

Question 8 (2024eb03003):

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

```
som@linux-vm: ~/Desktop
GNU nano 7.2      bg_move.sh
#!/bin/bash

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

DIR="$1"

# Validate directory exists and is writable
if [ ! -d "$DIR" ]; then
    echo "Error: $DIR is not a directory" >&2
    exit 1
fi

if [ ! -w "$DIR" ]; then
    echo "Error: $DIR is not writable" >&2
    exit 1
fi

# Create backup subdirectory if it doesn't exist
mkdir -p "$DIR/backup"

# Counter for process tracking
pid_count=0
pids=()

echo "Moving files from $DIR to $DIR/backup in background..."
echo "Script PID: $$"

# Loop through regular files and move in background
for file in "$DIR"/*; do
    # Skip if not a regular file or backup dir
    [ ! -f "$file" ] && continue
    [ "$(basename "$file")" = "backup" ] && continue

    pid_count=$((pid_count + 1))
    echo "Starting background move #${pid_count}: $(basename "$file")"

    # Move in background, capture PID with !
    mv "$file" "$DIR/backup/" &
    pids[$pid_count]=$!
    echo "  PID: ${pids[$pid_count]}"
done

echo
echo "Started $pid_count background processes. Waiting for completion..."

# Wait for ALL background processes
for pid in "${pids[@]}; do
    if [ -n "$pid" ]; then
        wait "$pid"
        echo "Process PID $pid completed"
    fi
done

echo "All background moves completed!"
echo "Backup directory: $DIR/backup"
ls -l "$DIR/backup/"

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute  ^C Location  M-U Undo
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify  ^_ Go To Line M-E Redo
```

Testing the bg_move.sh Script

Test Case:

Create testdir with multiple files, run ./bg_move.sh to verify each mv runs in background (&), displays PIDs (\$!), creates backup/ subdirectory, and wait blocks until all processes complete (\$\$ shown), proving all requirements met.

```
mkdir testdir && cd testdir
```

```
touch file1.txt file2.txt file3.log
```

```
chmod +x ../bg_move.sh
```

```
../bg_move.sh .
```

```
som@linux-vm: ~/Desktop/testdir
som@linux-vm:~/Desktop$ mkdir testdir && cd testdir
som@linux-vm:~/Desktop/testdir$ touch file1.txt file2.txt file3.log
som@linux-vm:~/Desktop/testdir$ chmod +x ../bg_move.sh
som@linux-vm:~/Desktop/testdir$ ../bg_move.sh .
Moving files from . to ./backup in background...
Script PID: 7676
Starting background move #1: file1.txt
  PID: 7680
Starting background move #2: file2.txt
  PID: 7683
Starting background move #3: file3.log
  PID: 7686

Started 3 background processes. Waiting for completion...
Process PID 7680 completed
Process PID 7683 completed
Process PID 7686 completed
All background moves completed!
Backup directory: ./backup
total 0
-rw-rw-r-- 1 som som 0 Feb  1 19:41 file1.txt
-rw-rw-r-- 1 som som 0 Feb  1 19:41 file2.txt
-rw-rw-r-- 1 som som 0 Feb  1 19:41 file3.log
som@linux-vm:~/Desktop/testdir$
```

Requirement followed:

- &: Each mv runs in background
- \$!: Captures PID of last background process
- \$\$: Displays script's own PID
- wait \$pid: Waits for specific background PIDs to finish
- Array \${pids[@]}: Tracks all background process PIDs

The script created backup/ subdirectory and moves all regular files there concurrently, displays each PID, and waits for complete completion.