Chapter 10: Standard Text Streams and Redirection

Standard Out

- When a command executes without any errors, the output produced is known as standard out (also called **stdout** or **STDOUT**)
- By default stdout is sent to the terminal
- You can redirect stdout from a command into a file

Standard Out

- To redirect stdout of a command to a file, use the “>”:

```
$ls > /tmp/ls.txt
```

- A single > will override existing file contents
- Use >> to append to the end of a file
- Stdout is assign a numeric value of 1, so it could also be used like this:

```
$ls 1> /tmp/ls.txt
```

Standard Error

- When command encounters an error, it will produce output that is known as standard error (also called **stderr** or **STDERR**).
- The stderr output is sent to the terminal
- The number associated with the standard error file descriptor is 2.

Standard Error

To redirect stderr, use the following syntax:

```
$ls /junk 2> /tmp/output.err
```

- A single 2> will override existing file contents
- Use 2>> to append to the end of a file
- To discard output, redirect output to the `/dev/null` file (trash can)

Standard Error

- To redirect both stdout and stderr to different files:

```
$cmd > output 2> error
```

- To redirect both stdout and stderr to the same file, use one of the following:

```
$cmd > output 2>&1
$cmd &> output
```

Standard Input

- Standard in (also called **stdin** or **STDIN**) normally comes from the keyboard
- Some commands expect input from STDIN rather than a filename argument
- Sometimes you may want to redirect standard input so the input comes from a file instead of they keyboard

Standard Input

- Example:

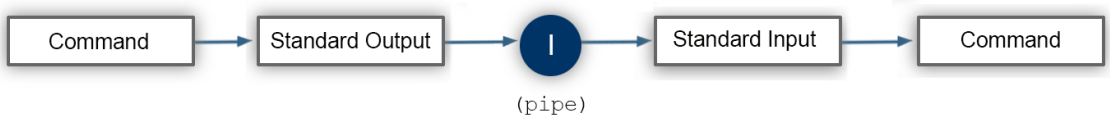
```
$tr 'a-z' 'A-Z' < ~/.bashrc
```

- Do not attempt to use the same file for input and output redirection (you end up losing all data)
- Instead, use different files:

```
$tr 'a-z' 'A-Z' < ~/.bashrc > output
```

Command Pipelines

- In a command pipeline, the output of one command is sent to another as input
- The "|" symbol is used between two commands to represent a command pipeline



- Examples:

```
$history|more
$ps -ef|grep head|sort +1|lp
```

The tee command

- The tee command duplicates output to be placed in a file and to the terminal simultaneously:

```
$date | tee timer.txt
Fri Nov 7 02:21:24 UTC 2014
$cat timer.txt
Fri Nov 7 02:21:24 UTC 2014
```

- The `-a` option is used to append output to an existing file