

# Breaking Down a Shell Program for Reading Files by Line

Learn how to create a shell program that accepts a filename and line numbers as inputs, and returns the lines between the given line numbers. Explore the code and examples.

A blurred screenshot of a terminal window displaying a shell script. The script uses various functions like `use\_array`, `sort`, and `read\_file` to process input arrays and files. It includes regular expression matching for line endings and conditional logic for line selection. The code is color-coded for readability.

# Introduction

In this presentation, we will discuss the creation of a shell program that reads a file and outputs the lines between the given line numbers. This is a fundamental task in programming, and understanding how to do this will help you understand more complex programs.

# Explanation of the Shell Program

A shell program is a script that is interpreted by a command-line interpreter or terminal emulator. The shell program we will create accepts a filename and two line numbers as arguments, thus allowing us to display the content of the file from line X to line Y.

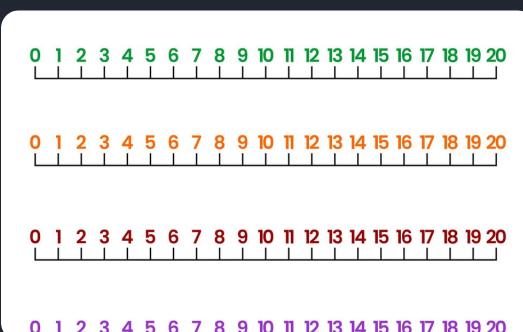


# Accepting Filename and Line Numbers as Arguments

The shell program accepts three arguments from the user: the filename, the starting line number, and the ending line number. The arguments are read from left to right using the \$1, \$2, and \$3 variables, respectively. The program then checks if the file exists and is readable.

# Reading the File and Displaying the Lines

```
steve@ubuntu:~/Dev/Code/Python  
steve@ubuntu:~/Dev/Code/Python$ ./ListDir.sh  
Welcome  
Desktop Downloads ListDir.sh Pictures Templates  
Documents examples.desktop Music Public Videos  
This completes the list of directories  
steve@ubuntu:~/Dev/Code/Python$
```

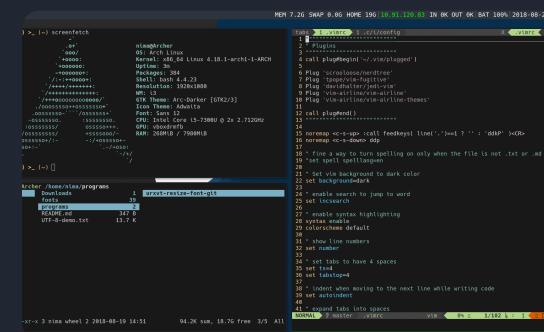


## Reading the File

Once the file has been validated, the shell program reads the file using the `read` command, which is used to read a single line from standard input.

## Displaying the Lines

The shell program then uses the `sed` command to display only the lines between the starting and ending line numbers.



## The Output

The output shows the lines between the given starting and ending line numbers, returned in the order they were found in the file.

# Error Handling for Invalid Inputs and Code

## Error Checking

The shell program includes checks for invalid filename or line numbers and provides an error message or prompt to the user.

## Clean Code

The program follows best practices such as proper indentation and comments to make its logic easy to read and understand.



# Example Usage and Demonstration

Invocation  
Usage example:  
read\_file\_between\_lines.sh filename.txt  
startingline endingline

1

2

3

## Sample Code

Here is a sample shell program that can be used for reading files by line:

## Output Example

The output shows the lines between the given starting and ending line numbers, returned in the order they were found in the file.



# Conclusion and Closing Remarks

## 1 What You've Learned

Through this presentation, you learned how to create a shell program that reads a file and outputs the lines between two given line numbers.

## 2 Practical Applications

This skill can be applied to a wide range of programming tasks, making it a powerful tool in your arsenal as a developer.

## 3 Exploring Further

Taking it further, you can expand your knowledge in shell scripting by learning about other Unix utilities, using variables, writing loops and conditionals, and creating custom functions.