

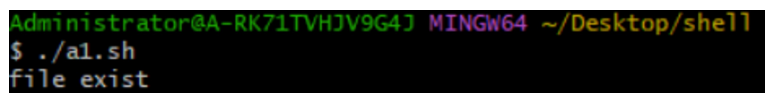
Day6Assignments

12 May 2024 10:58 AM

Assignment 1: Ensure the script checks if a specific file (e.g., myfile.txt) exists in the current directory. If it exists, print "File exists", otherwise print "File not found".

script:

```
if [ -f "myfile.txt" ]; then
    echo "File exists"
else
    echo "File not found"
fi
```

A terminal window with a black background and green text. The prompt is 'Administrator@A-RK71TVHJV9G4J MINGW64 ~/Desktop/shell'. The user enters '\$./a1.sh' and the output is 'file exist'.

```
Administrator@A-RK71TVHJV9G4J MINGW64 ~/Desktop/shell
$ ./a1.sh
file exist
```

Assignment 2: Write a script that reads numbers from the user until they enter '0'. The script should also print whether each number is odd or even.

```
#!/bin/bash

echo "Enter numbers (enter 0 to stop):"

while :
do
    read -p "Number: " num

    if [ "$num" -eq 0 ]; then
        echo "Exiting..."
        break
    fi

    if [ $((num % 2)) -eq 0 ]; then
        echo "$num is even"
    else
        echo "$num is odd"
    fi
done
```

Assignment 3: Create a function that takes a filename as an argument and prints the number of lines in the file. Call this function from your script with different filenames.

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

```

count_lines() {
    local file="$1" # Filename passed as argument
    if [ -f "$file" ]; then
        local num_lines=$(wc -l < "$file")
        echo "Number of lines in $file: $num_lines"
    else
        echo "Error: $file not found or is not a regular file"
    fi
}

count_lines "file1.txt"
count_lines "file2.txt"
count_lines "nonexistent_file.txt"

```

Assignment 4: Write a script that creates a directory named TestDir and inside it, creates ten files named File1.txt, File2.txt, ... File10.txt. Each file should contain its filename as its content (e.g., File1.txt contains "File1.txt").

```

#!/bin/bash

mkdir -p TestDir

cd TestDir || exit

for ((i = 1; i <= 10; i++)); do
    filename="File${i}.txt"
    echo "$filename" > "$filename"
done

echo "Files created successfully in TestDir"

```

Assignment 5: Modify the script to handle errors, such as the directory already existing or lacking permissions to create files. Add a debugging mode that prints additional information when enabled.

```

#!/bin/bash

debug() {
    if [ "$DEBUG" = true ]; then
        echo "[DEBUG] $1"
    fi
}

if [ -d "TestDir" ]; then
    echo "Error: Directory 'TestDir' already exists. Aborting."

```

```

    exit 1
fi

mkdir -p TestDir
if [ $? -ne 0 ]; then
    echo "Error: Failed to create directory 'TestDir'. Aborting."
    exit 1
fi
echo "Directory 'TestDir' created successfully."

cd TestDir || {
    echo "Error: Failed to change directory to 'TestDir'. Aborting."
    exit 1
}

for ((i = 1; i <= 10; i++)); do
    filename="File${i}.txt"
    debug "Creating file: $filename"
    echo "$filename" > "$filename"
    if [ $? -ne 0 ]; then
        echo "Error: Failed to create file '$filename'. Skipping."
    else
        echo "File '$filename' created successfully."
    fi
done

echo "Files created successfully in TestDir"

```

Assignment 6: Given a sample log file, write a script using grep to extract all lines containing "ERROR". Use awk to print the date, time, and error message of each extracted line.

Data Processing with sed

```

#!/bin/bash

grep "ERROR" log.txt | \

awk -F ' - ' '{print $1, $2, $3}' | \
sed -E 's/^([0-9]{4}-[0-9]{2}-[0-9]{2}) ([0-9]{2}:[0-9]{2}:[0-9]{2}) - (ERROR - .*)/\1 \2 \3/'

```

```

2024-05-10 10:20:45 Database connection failed
2024-05-10 11:05:12 File not found
2024-05-10 14:00:02 Server crashed unexpectedly

```

Assignment 7: Create a script that takes a text file and replaces all occurrences of "old_text" with "new_text". Use sed to perform this operation and output the result to a new file.

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

```
if [ "$#" -ne 3 ]; then  
    echo "Usage: $0 input_file old_text new_text"  
    exit 1  
fi
```

```
input_file="$1"  
old_text="$2"  
new_text="$3"  
output_file="${input_file%.txt}_modified.txt" # Output filename based on input filename
```

```
if [ ! -f "$input_file" ]; then  
    echo "Error: Input file '$input_file' not found."  
    exit 1  
fi
```

```
sed "s/$old_text/$new_text/g" "$input_file" > "$output_file"
```

```
echo "Text replacement completed."  
echo "Output written to '$output_file'."
```