

# Chapter 4 and 5 Review: Unix and Linux File Processing

## Input/Output Redirection Operators

- > Redirect standard output to a file: `cat test-file.txt > redirected-file-output.txt`
- >> Redirect standard output to a file appending to the end of the file: `cat test-file.txt >> appened-file.log`
- 2> Redirects standard error to a file: `rm full-dir 2> err.log`
- < Redirects the contents of a file to be used as the standard input of another command:
- | Sends the standard output of one command to the standard output of another command

## Basic File Manipulation

#creates a new file named my-file.txt:

```
> my-file.txt
```

#creates a new file named their-file.txt

```
touch their-file.txt
```

#removes a file named hello-there.txt from the current directory

```
rm hello-there.txt
```

#removes a directory named my-dir from the current directory regardless of whether or not the directory is full or empty

```
rm -r my-dir
```

#removes an empty directory named tom-folder

```
rmdir tom-folder
```

#copies a file named original.txt to poser.txt

```
cp original.txt poser.txt
```

#moves a file named homeland.txt from the current directory to /opt

```
mv homeland.txt /opt/homeland.txt
```

#renames a file named miss-ada.src to ms-ada.md

```
mv miss-ada.src ms-ada.md
```

#finds all the files that end with .log in the directory /var/log

```
find /var/log -name *.log
```

#combines the contents of file1 file2 and file3 into a single file named big-file.txt

```
cat file1 file2 file3 > big-file.txt
```

#makes two files, file1 and file2, a single file joined together horizontally named wide-file.txt

```
paste file1 file2 > wide-file.txt
```

#takes a vertical slice of file1 from columns 4-8 and stores it in the file col.csv

```
cut -b 4-8 file1 > col.csv
```

#sort the lines in the file qwerty.txt alphabetically and store them in the file abcdefg.txt

```
sort qwerty.txt > abcdefg.txt
```

#join the two files fruit.csv and bio.csv on the name of the fruit in columns 1 and 2 respectively:

```
join -1 1 -2 2 <(sort fruit.csv) <(sort -k 2 bio.csv)
```

#find lines that are the same in two files whose lines are mostly different

```
comm file1.txt file2.txt
```

#find lines that are different in two files whose lines are mostly the same

```
diff file1.txt file2.txt
```

#find the first line of all Hamlet's speaking parts in William Shakespeare's Hamlet marked by (Ham.)

```
grep < hamlet.txt "Ham\."
```

#count how many words are in hamlet.txt

```
wc -w hamlet.txt
```

#execute a script file name do-it.sh

```
sh do-it.sh
```

#executes a script file named do-it.sh with it's executable bit set

```
./do-it.sh
```

#executes a script named do-it.sh that is stored in one of the directories in the PATH variable

```
do-it.sh
```

#removes doubled lines from alphabetically sorted lines of the file, hamlet.txt

```
uniq <(sort hamlet.txt)
```

Our next unit is on scripting where we will look into advanced scripting and the subtleties of using \$, (), {}, ``, ' ', and " " effectively with all of the redirection operators