Shall Script Assignment

Omar Mohsin

Atypon Training July-2023

Instructors:

Dr. Motasim Aldiab

Dr. Fahed Jubair

Requirement:

build a shell script that does a following task.

1- Search the files in the given directory and specific file, and do a report about the file's information like name, last modified, size and owner

   and generate a summary file that contains the number of files and total size.

2- Implement user-friendly with help sectoin

My script is divided into the four parts

the first part is the help function, it will display the instruction and what each parameter do

the second part is to check what parameter is provided

and validation if the directory parameter exits and if the user put the directory or not

the third part is to generate the report file which name is file_analysis,

put the head of the table, which is the File Name   Size (bytes) Owner Permissions and Last Modified

They do the sorting thing and it depends on the user, so if the user put -s it will sort depending on the size of the files, the smallest file first,

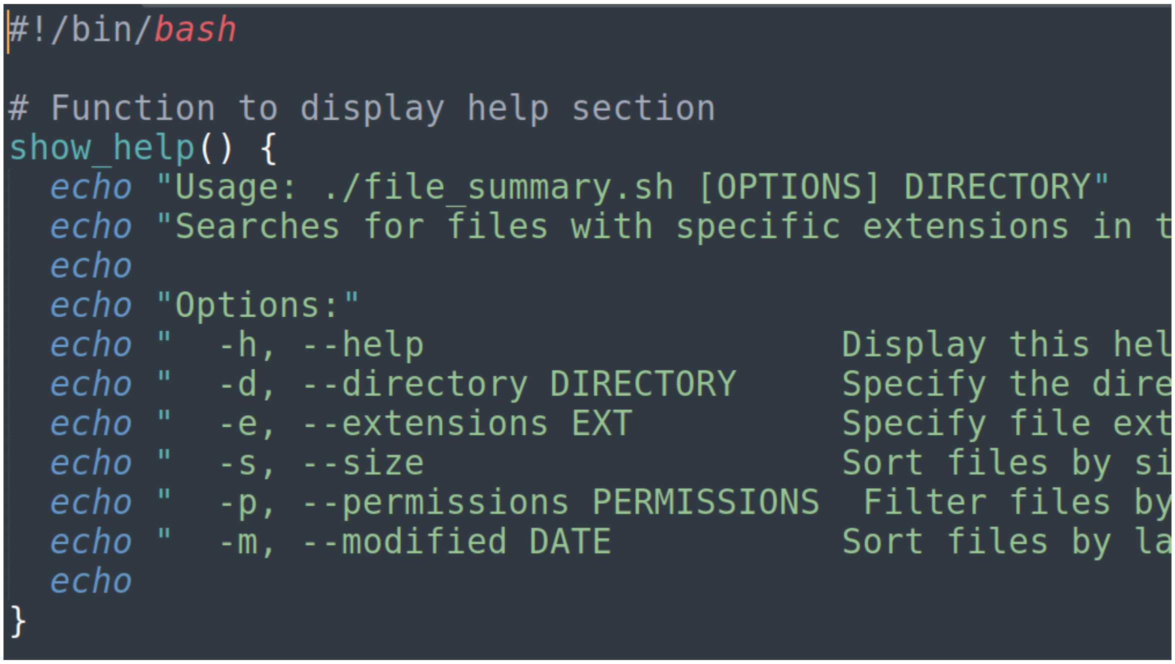
etc.,

the fourth part is to generate the summary file,

make 2 counters, a counter for the number of files, and a counter for the size of each file,

First part   show_help function :

When user type -h or --help the window will display and it will explain how to use the script and lists available options.



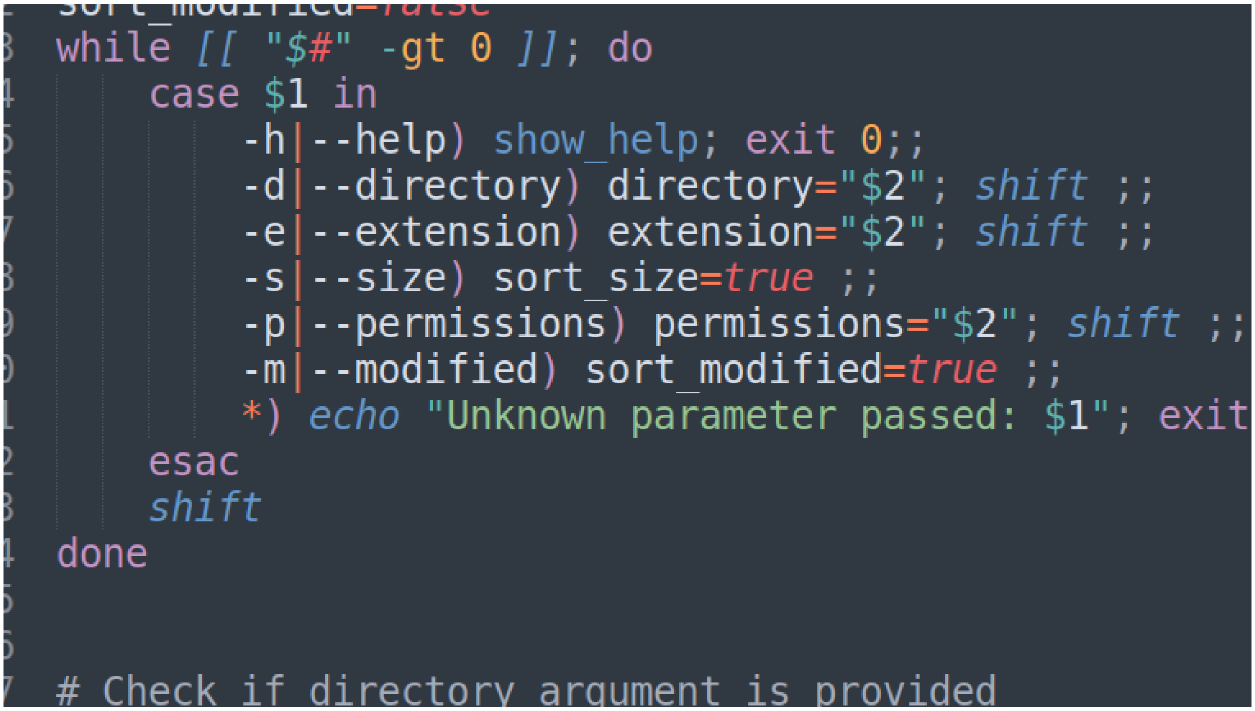
```
#!/bin/bash

# Function to display help section
show_help() {
  echo "Usage: ./file_summary.sh [OPTIONS] DIRECTORY"
  echo "Searches for files with specific extensions in t
  echo
  echo "Options:"
  echo "  -h, --help                    Display this hel
  echo "  -d, --directory DIRECTORY     Specify the dire
  echo "  -e, --extensions EXT          Specify file ext
  echo "  -s, --size                    Sort files by si
  echo "  -p, --permissions PERMISSIONS  Filter files by
  echo "  -m, --modified DATE           Sort files by la
  echo
}
```

Part two validation:

$# mean number of parameters and -gt mean greater than

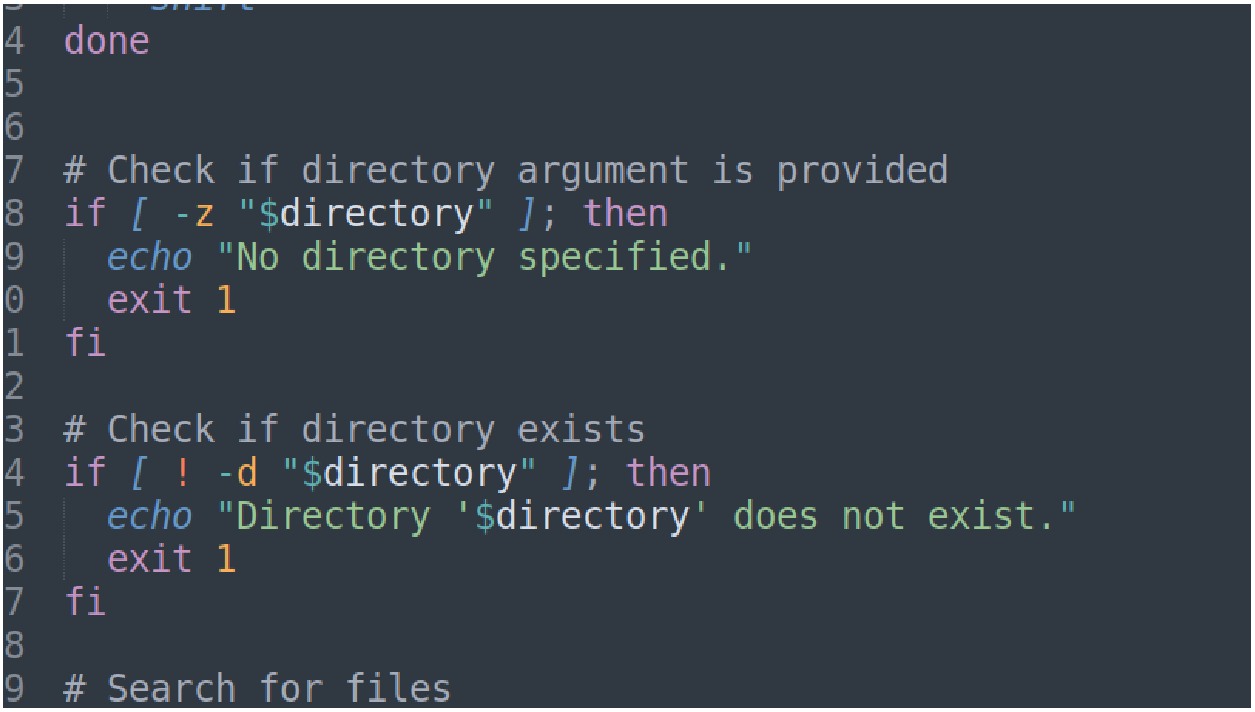
I will check how many parameters they are

and do the validation as follow, using switch statement



```
sort_modified=false
while [[ "$#" -gt 0 ]]; do
    case $1 in
        -h|--help) show_help; exit 0;;
        -d|--directory) directory="$2"; shift ;;
        -e|--extension) extension="$2"; shift ;;
        -s|--size) sort_size=true ;;
        -p|--permissions) permissions="$2"; shift ;;
        -m|--modified) sort_modified=true ;;
        *) echo "Unknown parameter passed: $1"; exit
    esac
    shift
done


# Check if directory argument is provided
```

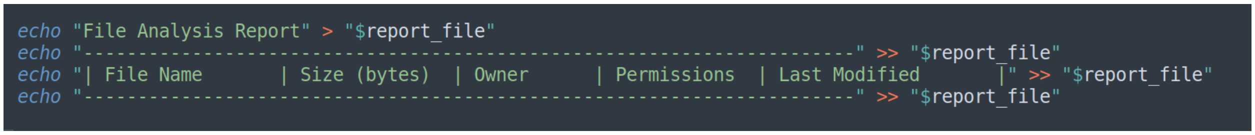
when you type the directory it will check if the directory exists, or if the client didn't put the directory as parameter



```
    shift
done


# Check if directory argument is provided
if [ -z "$directory" ]; then
  echo "No directory specified."
  exit 1
fi

# Check if directory exists
if [ ! -d "$directory" ]; then
  echo "Directory '$directory' does not exist."
  exit 1
fi

# Search for files
```

Part three generates the file_analysis,

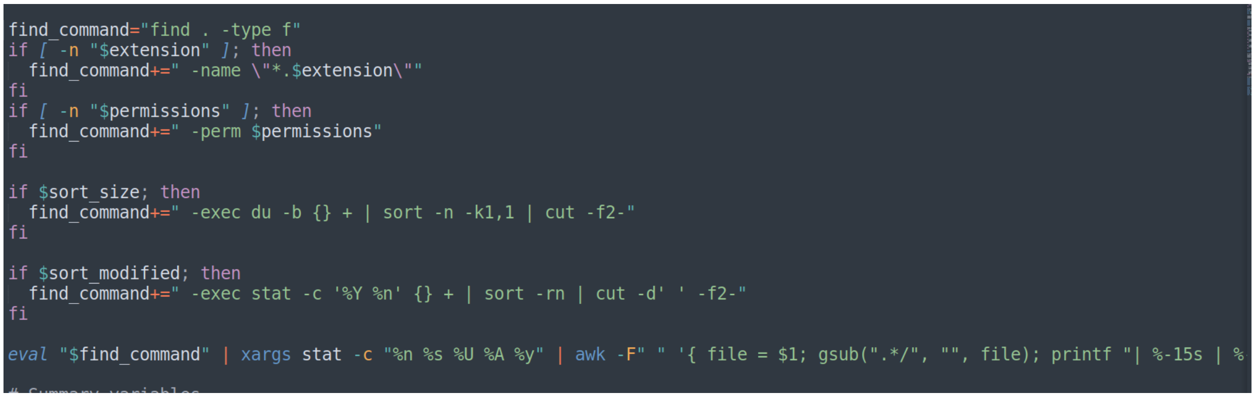
first thing put a general structure of the file, then append the content to it



```
echo "File Analysis Report" > "$report_file"
echo "----------------------------------------------------------------------------" >> "$report_file"
echo "| File Name       | Size (bytes)  | Owner      | Permissions  | Last Modified       |" >> "$report_file"
echo "----------------------------------------------------------------------------" >> "$report_file"
```

I will use the find command to find the files with a specific extension, and then store them in a variable,

I put a basic parameter to the find command, then append to that command another parameter, extensions, and sorting depending on the size or last modification, and then store it in the file_analysis file

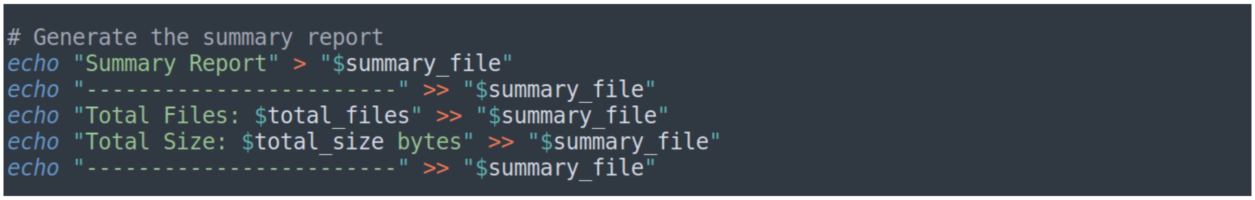


```
find_command="find . -type f"
if [ -n "$extension" ]; then
  find_command+=" -name \"*.$extension\""
fi
if [ -n "$permissions" ]; then
  find_command+=" -perm $permissions"
fi

if $sort_size; then
  find_command+=" -exec du -b {} + | sort -n -k1,1 | cut -f2-"
fi

if $sort_modified; then
  find_command+=" -exec stat -c '%Y %n' {} + | sort -rn | cut -d' ' -f2-"
fi

eval "$find_command" | xargs stat -c "%n %s %U %A %y" | awk -F" " '{ file = $1; gsub(".*/", "", file); printf "| %-15s | %
# Summary variables
```

```
10s | %-12s | %-19s |\n", file, $2, $3, $4, $5 }' >> "$report_file"
```

Part four: generate the summary file

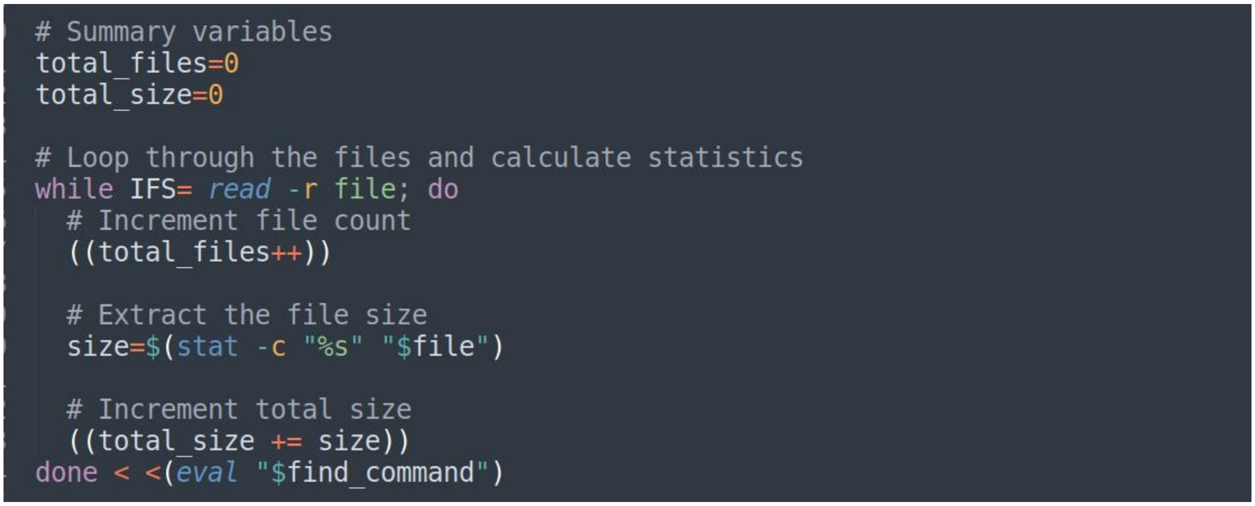
build a basic structure for the file



```
# Generate the summary report
echo "Summary Report" > "$summary_file"
echo "------------------------" >> "$summary_file"
echo "Total Files: $total_files" >> "$summary_file"
echo "Total Size: $total_size bytes" >> "$summary_file"
echo "------------------------" >> "$summary_file"
```

summary file the number of files and the total size of the files

I will make 2 counters which are the number of files and the total size



```
# Summary variables
total_files=0
total_size=0

# Loop through the files and calculate statistics
while IFS= read -r file; do
  # Increment file count
  ((total_files++))

  # Extract the file size
  size=$(stat -c "%s" "$file")

  # Increment total size
  ((total_size += size))
done < <(eval "$find_command")
```

End

Thank you !