

# Scripting

Goal : You are a heavy downloader. Songs, movies, documents mainly. You download so much that your Downloads folder, on your computer became some extension of your trash folder, and needs to be sorted. However, the task is huge, and you don't want to do it yourself.

The goal of this practical work is to solve this solution

We are going to automatically move files after downloads.

When a .pdf file will be created in /home/user/Downloads folder, it will automatically be moved to /home/user/Documents.

When a .mp3 file will be created in /home/user/Downloads folder, it will automatically be moved to /home/user/Music.

When a .mp4 file will be created in /home/user/Downloads folder, it will automatically be moved to /home/user/Videos.

Some sorting remains, so if you want to add new features to your solution, keep it imaginative !

## First step : detect IO in Downloads folder

Install inotifywait package.

Allow yourself to use inotifywait :

```
echo $USER | sudo tee -a /etc/inotifywait.allow
```

The syntax of inotifywait is the following : <path to listen> <event type> <command to execute>

Ex :

```
/home/user/Downloads IN_CLOSE_WRITE /home/user/Documents/inotifywait_script.sh
```

Now, read manual and find the command to use to display absolute file name of new files in Downloads folder. Use inotifywait -e to edit the inotifywait. You might specify the EDITOR environment variable to nano.

Create several files in Downloads folder. Check /var/log/syslog file to see if it behaves as specified.

## Second step : Detecting file extension

Create a script for parsing file extension.

Here is sample input and expected result to test your script :

Note : To fail with error code 1 use exit 1

```
$ ./parse_extension.sh /home/directory/file.ext
ext

$ ./parse_extension.sh /home/directory/file
Failed error code 1

$ ./parse_extension.sh /home/directory/file.trap.ext
ext

$ ./parse_extension.sh /home/directory/ext.trap.ext
ext

$ ./parse_extension.sh /home/directory.trap/file.ext
ext

$ ./parse_extension.sh /home/directory/file.
Failed error code 1
```

```
$ ./parse_extension.sh /home/direcory/  
Failed error code 1
```

Don't forget to give it execution right.

## Third step : extract file name from absolute path

To test your script :

```
$/parse_file_name.sh /home/directory/file.ext  
file.ext  
  
$/parse_file_name.sh /home/directory/file  
file  
  
$/parse_file_name.sh /file.ext  
file.ext  
  
$/parse_file_name.sh file.ext  
file.ext  
  
$/parse_file_name.sh /home/directory/  
Failed error code 1
```

## Fourth step : classification script

Use previously written scripts to write classification script.

To test it :

```
# For documents  
$ ls ~/Documents  
No test.pdf file  
  
$ touch test.pdf  
  
$ ./classify.sh test.pdf  
  
$ ls ~/Documents  
test.pdf file present  
  
# For music  
$ ls ~/Music  
No test.mp3 file  
  
$ touch test.mp3  
  
$ ./classify.sh test.mp3  
  
$ ls ~/Music  
test.mp3 is present  
  
# For movies  
$ ls ~/Videos  
No test.mp4 file  
  
$ touch test.mp4  
  
$ ./classify.sh test.mp4
```

```
$ ls ~/Videos
test.mp4 is present
```

**# Working with absolute paths**

```
$ ls ~/Videos
No abs.mp4
```

```
$ touch /home/user/Downloads/abs.mp4
```

```
$ ./classify.sh /home/user/Downloads/abs.mp4
```

```
$ ls ~/Videos
abs.mp4 should be present
```

## Fiveth step : plug it into incron

Edit incron -e to class classify.sh script with full path name upon writes close in Download folder.

To test it :

```
# For documents
$ ls ~/Documents
No test.pdf file
```

```
$ touch ~/Downloads/test.pdf
```

```
$ ls ~/Documents
test.pdf file present
```

```
# For music
$ ls ~/Music
No test.mp3 file
```

```
$ touch ~/Downloads/test.mp3
```

```
$ ls ~/Music
test.mp3 is present
```

```
# For movies
$ ls ~/Videos
No test.mp4 file
```

```
$ touch ~/Downloads/test.mp4
```

```
$ ls ~/Videos
test.mp4 is present
```

## You still have time

Improve the script !

You can :

- Handle more extensions
- Verify file extensions
- Have finer sorting decisions (ex : based on size, classify movies and series (shorter) )
- Whatever you can imagine