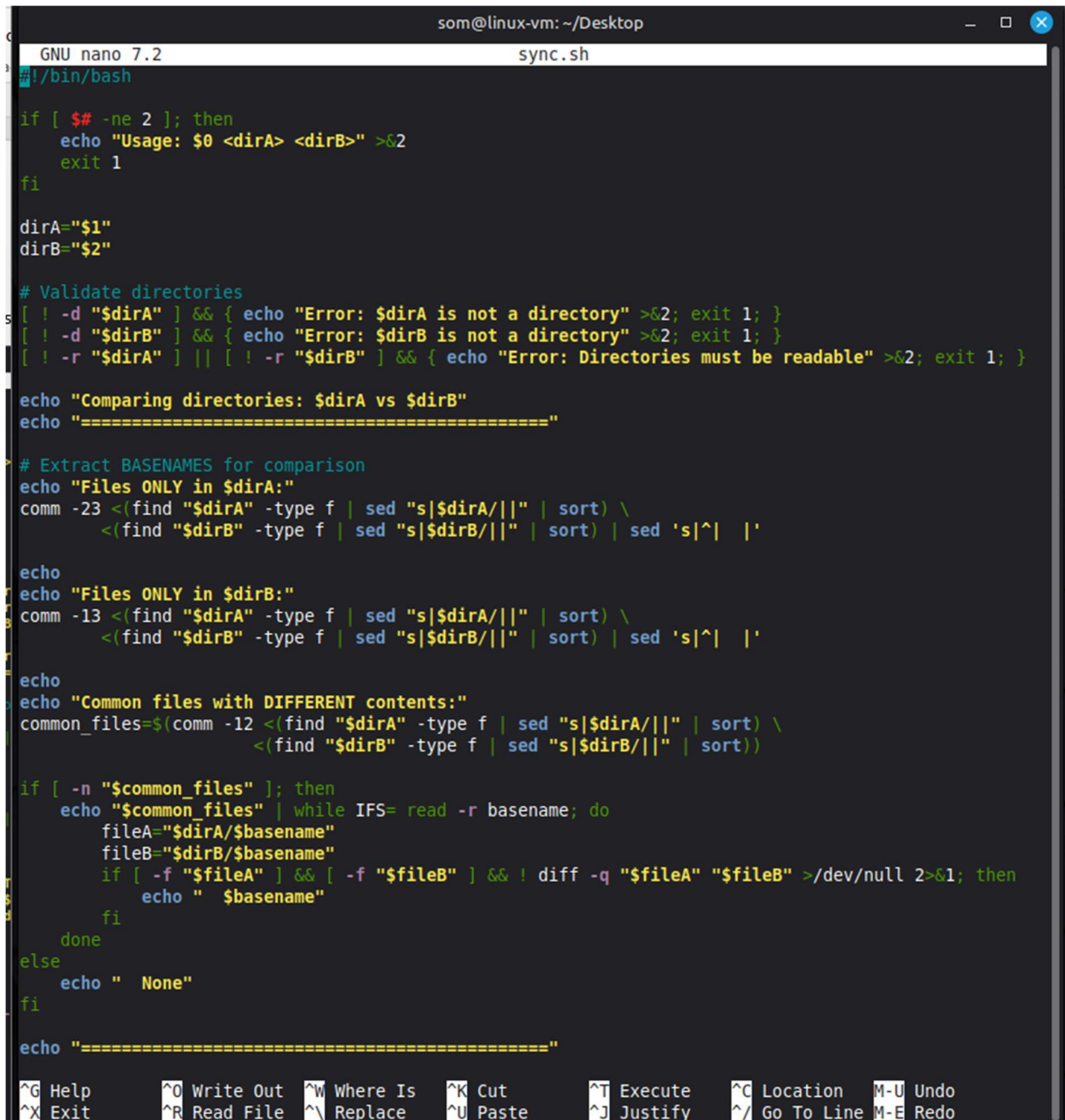


Question 5 (2024eb03003):

Please find screenshot of shell script below and attaching **sync.sh** script to GitHub repository:



```
GNU nano 7.2                                sync.sh
#!/bin/bash

if [ $# -ne 2 ]; then
    echo "Usage: $0 <dirA> <dirB>" >&2
    exit 1
fi

dirA="$1"
dirB="$2"

# Validate directories
[ ! -d "$dirA" ] && { echo "Error: $dirA is not a directory" >&2; exit 1; }
[ ! -d "$dirB" ] && { echo "Error: $dirB is not a directory" >&2; exit 1; }
[ ! -r "$dirA" ] || [ ! -r "$dirB" ] && { echo "Error: Directories must be readable" >&2; exit 1; }

echo "Comparing directories: $dirA vs $dirB"
echo "===== "

# Extract BASENAMES for comparison
echo "Files ONLY in $dirA:"
comm -23 <(find "$dirA" -type f | sed "s|$dirA/||" | sort) \
    <(find "$dirB" -type f | sed "s|$dirB/||" | sort) | sed 's|^| |'

echo
echo "Files ONLY in $dirB:"
comm -13 <(find "$dirA" -type f | sed "s|$dirA/||" | sort) \
    <(find "$dirB" -type f | sed "s|$dirB/||" | sort) | sed 's|^| |'

echo
echo "Common files with DIFFERENT contents:"
common_files=$(comm -12 <(find "$dirA" -type f | sed "s|$dirA/||" | sort) \
    <(find "$dirB" -type f | sed "s|$dirB/||" | sort))

if [ -n "$common_files" ]; then
    echo "$common_files" | while IFS= read -r basename; do
        fileA="$dirA/$basename"
        fileB="$dirB/$basename"
        if [ -f "$fileA" ] && [ -f "$fileB" ] && ! diff -q "$fileA" "$fileB" >/dev/null 2>&1; then
            echo "  $basename"
        fi
    done
else
    echo "  None"
fi

echo "===== "
```

Help Write Out Where Is Cut Execute Location M-U Undo
Exit Read File Replace Paste Justify Go To Line M-E Redo

Testing the `sync.sh` Script

Create directories:

```
mkdir -p dirA dirB
```

dirA contents

```
echo "Content A" > dirA/file1.txt
echo "Different" > dirA/file2.txt
echo "Line1" > dirA/file3.txt
echo "Only in A" > dirA/uniqueA.txt
```

dirB contents

```
echo "Content A" > dirB/file1.txt
echo "Same here" > dirB/file2.txt
echo "Line1\nLine2" > dirB/file3.txt
echo "Only in B" > dirB/uniqueB.txt
```

```
som@linux-vm: ~/Desktop
som@linux-vm:~/Desktop$ mkdir -p dirA dirB
som@linux-vm:~/Desktop$ echo "Content A" > dirA/file1.txt
som@linux-vm:~/Desktop$ echo "Different" > dirA/file2.txt
som@linux-vm:~/Desktop$ echo "Line1" > dirA/file3.txt
som@linux-vm:~/Desktop$ echo "Only in A" > dirA/uniqueA.txt
som@linux-vm:~/Desktop$
som@linux-vm:~/Desktop$ echo "Content A" > dirB/file1.txt
som@linux-vm:~/Desktop$ echo "Same here" > dirB/file2.txt
som@linux-vm:~/Desktop$ echo "Line1\nLine2" > dirB/file3.txt
som@linux-vm:~/Desktop$ echo "Only in B" > dirB/uniqueB.txt
som@linux-vm:~/Desktop$
```

Test Case:

```
./sync.sh dirA dirB
```

```
som@linux-vm: ~/Desktop
som@linux-vm:~/Desktop$ ./sync.sh dirA dirB
Comparing directories: dirA vs dirB
=====
Files ONLY in dirA:
    uniqueA.txt

Files ONLY in dirB:
    uniqueB.txt

Common files with DIFFERENT contents:
    file2.txt
    file3.txt
=====
som@linux-vm:~/Desktop$
```

Requirement validation:

Created test directories dirA and dirB with matching (file1.txt), differing (file2.txt, file3.txt), and unique files (uniqueA.txt, uniqueB.txt), then ran `./sync.sh dirA dirB` to verify it correctly listed files only in each directory by basename and identified content differences using `diff -q` on common files, confirming all three requirements without file modification.

- Lists files by basename only (Req 1, 2)
- Finds common files correctly (Req 3)
- Compares content with `diff -q` (Req 3)
- No copying/modification (Req 3)