

# Welcome Linux 2022

[www.adafrolabs.com](http://www.adafrolabs.com)

---

## **Piping**

The process of sending standard output from a command directly to another command's standard input for further processing is called **piping**. The symbol used to connect two commands together is the vertical bar (|), commonly referred to as **pipe**. You can pipe two or more commands together and a pipe is needed between each command pair, for example:

**command1 | command2 | command3**

In the above generic example the output of command1 is sent to input of command2 and output of command2 is sent to input of command3. Output of command3 (if any) appears in the terminal. The first command may use a file as input but it would be unusual for the subsequent commands to have input files. Conversely, the first command usually does not have an output file as the results are sent to the second command through the pipe.

You need to take care when typing pipeline commands - there is a space separating each command (and possibly parameters) and the pipe symbol.

- Make sure you still have the **friends** file from the other lab.
- Try the following pipeline commands. Observe the results and discuss them with a classmate.

**tree / | less**

**cut -d, -f4 friends | sort | uniq -cd**

**ls -A ~ | wc -l**

**ls -l /etc | cut -b2-10**

**grep Trish friends | cut -d, -f3-4**

---

**ls /etc | sort | head -12**

**head -5 friends | tail -2**

**echo This is interesting | cut -b 9-**

### **Redirection**

Output of a command with standard output (one that produces output to the terminal) can be redirected to a file. Because the outcome is different than in the piping scheme, a different symbol is used to redirect command output - we use ">" operator. This arrow-shaped operator literally shows in which direction the data flows. In simplest case, the arrow points from the command on the left to a file on the right, for example:

**ls -l > file.list**

In the above example, the **file.list** file will contain the detailed listing of your present working directory. If the file existed before, the redirection process overwrites it - any previous data is destroyed. It is possible to preserve the contents of existing files and **append** the new content to it, using the >> operator, for example:

**ls -l /etc >> file.list**

In the above example the listing of the /etc directory will appear below the previous command's output in file. list.

- Go ahead and try the above commands. Use the less file. list command to view the contents at each step.

It is possible to reverse the direction of the redirection symbol to have data flow from the file to the command, but this approach is used rarely and we will not cover it in this course.

It might be worth noting that just with piping, redirection concerns itself only with standard output. Any errors occurring during command execution will still appear in the terminal. You can redirect the errors to a file if you want using the "2>" (or "2>>") operator.

---

Try the following commands in the terminal. Use the cat command after each step to see the contents of the files used at the receiving end of the redirection. Some of the examples use a special system file - **/dev/null** (sometimes called "the bit bucket", which is a convenient location to use when you have some data to be discarded. As shown in a later example, /dev/null can be a very handy source of "nothing", useful when you want to quickly empty a file.

```
ls /blah-blah /
```

```
ls /blah-blah / > output.lst
```

```
ls /blah-blah / 2> error.log
```

```
ls /blah-blah / > output.lst 2> error.log
```

```
ls /blah-blah / > output.lst 2> /dev/null
```

```
echo Hello > hello.txt
```

```
echo Bye >> hello.txt
```

```
cat /dev/null > hello.txt
```

```
grep 1991 friends > 1991friends.txt
```

```
cut -d, -f1 friends | sort > friends.sorted
```

You can use piping and redirection in one large command line, but remember the following rules:

- Pipe symbol ( | ) appears only between commands
- Redirection symbols ( > or >> ) appear between command and a file-name
- Two files cannot be connected using a pipe or redirection, you need at least one command
- Redirection symbol acts like a sink. Once output is redirected there is no further data flowing - there is no point to continue with another redirection or a pipe after redirection.

- 
- Create a file called **companies** in your home directory with the following content inside.  
(You may want to copy and paste that from the on-line version of this document.)

Sem Ut Dolor Corp. , (271) 832-14474

Et Rutrum Eu Corporation, (686) 390-5070

Laoreet Ipsum LLP, (562) 574-2925

Fermentum Consulting, (918) 435-2072

Duis Incorporated, (837) 782-3061

Sit Amet Corp., (742) 441-5359

Congue Elit Sed Industries, (699) 827-2940

Facilisis Corp., (906) 814-2387

Ut Lacus Company, (138) 724-7139

- Figure out what commands (using filters and piping) would result in the following output in terminal:

Ut

Rutrum

Command #1: \_\_\_\_\_

906

138

Command #2: \_\_\_\_\_

---

Also, provide command lines using piping and redirection to accomplish the following:

- Store information about the number of records in **companies** file in the **phone.list**

Command #3: \_\_\_\_\_

- Append the entire contents of the **companies** file in reverse order to the **phone.list** file

Command #4: \_\_\_\_\_

- Find out which **companies** have an area code starting with a 9 and store just the company name in file called **area.9**

Command#5: \_\_\_\_\_