

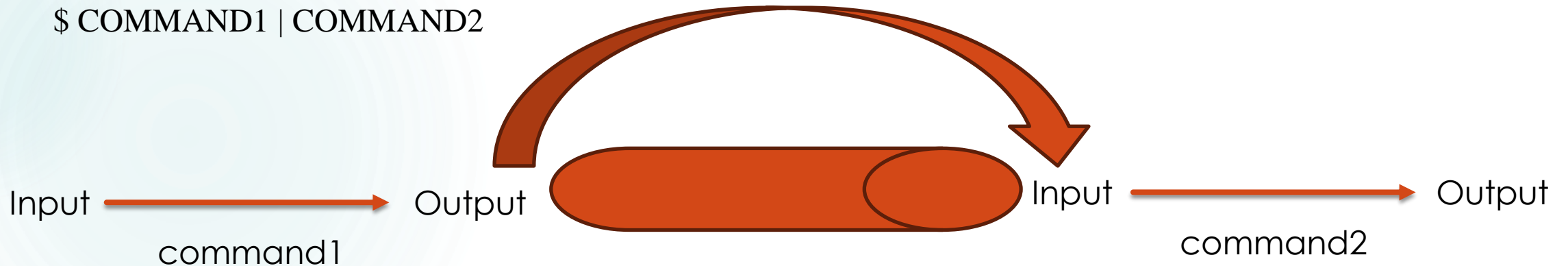
Piping in Linux

- The "pipe" is a feature of the command line shell that allows you to connect the output of one command to the input of another command.
- The linux *pipe* is represented by a vertical bar

* | *

- SYNTAX:

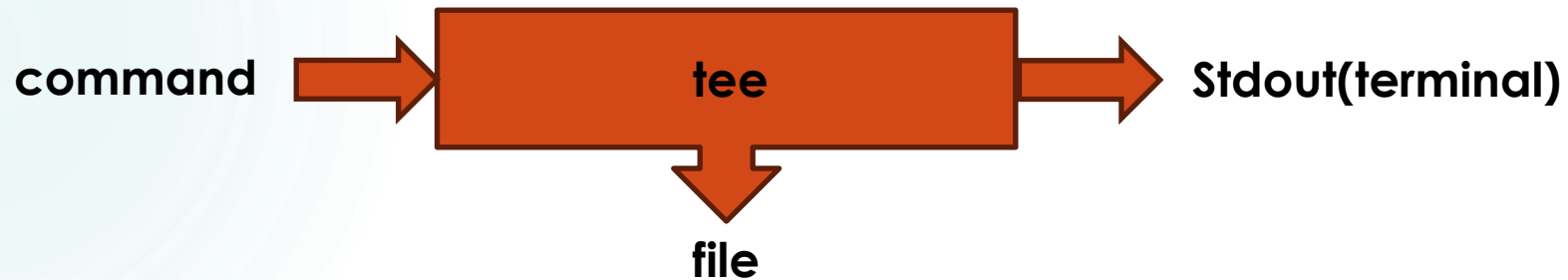
\$ COMMAND1 | COMMAND2



Tee command

- It allows you to read standard input and write it to both standard output(terminal) and one or more files.
- SYNTAX:

`$ COMMAND | tee [OPTION]... [FILE]...`



Head command

- It print the first lines of a text file or stream. By default, it prints the first 10 lines of a file, but you can also specify a different number of lines to display using the -n option.
- SYNTAX:

```
$ head [OPTIONS] [FILENAME]
```

Examples

- To display maximum 10 lines of the file
- To to display first 15 lines of the file
- To display first 10 lines of more then one file
- To display first 7 line from both files
- To display first 1000 lines of file.
- To display first 1 MB from file.

tail command

- It print the last lines of a text file or stream.
- By default, it prints the last 10 lines of a file, but you can also specify a different number of lines to display using the -n option.
- SYNTAX:

\$ tail [OPTIONS] [FILENAME]

Examples

- To display last 10 lines of the file
- To display last 20 lines of file
- To display last 10 lines from file and get real time output as file written

more command

- more command allows you to view the contents of a text file one page at a time.
- The text is displayed on the screen one page at a time, and you can use the space bar to move forward one page or the enter key to move forward one line at a time
- SYNTAX:

\$ more [OPTIONS] [FILENAME]

Examples

- To read file page by page
- To start reading file from particular line number
- To piping the output to more. If ls is going to print lots of text at once we can pipe it output to more command.

less command

- less command allows you to view the contents of a text file one page at a time.
- The text is displayed on the screen one page at a time.
- `less` command provides additional features and greater functionality than the `more` command.
- SYNTAX:

```
$ less [OPTIONS] [FILENAME]
```

Examples

- To read file page by page
- To view multiple files at once
- To display the line number along with output
- To pipe the output of other command to *less*

cut command

- It allows you to extract specific columns or fields from a text file or stream.
- It can be used to select specific characters, bytes, or fields from each line of a file or input stream, based on a delimiter.
- SYNTAX:

```
$ cut [OPTIONS] [FILENAME]
```

Examples

- To display 4th character of each line
- To display 1st to 4th character of each line
- To display from the 4th character to end of line.
- To split the line based on delimiter and display specific field.
- To split the line based on delimiter and display more than one fields like 1,2 and 6.

sort command

- It is used to sort the contents of a file or the output of another command.
- The `sort` command can be used to sort text files, log files, CSV files, and other types of data files
- By default, the `sort` command sorts the lines in alphabetical order
- Syntax :

```
$ sort [OPTIONS] [FILENAME]
```

Examples

- To sort the file in alphabetical order
- To sort the file and save results in new file.
- To sort in reverse order
- To check if file contents is in sorting order. It will return line number of unsorted text

Input Output redirection

- Input/output redirection is a way to redirect the input or output of a command or program to a file or device other than the default.
- This can be useful for saving output to a file, reading input from a file, or redirecting output to a different program or device.

Name	Symbol
Input redirection	<
Output redirection	>
Append	>>