

ICCS121 System Skills & Low-Level Programming:

[Ex3] Shell Scripting

Due on Sunday, September 28, 2025, 11:59 pm (Thai time)

Songpon Teerakanok (Section 2)

Jiraroj Wiruchpongsanon (6781617)

File Organizer: Basic Folder Automation

Students will write a bash script to organize files by extension into corresponding folders.

Tasks:

1. Setup:
 - Create a folder named `unorganized/` and add 10 files of various types: i.e., `.txt`, `.jpg`, `.pdf`.
 - Create and name 3 additional folders under the same parent directory: `images/`, `documents/`, and `texts/`.
2. Scripting: Create a script to move each file in `unorganized/` to its respective folder.

Script Requirements:

- Loop through files in `unorganized/`.
- Create folders for each file extension if they don't exist.
- Move files to the correct folder.
- For each file moved, print a one-line log on the screen. The log should look like the following: "Moving <filename> to <foldername>/"

Sample screen output:

```
=====
Moving report.pdf to documents/
Moving image1.jpg to images/
Moving file1.txt to texts/
Organization Complete!
```



```
=====
```

Note: The script must actually move the files to their respective folders, not just print the log.

Deliverable:

Students must submit a single PDF file containing the following:

1. Bash Script (`setup.sh`):

- Paste the full script into the PDF.
- Ensure the script includes comments explaining each step.

Solution

Listing 1: `organize.sh`

```
#!/bin/zsh

# Setup:
# Create a folder named unorganized/ and add 10 files of various types: i.e.,
# .txt, .jpg, .pdf.
# Create and name 3 additional folders under the same parent directory: images
# /, documents/, and texts/.

# create `unorganized` directory
mkdir ./unorganized/

# array of various file types
extensions=(txt log webp joblib pdf jpg wav arw tex csv json)

# loop through `extensions` and create files
for extension in $extensions; do
    touch "./unorganized/file.$extension"
done

# create `images/`, `documents/` and `texts/` directories as instructed
mkdir ./images/
mkdir ./documents/
mkdir ./texts/
```

Listing 2: `organize.sh`

```
#!/bin/zsh

# Scripting: Create a script to move each file in unorganized/ to its
# respective folder.

# image extensions array
image_extensions=(png jpg jpeg gif bmp tiff tif webp heic heif svg ico avif)

# text extensions array
text_extensions=(txt text log md markdown rtf tex rst cfg conf ini yaml yml
json toml)

# document extensions array
document_extensions=(pdf doc docx odt rtf csv tsv xls xlsx ods ppt pptx odp)

# Can't tolerate O(n) lookup time, so we change to hashmap
typeset -A extensions_map=()
```

```

# map extensions to image
for extension in $image_extensions; do
    extensions_map[$extension]=image
done

# map extensions to text
for extension in $text_extensions; do
    extensions_map[$extension]=text
done

# map extensions to document
for extension in $document_extensions; do
    extensions_map[$extension]=document
done

# Loop through files in unorganized/.
for file in ./unorganized/*; do
    # i don't think we learn this in class -- i just google it; this should
    # keep only string after '.'
    file_extension=${file##*.}
    # this one is similar to the ones above, just cut at the backmost `/' in
    # file directory
    file_name=${file##*/}

    # '-z' prefix means missing/ don't exist
    if [[ -z ${extensions_map[$file_extension]} ]]; then
        # Create folders for each file extension if they 'dont exist.
        mkdir ./${file_extension}/
        # Move files to the correct folder.
        mv $file ./${file_extension}/${file_name}
        # For each file moved, print a one-line log on the screen. The log
        # should look like the following: "Moving <filename> to <foldername>/"
        echo "Moving $file_name to ./${file_extension}/"
    else
        case ${extensions_map[$file_extension]} in
            image)
                # Move files to the correct folder.
                mv $file ./images/${file_name}
                # For each file moved, print a one-line log on the screen. The log
                # should look like the following: "Moving <filename> to <foldername>/"
                echo "Moving $file_name to ./images/"
                ;;
            text)
                # Move files to the correct folder.
                mv $file ./texts/${file_name}
                # For each file moved, print a one-line log on the screen. The log
                # should look like the following: "Moving <filename> to <foldername>/"
                echo "Moving $file_name to ./texts/"
                ;;
        esac
    fi
done

```

```

        document)
            # Move files to the correct folder.
            mv $file ./documents/$file_name
            # For each file moved, print a one-line log on the screen. The
            # log should look like the following: "Moving <filename> to
            # <foldername>/"
            echo "Moving $file_name to ./documents/"
        ;;
    *)
        # panic, scream
        echo "Ahhhhhhhhhhhhhhhh!!!!"
    ;;
esac
fi
done

```

2. Before and After Screenshots:

- Before: Screenshot showing the `unorganized/` directory with mixed files.
- After: Screenshot showing organized folders (e.g., `texts/`, `images/`).

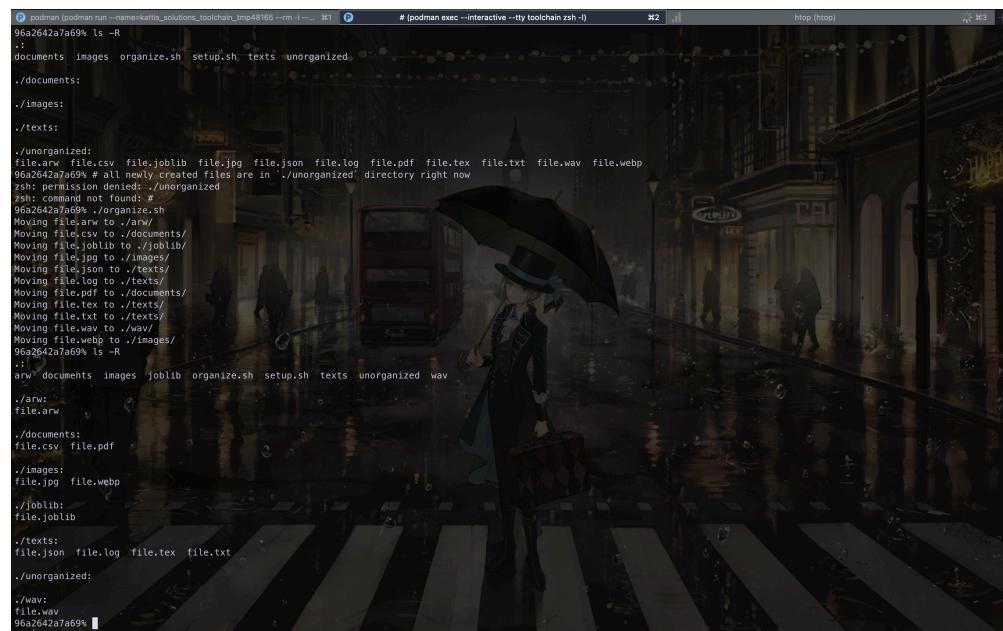
Solution



```

96a2642a7a69% ls
organize.sh setup.sh
96a2642a7a69% # run the 'setup.sh' script -- to setup
zsh: command not found: setup.sh
zsh: command not found: #
96a2642a7a69% ./setup.sh
96a2642a7a69% ls -R
./documents images organize.sh setup.sh texts unorganized
96a2642a7a69% ls -R
.:.
./documents images organize.sh setup.sh texts unorganized
./documents
./images
./texts
./unorganized:
file.csv file.csv file.joblib file.jpg file.json file.log file.pdf file.tex file.txt file.wav file.webp
96a2642a7a69% # all newly created files are in ./unorganized directory right now

```



```

96a2642a7a69% ls -R
.:.
./documents images organize.sh setup.sh texts unorganized
./documents
./images
./texts
./unorganized:
file.csv file.csv file.joblib file.jpg file.json file.log file.pdf file.tex file.txt file.wav file.webp
96a2642a7a69% # all newly created files are in ./unorganized directory right now
zsh: permission denied: ./unorganized
zsh: command not found: #
96a2642a7a69% ./organize.sh
Moving file.csv to ./unorganized/
Moving file.csv to ./documents/
Moving file.joblib to ./joblib/
Moving file.jpg to ./images/
Moving file.json to ./texts/
Moving file.log to ./documents/
Moving file.pdf to ./documents/
Moving file.tex to ./texts/
Moving file.txt to ./texts/
Moving file.wav to ./wav/
Moving file.webp to ./images/
96a2642a7a69% ls -R
.:.
arw: documents images joblib organize.sh setup.sh texts unorganized wav
./arw:
file.arw
./documents:
file.csv file.pdf
./images:
file.jpg file.webp
./joblib:
file.joblib
./texts:
file.json file.log file.tex file.txt
./unorganized:
./wav:
file.wav
96a2642a7a69%

```

3. Sample Output (Log):

- Copy and paste the terminal output showing the log for each file moved.

Solution

```
96a2642a7a69% ./organize.sh
Moving file.arw to ./arw/
Moving file.csv to ./documents/
Moving file.joblib to ./joblib/
Moving file.jpg to ./images/
Moving file.json to ./texts/
Moving file.log to ./texts/
Moving file.pdf to ./documents/
Moving file.tex to ./texts/
Moving file.txt to ./texts/
Moving file.wav to ./wav/
Moving file.webp to ./images/
```

4. Explanation:

- Write some sentences describing how the script works and the key commands used.

Solution

Firstly, `setup.sh` sets up the environment as instructed (you can check that by the screenshot). `organize.sh` organizes different file extensions into its place, else create a new directory for that unrecognised file. We creates a hash map like structure first, for faster lookup time – mapping file type (**key**) to its value (**destination folder**). Then, we use that magical lines to get `file_extension` and `file_name` – i don't think we learn this in class, I searched it up from google. The rest is just checking whether the key matches or not and move the files to its destination.