

# ULI101: INTRODUCTION TO UNIX / LINUX AND THE INTERNET

## WEEK 5: LESSON 1

---

ADDITIONAL LINUX COMMANDS

REDIRECTION SYMBOLS

/DEV/NULL FILE , THE HERE DOCUMENT

PHOTOS AND ICONS USED IN THIS SLIDE SHOW ARE LICENSED UNDER [CC BY-SA](#)

# LESSON 5.1 TOPICS

---

## Redirection – Part I

- Additional Commands (**tr**, **cut**, **wc**)
- Concepts:
  - **Standard Input, Standard Output, Standard Error**
- Redirection Symbols: (<, >, >>, 2>, 2>>)
- Additional Redirection Concepts:
  - **/dev/null** File, The **Here Document**

## Perform Week 5 Tutorial

- Investigation I

# ADDITIONAL FILE MANIPULATION COMMANDS

---

## Additional Text File Manipulation Commands

Here are some additional commands to manipulate content of text files.

Command	Description
<code>tr</code>	Used to <b>translate</b> characters to different characters. eg. <code>tr 'a-z' 'A-Z' &lt; filename</code>
<code>cut</code>	Used to <b>extract</b> fields and characters from records. The option <code>-c</code> option is used to cut by a character or a range of characters. The <code>-f</code> option indicates the field number or field range to display (this may require using the <code>-d</code> option to indicate the field separator (delimiter) which is tab by default). eg. <code>cut -c1-5 filename</code> , <code>cut -d":" -f2 filename</code>
<code>wc</code>	Displays various <b>counts</b> of the contents of a file. The <code>-l</code> option displays the number of lines, the <code>-w</code> option displays the number of words, and the <code>-c</code> option displays the number of characters. eg. <code>wc filename</code> , <code>wc -l filename</code> , <code>wc -w filename</code>

# ADDITIONAL FILE MANIPULATION COMMANDS



---

## Instructor Demonstration

Your instructor will now demonstrate using the following Linux commands:

- **tr**
- **cut**
- **wc**

# REDIRECTION



---

**Redirection** can be defined as **changing** *from* where commands **read input** to where commands **send output**. You can redirect the input and output of a command.

For redirection, **meta characters** are used.

Redirection can be into a **file** (shell meta characters are angle **brackets** '<', '>') or a **program** (shell meta characters are **pipe** symbol '|').

Reference: <https://www.javatpoint.com/linux-input-output-redirection>

# REDIRECTION

---

**Standard input (stdin)** is a term which describes from where a command receives **input**.

The meta character “<” will redirect **stdin** into a command.

This would only apply to Unix/Linux commands that can **accept stdin** like **cat, more, less, sort, grep, uniq, head, tail, tr, cut, and wc**.

*Examples:*

```
tr 'a-z' 'A-Z' < words.txt
```

```
cat < abc.txt
```

```
sort < xyz.txt
```



command



filename

# REDIRECTION

---

**Standard output (stdout)** describes where a command sends its **output**.

The meta character “>” will redirect **stdout** to a file either **creating** a new file if it doesn’t exist or **overwriting** the content of an existing file.

The meta characters “>>” will redirect **stdout** to a file either **creating** a new file if it doesn’t exist or **adding** stdout to the **bottom** to the existing file’s contents.

*Examples:*

```
ls -l
```

```
ls -l > detailed-listing.txt
```

```
ls /bin >> output.txt
```



A diagram illustrating single redirection. It shows the word "command" in blue on the left, followed by a large black greater-than sign (>), and then the word "filename" in blue on the right. The entire diagram is enclosed in a blue rectangular border.



A diagram illustrating double redirection. It shows the word "command" in blue on the left, followed by two large black greater-than signs (>>), and then the word "filename" in blue on the right. The entire diagram is enclosed in a blue rectangular border.

# REDIRECTION

---

**Standard Error (stderr)** describes where a command sends its **error messages**.

The meta characters “**2>**” will redirect **stderr** to a file either **creating** a new file if it doesn't exist or **overwriting** the content of an existing file.

The meta characters “**2>>**” will redirect **stderr** to a file either **creating** a new file if it doesn't exist or **adding** stdout to the **bottom** to the existing file's contents.

*Examples:*

```
PWD
```

```
PWD 2> error-message.txt
```

```
PWD 2 >> error-messages.txt
```

```
PWD 2> /dev/null
```

```
command 2> filename
```

```
command 2>> filename
```



# REDIRECTION

---

The `/dev/null` file (sometimes called the **bit bucket** or **black hole**) is a special system file that **discards** stdout or stderr.

This is useful to “*throw-away*” **unwanted** command output or errors.

*Examples:*

```
ls 2> /dev/null
```

```
ls > /dev/null
```

```
find / -name "tempfile" 2> /dev/null
```



# REDIRECTION

---

The **Here Document** allows stdin to be redirected into a command **within** the command-line.

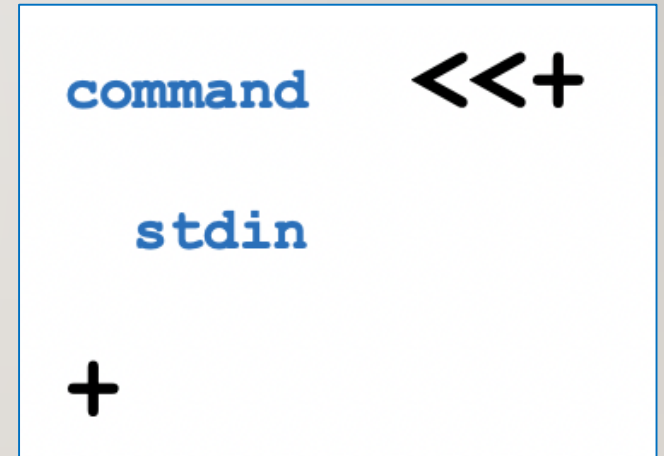
The meta characters “<<+” will redirect **stdin** into the command.

The **+** symbol is used to identify the beginning and ending of the stdin.

You can use ANY symbol or series of characters to mark stdin as long as that symbols or characters are **IDENTICAL** and the ending symbol or characters are on a **separate** line with only that symbol or characters.

*Example:*

```
cat <<+  
Line 1  
Line 2  
Line 3  
+
```



# REDIRECTION



---

## Instructor Demonstration

Your instructor will now demonstrate redirection:

- Standard Input
- Standard Output
- Standard Error
- Both Standard Output and Standard Error
- Both Standard Input and Standard Output
- Redirecting to **/dev/null**
- The **Here Document**

# HOMEWORK

---

## Getting Practice

Perform **Week 5 Tutorial**

(Due: Friday Week 6 @ midnight for a 2% grade):

- [INVESTIGATION I: BASICS OF REDIRECTION](#)
- [LINUX PRACTICE QUESTIONS](#) (Questions 1 – 4)