

Bash Script Examples

In this presentation, we will explore two powerful Bash scripts that can help you manipulate and process text files. Whether you need to display lines within a specific range or delete lines with a particular word, these scripts will simplify your tasks and streamline your workflow.

 by hemanth m

```
fa.wikipedia.org
g (208.80.152.2) 56(84) bytes of data.

    ping statistics ---
    received, 0% packet loss, time 0ms
28/540.528/540.528/0.000 ms

5 Jul 30 22:43 .
5 Sep 14 20:42 ..
5 May 14 00:15 account
5 Jul 31 22:26 cache
5 May 18 16:03 db
5 May 18 16:03 empty
5 May 18 16:03 games
5 Jun 2 18:39 gdm
5 May 18 16:03 lib
5 May 18 16:03 local
1 May 14 00:12 lock -> ../run/lock
5 Sep 14 20:42 log
9 Jul 30 22:43 mail -> spool/mail
5 May 18 16:03 nis
5 May 18 16:03 opt
5 May 18 16:03 preserve
5 Jul 1 22:11 report
5 May 14 00:12 run -> ../run
5 May 18 16:03 spool
5 Sep 12 23:50 tmp
5 May 18 16:03 yp
arch wiki
resto, refresh-packagekit, remove-with-leaves
ry_db
```



Script 1: Displaying Lines

Often, we need to extract specific lines from a text file for analysis or further processing. With Script 1, you can effortlessly display lines within a defined range. Let's take a look at the code:

```
```
#!/bin/bash

Prompt the user for the file name
echo "Enter the file name:"

read name

Prompt the user for the first line to start
echo "Enter 1st line to start:"

read one

Prompt the user for the last line to end
echo "Enter last line to exit:"

read two

Use awk to display lines within the specified range
awk "NR >= $one && NR <= $two" "$name"

````
```

Example for Script 1

Let's say we have a file named 'example.txt' with the following content:

text file:

Line 1

Line 2

Line 3

Line 4

Line 5

Script 2: Deleting Lines with a Word

Script 2 allows you to quickly delete lines that contain a specific word in one or multiple files. This can be particularly useful for filtering out unnecessary or sensitive information. Take a look at the code:

```
```
#!/bin/bash

Prompt the user for the total number of files to process
echo "Enter number of files:"

read n

Prompt the user for the word to search for and delete
echo "Enter word to delete:"

read word

Loop through the specified number of files
for ((i=1;i<= $n; i++)); do

 echo "Enter file name:"

 read file

 # Display the file contents before removing the word
 echo "File before removing '$word':"

 cat "$file"

 # Use grep to remove lines containing the specified word
 grep -v -i "$word" "$file" > test

 # Rename the 'test' file back to the original file name
 mv test "$file"

 # Display the file contents after removing the word
 echo "File after removing '$word':"

 cat "$file"

done

```

```

Example for Script 2

Let's consider we have two files named 'file1.txt' and 'file2.txt' with the following contents:

file1.txt:

This is an example.

Please remove this line.

Another example here.

file2.txt:

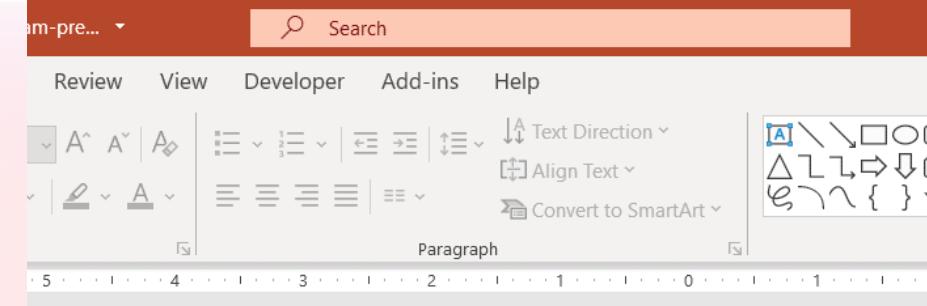
Hello, world!

This line contains no 'example' word.

More text.

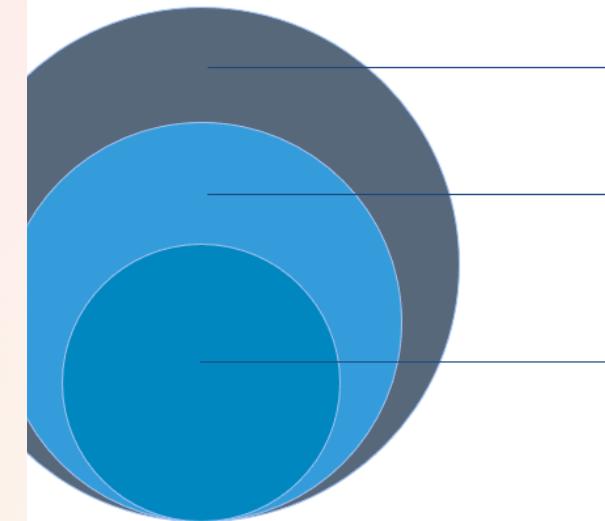
Create an Informative Presentation

Now that you have familiarized yourself with these efficient Bash scripts, it's time to create an impressive and engaging presentation. Utilize creative and eye-catching layouts like images, columns, and timelines to captivate your audience. Make use of visually stunning presentation templates to leave a lasting impression. Combine your newly acquired script skills with your presentation prowess to deliver an exceptional experience!



MARKET SIZE

This is a subtitle that you can edit



TAM Total Available

Example total market

SAM Serviceable Market

Example serviceable market

SOM Serviceable Opportunity Market

Example serviceable opportunity market

This is a text placeholder that you can edit. This is a text placeholder that you can edit. This is a text placeholder that you can edit. This is a text placeholder that you can edit.