Redirections in Linux

In Linux, **redirection** refers to the process of changing the standard input, output, or error streams of commands. This allows you to control where the input for a command comes from and where its output goes. Here are the primary types of redirection in Linux:

1. Standard Output Redirection (>)

- **Purpose**: Redirects the output of a command to a file, overwriting the file if it already exists.
- **Usage**:

command > output.txt

Example:

echo "Hello, World!" > greeting.txt

This command writes "Hello, World!" to greeting.txt, creating the file if it doesn't exist or overwriting it if it does.

cat > ouput.txt

So whatever you will write after running this command, will be redirected and copied to the “file.txt”. This is standard output redirection.

2. Standard Output Appending (>>)

- **Purpose**: Redirects the output of a command to a file, appending the output to the file if it already exists.
- **Usage**:

command >> output.txt

Example:

echo "Another line" >> greeting.txt

This adds "Another line" to the end of greeting.txt.

3. Standard Input Redirection (<)

- **Purpose**: Redirects input from a file instead of the keyboard.
- **Usage**:

command < input.txt

Example:

sort < unsorted.txt

This sorts the contents of unsorted.txt and outputs the result to the terminal.

cat < file.txt

cat command will take the input from “file.txt” and print it to the terminal screen.

4. Here Documents (<<)

- **Purpose**: Allows you to create a multi-line string input for a command.
- **Usage**:

command << EOF

line1

line2

EOF

Example:

cat << EOF

This is a multi-line string.

It can contain multiple lines.

EOF

This will print the multi-line text to the terminal.

5. Here Strings (<<<)

- **Purpose**: Provides a string as standard input to a command.
- **Usage**:

command <<< "input string"

Example:

grep "pattern" <<< "This is a sample input string."

This will search for "pattern" in the provided string.

6. Pipes (|)

- **Purpose**: Passes the output of one command directly as input to another command.
- **Usage**:

command1 | command2

Example:

ps aux | grep "process_name"

This lists all processes and filters the output to show only those matching "process_name".

Redirection is a powerful feature in Linux that allows you to manage the input and output of commands effectively. By mastering redirection, you can automate tasks, log outputs, and manipulate data streams in flexible ways. Understanding these concepts will enhance your command-line skills and efficiency.