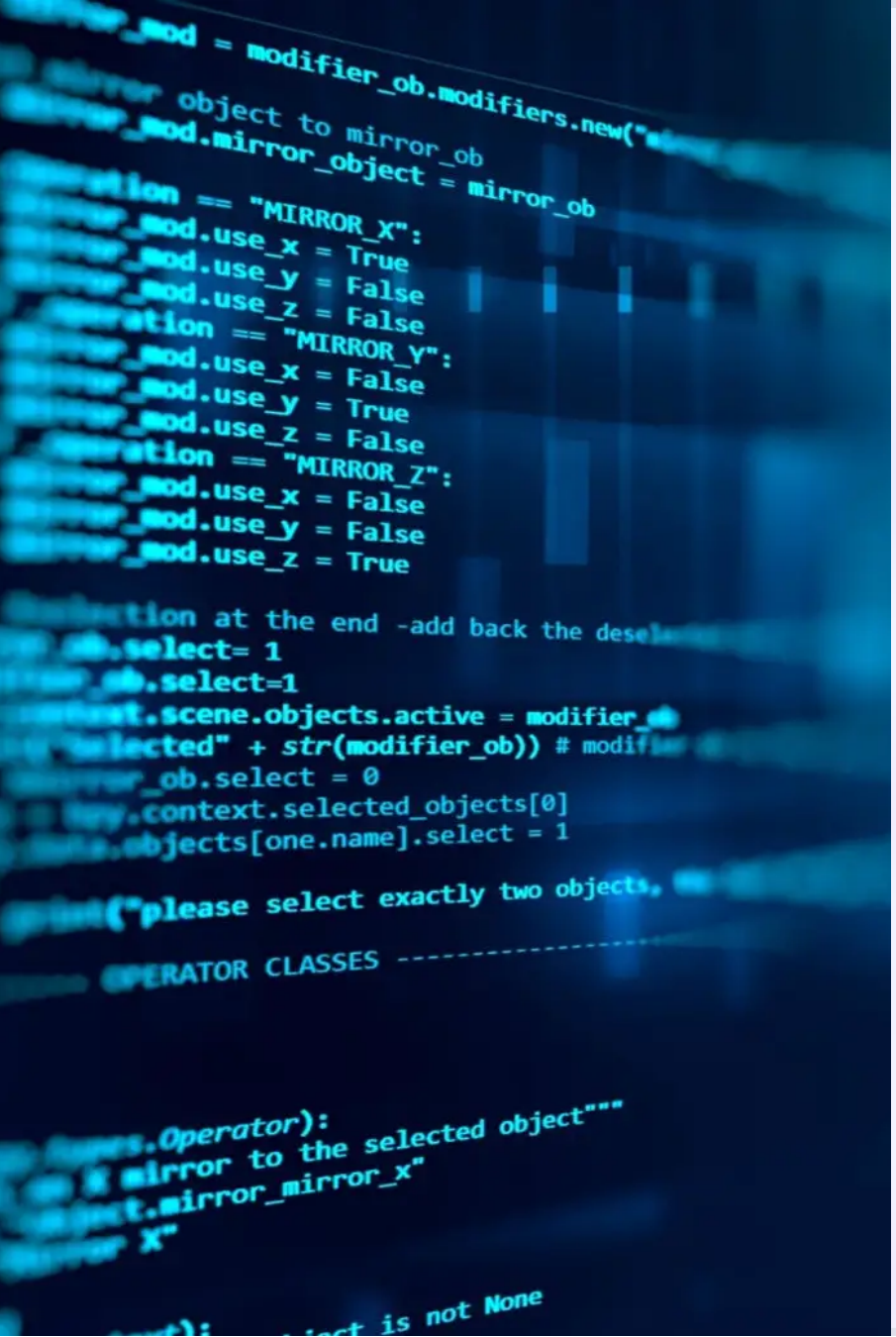


Cat, Sed, and Awk Commands: Powerful Tools for Data Manipulation

Welcome to this presentation exploring the cat, sed, and awk commands. Learn how to efficiently manipulate data in Unix/Linux environments.

A **by Abhilash Jadala**

The cat Command

Overview & Purpose

The cat command is used to concatenate and display the contents of files. It is a versatile tool for combining, creating, and displaying text files.

Example Uses & Common Options

Use cat to display the contents of a file, create and append to files, or even combine multiple files together. Common options include -n for numbering lines and -s for squeezing blank lines.

Pro Tip

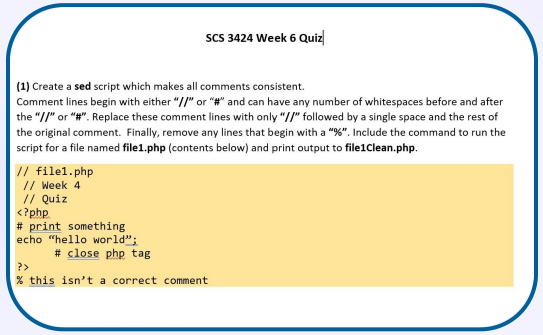
Redirect the output of cat to another command or a file using the ">" symbol. For example, "cat file.txt > newfile.txt" will create a new file with the contents of file.txt.

	non blank lines
cat -n	This is used to add line numbers to all lines
cat -s	This is used to squeeze blank lines into one line

The sed Command

Overview & Purpose

The sed command (stream editor) is a powerful tool for editing files without changing the original content. It is useful for tasks such as find and replace, inserting/deleting lines, and more.



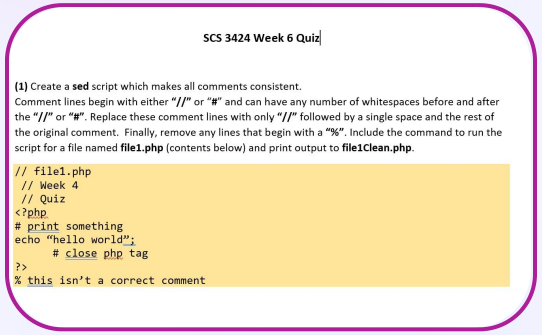
SCS 3424 Week 6 Quiz

(1) Create a **sed** script which makes all comments consistent.
Comment lines begin with either **"//"** or "**#**" and can have any number of whitespaces before and after the **"//"** or **"#"**. Replace these comment lines with only **"//"** followed by a single space and the rest of the original comment. Finally, remove any lines that begin with a **"%"**. Include the command to run the script for a file named **file1.php** (contents below) and print output to **file1Clean.php**.

```
// file1.php
 // Week 4
 // Quiz
<?php
# print something
echo "hello world";
      # close php tag
?>
% this isn't a correct comment
```

Example Uses & Common Options

Use sed to substitute specific patterns, delete lines based on conditions, or even perform complex text transformations using regular expressions. Common options include s for substitution and d for delete.



SCS 3424 Week 6 Quiz

(1) Create a **sed** script which makes all comments consistent.
Comment lines begin with either **"//"** or "**#**" and can have any number of whitespaces before and after the **"//"** or **"#"**. Replace these comment lines with only **"//"** followed by a single space and the rest of the original comment. Finally, remove any lines that begin with a **"%"**. Include the command to run the script for a file named **file1.php** (contents below) and print output to **file1Clean.php**.

```
// file1.php
 // Week 4
 // Quiz
<?php
# print something
echo "hello world";
      # close php tag
?>
% this isn't a correct comment
```

Pro Tip

Store your sed commands in a script file and use the -f option to execute the script. This allows for easier maintenance and reuse of complex sed operations.

The below substitution command is divided into five parts:

s ==> It specifies the substitution command

/ ==> It specifies the delimiters

Linux ==> It specifies the search pattern (Regular expression)

Unix ==> It specifies the replacement string.

g ==> It specifies the global replacement

Options	Description
-i	This option can be used to edit the source f
-e	This option can be used to combine multipl commands using a single call
-n	This option can be used to suppress the de behavior of printing the output
-f	This option can be used to execute the scri priority basis
Flags	
g	This flag is mainly used to replace all the o a string in a file globally.
i	It is used to ignore case-sensitive issues wi substitutions while replacing
p	It allows us to print the current pattern spac
w	It is used to write the present pattern space
x	It is used to exchanges the pattern space a
Character	
^	It acts as a special character and matches a beginning of the regular expression/pattern
$	It also works the same as ^ but, the only diff matches at the end of the regular expressio space.
.	It matches any single character, even inclu line.
*	It matches the sequence of zero or more ins previous character.

$ sed 's/Linux/Unix/' linuxteck.txt

How to search and replace all the strings/patterns without opening a file?Output:

Unix is a free and open-source operating system. Linux is a cross-platform operating system that runs on many computer models.
Unix has attractive features and performance. Linux is a graphical user interface.
Unix operating systems are more secure than Windows operating systems. Linux is always a multi-user operating system.

$ sed 's/Linux/Unix/2' linuxteck.txt

How to replace only the second occurrence on each line?

Output:

Linux is a free and open-source operating system. ***Unix*** is a cross-platform operating system that runs on many computer models.
Linux has attractive features and performance. ***Unix*** is a graphical user interface.
Linux operating systems are more secure than Windows operating system. ***Unix*** is always multi-user operating systems.

$ sed 's/Linux/Unix/g' linuxteck.tx

How to replace all occurrences of a string/pattern in a file?

1. Output:

   Unix is a free and open-source operating system. ***Unix*** is a cross-platform operating system that runs on many computer models.
   Unix has attractive features and performance. ***Unix*** is a graphical user interface.
   Unix operating systems are more secure than Windows operating systems. ***Unix*** is always a multi-user operating system.

$ sed -n p linuxteck.txt

How to print/display a series of lines from a file?

Output:

1. Linux is a free and open-source operating system.
2. Linux is a cross-platform operating system that runs on many computer models.
3. Linux has attractive features and performance.
4. Linux is a graphical user interface.
5. Linux operating systems are more secure than Windows operating systems.
6. Linux is always a multi-user operating system.

Deleting a line using sed command

To delete a line with sed from a file, use the following command,

```
$ sed Nd testfile.txt
```

where ‘N’ is the line number & option ‘d’ will delete the mentioned line number. To delete the last line of the file, use

```
$ sed $d testfile.txt
```

The awk Command

Overview & Purpose

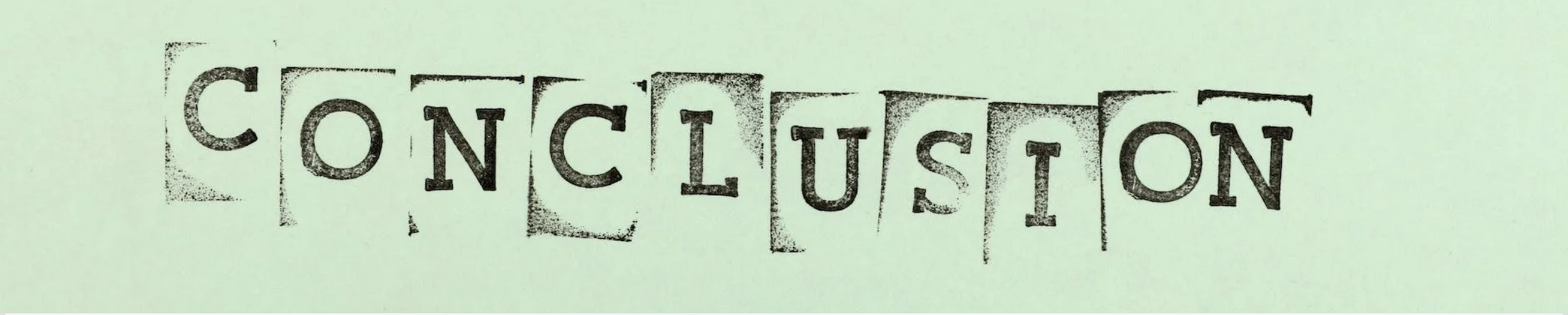
The awk command is a versatile programming language designed for text processing and data extraction. It excels at field-based operations, calculations, and generating reports.

Example Uses & Common Options

Use awk to extract specific fields from files, aggregate data, perform calculations, and apply conditional statements. Common options include -F to specify a custom field separator and -v to assign variables.

Pro Tip

Utilize the power of awk's built-in functions and associative arrays to handle more complex data processing tasks. Explore the extensive documentation to unlock its full potential.



Conclusion

The sed, awk, and grep commands are powerful tools for data manipulation and text processing in the Linux/Unix environment.

The sed command is primarily used for performing substitution and text transformations. It allows you to search for specific patterns and replace them with desired content.

The awk command is a versatile tool for processing and analyzing text files. It provides a wide range of functionalities, including pattern matching, field extraction, and data manipulation.

The grep command is used for searching and filtering text based on patterns. It allows you to find specific lines that match a given pattern or regular expression.

By mastering these commands, you can efficiently manipulate and extract useful information from large datasets, automate tasks, and streamline your workflow in the Linux/Unix environment.