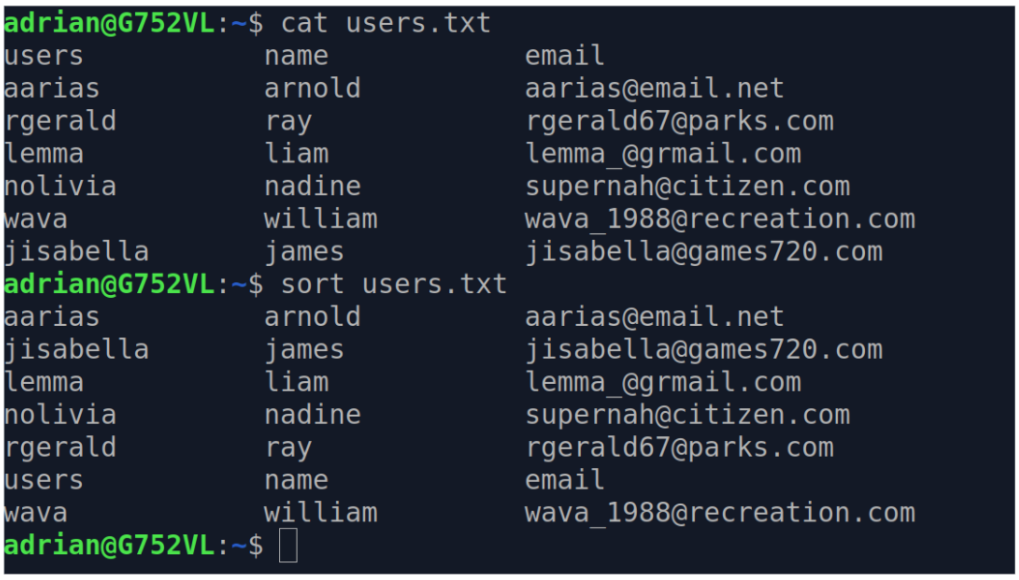
```
paste -s users.txt ips.txt
```

Sort

- The sort command is used for sorting files.
- Sorting means arranging the content of the file in a particular order.
- The sort command sorts the contents of a text file, line by line.
- The sort command supports sorting:
  - Alphabetically
  - in reverse order
  - by number
  - by month
- You can use sort for sorting by column number.
- Sort can be used ignoring case sensitivity
- Sort can return whether a file is sorted or not (this is really useful in scripts)
- By default, the sort command interprets a blank space as the default field separator.
- **The sort command follows this order unless specified otherwise:**
  - Lines starting with a number will appear before lines starting with a letter.
  - Lines starting with a letter that appears earlier in the alphabet will appear before lines starting with a letter that appears later in the alphabet.
  - Lines starting with a lowercase letter will appear before lines starting with the same letter in uppercase.

Assuming you have a file with a list of users. Sort the file.

```
sort users.txt
```



```
adrian@G752VL:~$ cat users.txt
users           name            email
aarias          arnold          aarias@email.net
rgerald         ray             rgerald67@parks.com
lemma           liam            lemma_@grmail.com
nolivia         nadine          supernah@citizen.com
wava            william         wava_1988@recreation.com
jisabella       james           jisabella@games720.com
adrian@G752VL:~$ sort users.txt
aarias          arnold          aarias@email.net
jisabella       james           jisabella@games720.com
lemma           liam            lemma_@grmail.com
nolivia         nadine          supernah@citizen.com
rgerald         ray             rgerald67@parks.com
users           name            email
wava            william         wava_1988@recreation.com
adrian@G752VL:~$
```

wc

- The **wc** command is used for printing the number of lines, and bytes in a file.
- Usage:
  - > **wc** + **option** + **file**

Tr

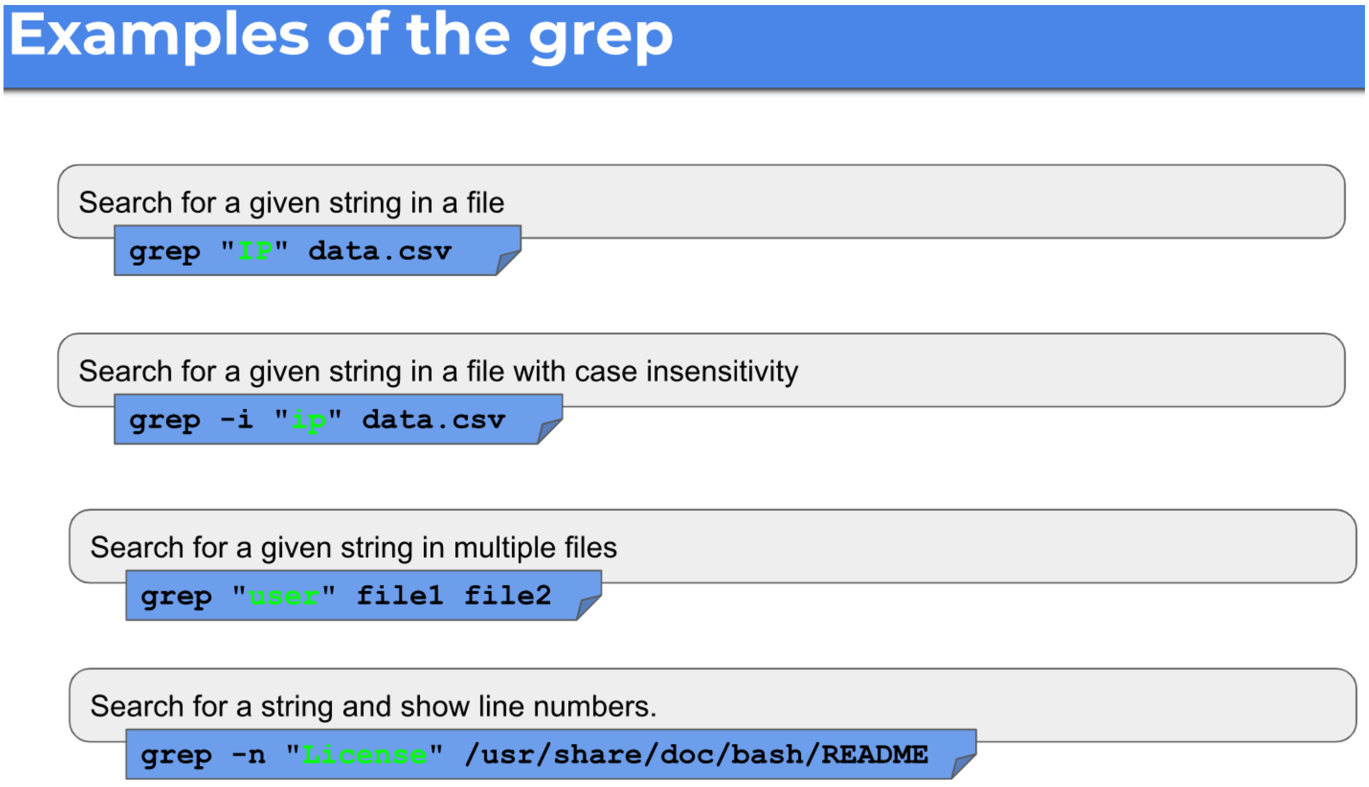
- The **tr** command is used for translating or deleting characters from standard output.
  - > **standard output** | **tr** + **option** + **set** + **set**

Diff

- The **diff** command compares files and displays the differences between them
    - > diff + option + file1 + files2

Grep

- The **grep** command is used to match a string pattern from a file or standard output when using the pipe
    - > grep + option + pattern to match + file
    - standard output + pipe | + grep + pattern to match



Examples of the grep

Search for a given string in a file

```
grep "IP" data.csv
```

Search for a given string in a file with case insensitivity

```
grep -i "ip" data.csv
```

Search for a given string in multiple files

```
grep "user" file1 file2
```

Search for a string and show line numbers.

```
grep -n "License" /usr/share/doc/bash/README
```

Rev

- The **rev** command is used for reversing the characters position in a given text.
    - > **rev** + file

Working with I/O Redirection.

- In lnux, we can redirect the **input and output** of commands to and from files, as well as connect multiple commands together into powerful command piperlines.

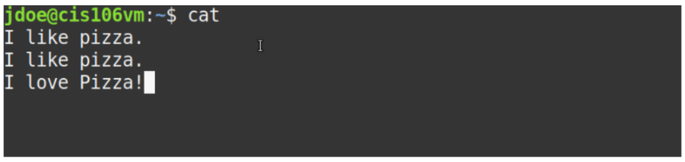
Redirecting STDOUT and STDERR

- To redirect standard output, we use: >
  - **Example:**
    - `ls -lax > list_of_files.txt`
- To redirect standard error, we use: 2>
  - **Example:**
    - `cat badFile.txt 2> error_cat_command.txt`
- To redirect standard output and append the output to a file, we use: >>
  - **Example:**
    - `ls -1alh >> list_of_files.txt`
- We can use the output redirection to create an empty file:
  - **Example:**
    - `> newfile`
    - `: > newfile` (in older versions of bash)
- We can also get rid of output that we do not want:
  - **Example:**
    - `ls -l ~/Downloads ~/documents 2> /dev/null`

Something interesting about the cat command

What happens when you use cat without any file?

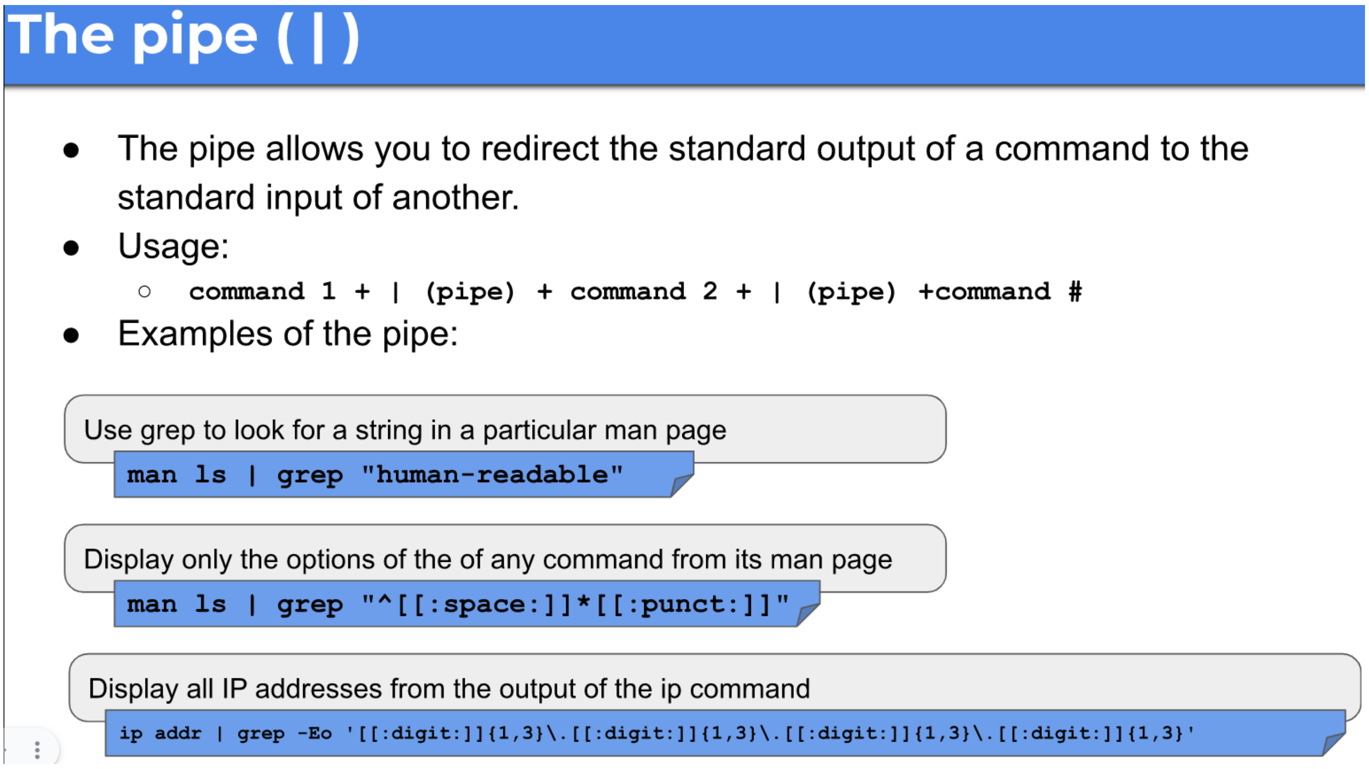
- In the absence of filename arguments, cat copies standard input to standard output, which means that cat will take any text that you type and display it in the terminal. Looking like this:



```
jdoe@cis106vm:~$ cat
I like pizza.
I like pizza.
I love Pizza!
```

- To exit press CTRL + d (hold down the Ctrl key and press “d”).
- You can redirect the output of cat and essentially use cat to create a file with any given text you want.

The pipe (|)



The pipe (|)

- The pipe allows you to redirect the standard output of a command to the standard input of another.
- Usage:
  - `command 1 + | (pipe) + command 2 + | (pipe) +command #`
- Examples of the pipe:

Use grep to look for a string in a particular man page

```
man ls | grep "human-readable"
```

Display only the options of the of any command from its man page

```
man ls | grep "^[[:space:]]*[[:punct:]]"
```

Display all IP addresses from the output of the ip command

```
ip addr | grep -Eo '[[:digit:]]{1,3}\.[[:digit:]]{1,3}\.[[:digit:]]{1,3}\.[[:digit:]]{1,3}'
```

Alias

Creating you own commands with alias

- **What is alias?** - A shorthand for a more complicated command.
- How to create an alias?
  - `alias name_of_alias="command here"`
- **Examples:**
  - An alias to upgrade a linux (debian system):
    - `alias update="sudo apt update; sudo apt upgrade -y; sudo apt full-upgrade -y"`
  - An alias to obtain a computers public ip:
    - `alias publicip="curl ifconfig.me && echo ''"`
  - Some useful git aliases:
    - `alias add="git add ."`
    - `alias push="git push"`
    - `alias merge="git merge"`
    - `alias pull="git pull"`
    - `alias checkout="git checkout -b"`
    - `alias commit="git commit -m"`
    - `alias qpush="git add .; git commit -m 'quick push'; git push"`

- Before creating an alias, always check to see if the word you will use is reserved. Use the type command to figure it out.
- For example:
  - type random - random is a reserved word because there already is a command using it.
  - type randomXYZ - is a good choice because there isn't a command using it.
- If you make an alias to a reserved word, you may break your system because the original command will no longer be used when using the alias.
- To make your aliases permanent, place them either at the end of your .bashrc file or in Ubuntu, place it in the file .bash_aliases located in your home directory.
- Aliases are a good way of remembering tough commands.
- To remove an alias, use the unalias command
  - unalias quoteanime

Sources

Sources

- Blum, Richard, and Christine Bresnahan. *Linux Command Line and Shell Scripting Bible*. John Wiley & Sons, 2021.
- Shotts, William E. *The Linux Command Line: a Complete Introduction*. No Starch Press, 2019.