

Redirecting Input and Output

Append Redirect shell command

The `>>` shell command is used to redirect the standard output of the command on the left and append (add) it to the end of the file on the right.

```
# This command will append "Hello World!" to
greetings.txt
echo "Hello World!" >> greetings.txt
```

Pipe shell command

The `|` command is called a *pipe*. It is used to *pipe*, or transfer, the standard output from the command on its left into the standard input of the command on its right.

```
# First, echo "Hello World" will send Hello
World to the standard output.
# Next, pipe | will transfer the standard
output to the next command's standard input.
# Finally, wc -w will count the number of
words from its standard input, which is 2.
echo "Hello World" | wc -w
```

Redirecting Output

The `>` symbol is used to redirect output by taking the output from the command on the left and passing as input to the file on the right.

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

cat Display

The shell command `cat` displays the contents of one or more files to the terminal.

```
$ cat poem.txt
$ cat poem.txt kitties.txt
```

grep Search

The shell command `grep` is used to search files for lines that match a pattern and returns the results.

Various options can be specified along with the `grep` command to specify the search.

In the provided example, the lines in the file **names.txt** which contain “sonny” will be returned.

```
grep 'sonny' names.txt
```

Case insensitive search

The shell `grep` command searches files for a particular pattern. The `grep` command with the `-i` option can be used to search files for lines that match a pattern, case insensitive, and returns the results.

grep -R shell command

The shell command `grep` has a `-R` option (`grep -R`) that searches all files in a directory, including its subdirectories, and outputs filenames and lines containing matched results.

Command Line Redirection

On a command line, `redirection` is the process of using the input/output of a file or command to use it as an input for another file. It is similar but different from pipes, as it allows reading/writing from files instead of only commands.

Redirection can be done by using the operators `>` and `>>` .

```
ls > directories_list.txt  
ls >> directories_list.txt
```