

# Task 1

By: Ziad Ahmed Ibrahim

In this task I was asked to create script to copy a file from a place to another, script is required to take 2 types of arguments and should have a particular permission

1. name of the files to be copied
2. path of the destination directory

## 2 methods

1. take only 2 arguments
2. take multiple arguments

## Method 1

```
CP() {  
mkdir -p $(dirname "$2") && cp "$1" "$2"  
}
```

- In this method a function is created and is called by typing CP in the terminal
- then provide the name of the file to be copied and its path
- then provide the path you want to save it at
- notice that the the statement before the && will be executed first
- so if the the directory is not present it will be automatically created
- then the path used to make the directory is used with a cp command as destination of copying

## Method2

```
#By ziad ahmed  
#In this script I will be using the bash scripting to copy multiple files to a  
destination  
#Files must be in my current directory where I'm at in the terminal else won't  
work  
#example of command to use this script: ./script.sh text1.txt text2.txt  
/path/to/destination  
#####  
#####  
  
if [ "$#" -lt 2 ]; then # I used this condition to check the number of arguments  
is at least 2
```

```

    echo "Arguments are less than 2"
    exit 1
fi

# Now we made sure the arguments are 2 or more
des="${@: -1}" #Knowing the last arguments is the destination I will store it
here
if [ ! -d "$des" ]; then #if this destination doesn't exist we create it
    mkdir -p "$des"      #line where we create the directory needed
fi

for ((i=1;i< $#;i++)); do          #loop through the arguments where $# means number
of arguments
    file="${!i}"                  #variable to store file name for each iteration
through arguments
    if [ -f "$file" ]; then      # if file is a type file lue .txt or .sh tgeb
proceed
        cp "$file" "$des"       #copy the file to the destination
    else
        echo "This file \"$file\" is not valid "
    fi
done
echo "Operation successful"

```

- This method I took a completely different approach to take more than 2 arguments
- User should different files with their paths, as much as the user wants
- last argument must be the destination directory path
- This method ensures also that the arguments passed are of the right type to be copied, if they are not the invalid file will be skipped and the process will continue
- on the other hand if less than 2 arguments are passed the right error message is provided

The left screenshot shows a terminal window with the following output:

```

msi@msi-VirtualBox: ~
Copied text1.txt to /home/msi/destination
Files copied successfully.
msi@msi-VirtualBox:~$ ./script.sh text1.txt text2.txt /home/msi/destination
Done
msi@msi-VirtualBox:~$ ./script.sh text1.txt text2.txt /home/msi/destination
./script.sh: line 4: [: command not found
./script.sh: line 10: [: command not found
./script.sh: line 16: [: missing `]'
This file text1.txt is not valid
./script.sh: line 16: [: missing `]'
This file text2.txt is not valid
Operation successful
msi@msi-VirtualBox:~$ ./script.sh text1.txt text2.txt /home/msi/destination
Operation successful
msi@msi-VirtualBox:~$ mkdir
mkdir: missing operand
Try 'mkdir --help' for more information.
msi@msi-VirtualBox:~$ mkdir Copying_script
msi@msi-VirtualBox:~$ mv script.sh Copying_script
msi@msi-VirtualBox:~$ ls
Copying_script  Documents  Music  Pictures  Test  Videos
Desktop        Downloads  mscript Public    text1.txt
destination    examples.desktop  templates  Templates  text2.txt
msi@msi-VirtualBox:~$

```

The right screenshot shows the script's source code in a text editor:

```

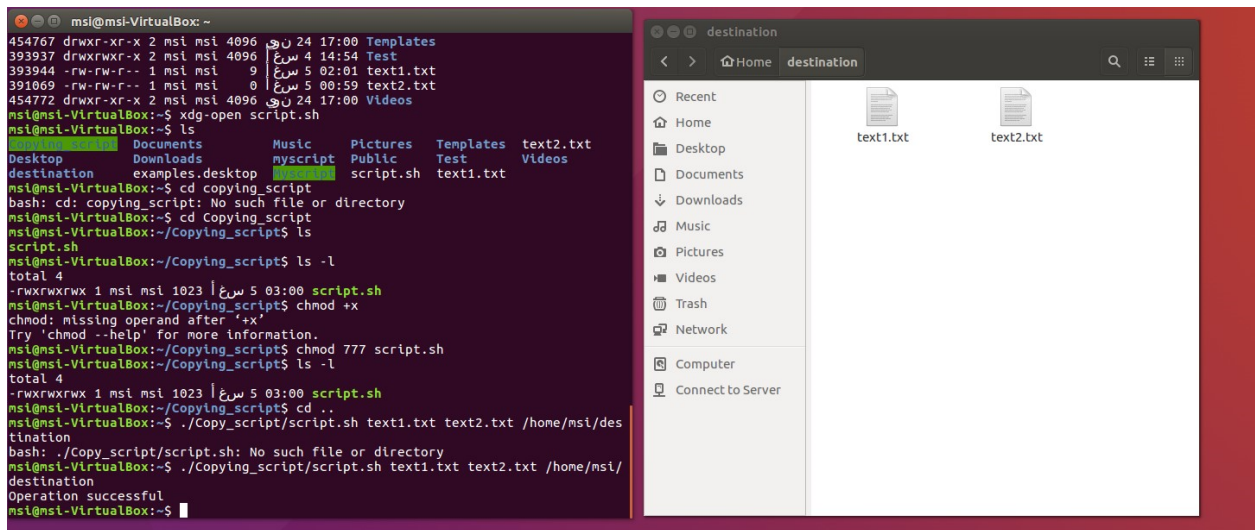
#By ziad ahmed
#In this script I will be using the bash scripting to copy multiple files to a destination
#Files must be in my current directory where I'm at in the terminal else won't worl

if [ "$#" -lt 2 ]; then # I used this condition to check the number of arguments is at least 2
    echo "Arguments are less than 2"
    exit 1
fi

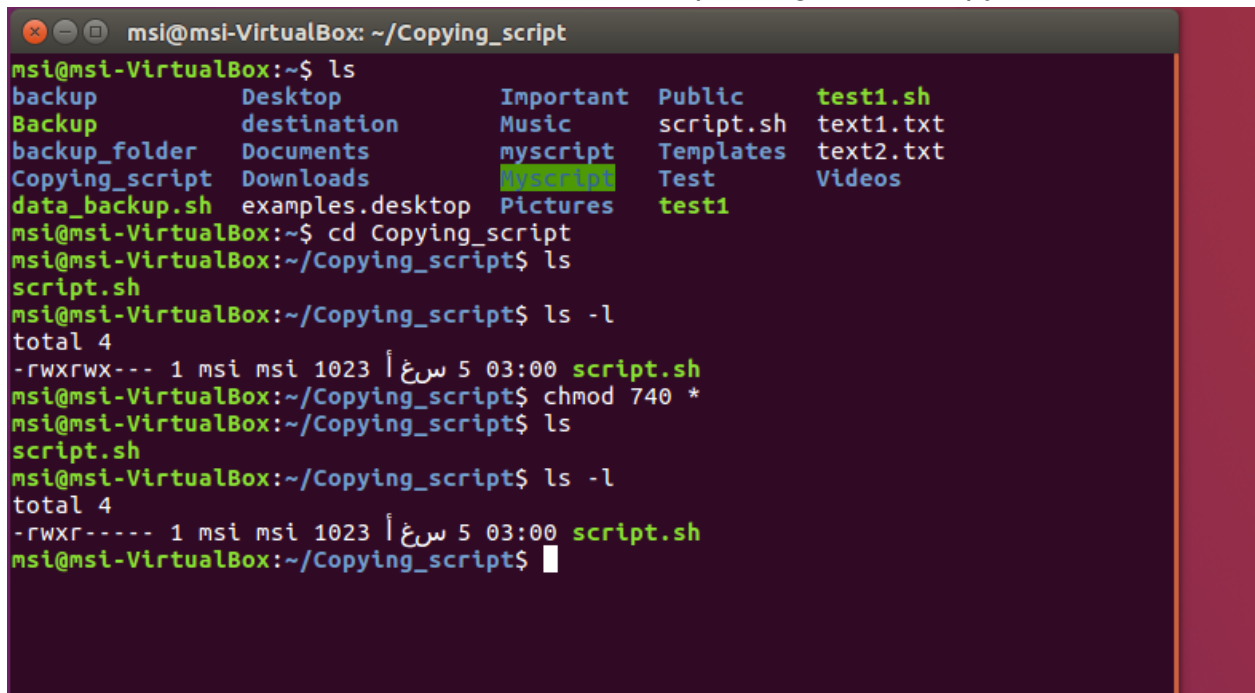
# Now we made sure the arguments are 2 or more
des="${@: -1}" #Knowing the last arguments is the destination I will store it here
if [ ! -d "$des" ]; then #If this destination doesn't exist we create it
    mkdir -p "$des"      #line where we create the directory needed
fi

for ((i=1;i< $#;i++)); do          #Loop through the arguments where $# means number of arguments
    file="${!i}"                  #Variable to store file name for each iteration through arguments
    if [ -f "$file" ]; then      # If file is a type file lue .txt or .sh tgeb proceed
        cp "$file" "$des"       #copy the file to the destination
    else
        echo "This file \"$file\" is not valid "
    fi
done
echo "Operation successful"

```



Like we can see here the terminal shows the script being used to copy files



Setting the appropriate permissions to the script