
SEEM3460 Tutorial

Advanced Linux Commands

Outline

- ❑ Input & Output redirection
- ❑ Metacharacters
- ❑ Pipes

Input&Output Redirection

- ❑ There are three default input/output (I/O) channels that are used by programs to interact with the environment. These channels are:
 - ❑ **Standard input (stdin):** This is the default input channel, which accepts input from the user or from another program.
 - ❑ **Standard output (stdout):** This is the default output channel, which displays the output of a program to the user or sends it to another program as input.
 - ❑ **Standard error (stderr):** This is the default error channel, which displays error messages or diagnostic information about the program's execution.
- ❑ These default I/O channels are an essential part of the Unix philosophy of "everything is a file", which means that all input/output operations in Unix-like operating systems are treated as file operations.

Input&Output Redirection

- ❑ There are three default input/output (I/O) channels that are used by programs to interact with the environment. These channels are:
 - ❑ **Standard input (stdin)**: This is the default input channel, which accepts input from the user or from another program.
 - ❑ **Standard output (stdout)**: This is the default output channel, which displays the output of a program to the user or sends it to another program as input.
 - ❑ **Standard error (stderr)**: This is the default error channel, which displays error messages or diagnostic information about the program's execution.
- ❑ By default, the **standard input** of a process is the **keyboard** and the **standard output** and standard error are the **screen**.
- ❑ **What if we want to change I/O channels?**

Input&Output Redirection

- ❑ Input redirection: use the contents of a file as input to a process
 - ❑ use < metacharacter
 - ❑ syntax: command < fileName
- ❑ Output redirection: store the output of a process to a file
 - ❑ use > or >> metacharacter
 - ❑ syntax: command > fileName

Input Redirection

- ❑ Input redirection: use the contents of a file as input to a process
 - ❑ use `<` metacharacter
 - ❑ syntax: `command < fileName`
- ❑ In Linux, the "cat" command is used to concatenate and display the contents of one or more files on the terminal
- ❑ Try the followings
 - ❑ `cat`
 - ❑ `cat < test.txt`
 - ❑ `cat test.txt`

Output Redirection

- ❑ Output redirection: store the output of a process to a file
 - ❑ use > or >> metacharacter
 - ❑ syntax: command > fileName
- ❑ In Linux, the "cat" command is used to concatenate and display the contents of one or more files on the terminal.
- ❑ Try the followings
 - ❑ cat test.txt
 - ❑ cat test.txt > test2.txt
 - ❑ cat test.txt >> test2.txt

Metacharacters

- ❑ These are the special characters that are first interpreted by the shell before passing them to the command. They are also known as shell wildcards

Symbol	Meaning
>	Output redirection; writes standard output to a file.
>>	Output redirection; appends standard output to a file.
<	Input redirection; reads standard input from a file.
*	File substitution wildcard; matches zero or more characters.
?	File substitution wildcard; matches any single character.
[...]	File substitution wildcard; matches any character between brackets.
`command`	Command substitution; replaced by the output from <i>command</i> .
	Pipe symbol; sends the output of one process to the input of another.

Metacharacters

- ❑ Backslash `\` prevents special interpretation of the next character.
- ❑ `echo "quotation"`
- ❑ `echo \"quotation\"`
- ❑ `echo hi \> file.txt`

Pipes

- ❑ In Linux, a pipe is a mechanism for connecting the output of one command to the input of another command. The pipe operator is represented by the vertical bar symbol "|".

- ❑ `command1 | command2`

- ❑ Here, the output of “command1” is passed as input to “command2”.

The pipe allows you to chain commands together and create complex data processing workflows.

- ❑ `cat test.txt | head -7`

- ❑ `cat test.txt | head -7 | tail -5`



Q&A