LAB TIME

Try Try Try ... Success

Examples for Output Redirection

Many routine administration tasks are simplified by using redirection. Use the previous table to assist while considering the following examples:

• Save a time stamp for later reference.

```
[student@servera ~]$ date > /tmp/saved-timestamp
```

• Copy the last 100 lines from a log file to another file.

```
[student@servera ~]$ tail -n 100 /var/log/dmesg > /tmp/last-100-boot-
messages
```

• Concatenate four files into one.

```
[student@servera ~]$ cat file1 file2 file3 file4 > /tmp/all-four-in-one
```

• List the home directory's hidden and regular file names into a file.

```
[student@servera ~]$ ls -a > /tmp/my-file-names
```

• Append output to an existing file.

```
[student@servera ~]$ echo "new line of information" >> /tmp/many-lines-of-
information

[student@servera ~]$ diff previous-file current-file >> /tmp/tracking-
changes-made
```

• The next few commands generate error messages because some system directories are inaccessible to normal users. Observe as the error messages are redirected. Redirect errors to a file while viewing normal command output on the terminal.

```
[student@servera ~]$ find /etc -name passwd 2> /tmp/errors
```

• Save process output and error messages to separate files.

```
[student@servera ~]$ find /etc -name passwd > /tmp/output 2> /tmp/errors
```

• Ignore and discard error messages.

```
[student@servera ~]$ find /etc -name passwd > /tmp/output 2> /dev/null
```

• Store output and generated errors together.

```
[student@servera ~]$ find /etc -name passwd &> /tmp/save-both
```

• Append output and generated errors to an existing file.

```
[student@servera ~]$ find /etc -name passwd >> /tmp/save-both 2>&1
```

Pipeline Examples

This example takes the output of the ls command and uses less to display it on the terminal one screen at a time.

```
[student@servera ~]$ ls -l /usr/bin | less
```

The output of the ls command is piped to wc -l, which counts the number of lines received from ls and prints that to the terminal.

```
[student@servera ~]$ ls | wc -l
```

In this pipeline, head will output the first 10 lines of output from ls -t, with the final result redirected to a file.

```
[student@servera ~]$ ls -t | head -n 10 > /tmp/ten-last-changed-files
```

Pipelines, Redirection, and the tee Command

When redirection is combined with a pipeline, the shell sets up the entire pipeline first, then it redirects input/output. If output redirection is used in the middle of a pipeline, the output will go to the file and not to the next command in the pipeline. In this example, the output of the ls command goes to the file, and less displays nothing on the terminal.

```
[student@servera ~]$ ls > /tmp/saved-output | less
```

Pipeline Examples Using the tee Command

This example redirects the output of the ls command to the file and passes it to less to be displayed on the terminal one screen at a time.

```
[student@servera ~]$ ls -l | tee /tmp/saved-output | less
```

If tee is used at the end of a pipeline, then the final output of a command can be saved and output to the terminal at the same time.

```
[student@servera ~]$ ls -t | head -n 10 | tee /tmp/ten-last-changed-files
```