

Санкт-Петербургский Национальный Исследовательский Университет  
Информационных Технологий, Механики и Оптики

Факультет Систем Управления и Робототехники

**Практическая работа №1**  
**Анализатор файлов**

Выполнил: Антипов В. А.

Проверил: Капитонов А.А

Санкт-Петербург  
2019

## Описание работы

Необходимо написать скрипт, который создает таблицу с данными о содержащихся в директории файлах.

В таблице обязательно должны присутствовать следующие столбцы:

- название файла;
- расширение файла;
- дата изменения;
- размер (желательно в МБ)
- длительность аудио и видео файлов

## Выполнение работы

В самом начале программы проверяется наличие введенных аргументов: папки, в которой необходимо искать и имени выходного файла. В случае отсутствия последнего, файлу с результатами дается стандартное имя *out.xls*

Для выполнения поиска файлов по всем вложенным папкам использовалась команда:

```
find <folder> -type f -not -name '.*'
```

Далее для строк полученных таким образом итеративно выполнялась операция взятия имени файла с помощью *awk* с применением регулярного выражения:

```
files_name=$((echo "$line" | awk -F/ '{print $NF}')\n| awk '{gsub(" ", "_", $0)} 1'))
```

Размер файла определяется при помощи *du -hs*:

```
sizes=( "$((du -hs "$line" | cut -f1)\n| awk '{gsub(",", ".", $0)} 1')" )
```

Дата последнего редактирования была получена при помощи *stat*:

```
datetime="$ (stat -c%.10y "$line")"
```

Extension файла был определен по последней точке в имени файла. В случае отсутствия расширения у файла, в поле будет записано значение *unknow*

```
types=("${files_name#*.}")
```

Длительность видео и аудио позволяет получить чудесная программа *ffmpeg*. Для этого при помощи *ffprobe* все доступной информации с помощью определенных флагов получается время в секундах, которое затем переводится в минуты:

```
chk_video=$((ffprobe -v error -show_entries format=duration -of\ndefault=noprint_wrappers=1:nokey=1\n"$line" 2>&1 | tail -n1)| awk -F': ' '{print $NF}'))\n\nvideo_time="$ (bc <<< "scale=2; $chk_video/60") Min")
```

Этой же командой проверяется видео и музыкальный файл на битость, результат не зависит от расширения видео или аудио файлов.

```
texnoman@texnomanPC:/media/files/Uni/SSP_Labs/lab1$ ./fileAnalyzer.sh data/
Analyze files in data/:
/ Search files...
```

Рис 1. Запуск файла и индикация процесса анализа файлов

Далее формируется строка с разделенными в ней полями при помощи запятых. Строка пишется в *csv* файл, который может быть контейнером для *xls* файла. На каждой итерации анализа файла и записи информации в *csv* файл обновляется статус бар:

```
spin() {
    printf "\r${sp:sc++:1} Search files..."
    ((sc==${#sp})) && sc=0
}
```

, где *sp* - массив из различных видов слэша

```
sp=' / - \ | '
```

```
1  #!/usr/bin/env bash
2  folder=$1
3
4  output=$2
5  if [ "$output" == "" ]; then
6      output="out"
7  fi
8
9  if [ "$folder" == "" ]; then
10     folder=""
11 fi
12
13 out_string="Name,Create data,Size,Extension,Time for video and music"
14 sp=' / - \ | '
15 sc=0
16 spin() {
17     printf "\r${sp:sc++:1} Search files..."
18     ((sc==${#sp})) && sc=0
19 }
20 endspin() {
21     printf '\r%s\n' "$@"
22     sleep 0.1
23 }
24
25 echo "Analyze files in $folder: "
26 echo -e $out_string > $output".xls"
27
28 while IFS= read -r line; do
29     out_string=""
30     files_name=$((echo "$line" | awk -F/ '{print $NF}')) | awk '{gsub(" ", "_", $0)} 1')
31     sizes=( $(du -hs "$line" | cut -f1) | awk '{gsub(" ", "_", $0)} 1')
32     datetime=$(stat -c%.10y "$line")
33     if [ "${files_name#*}" == "$files_name" ]; then
34         types=("unknow")
35         video_time=("-")
36     else
37         types=("${files_name#*}")
38         chk_video=$((ffprobe -v error -show_entries format=duration -of \
39             default=noprint_wrappers=1:nokey=1 "$line" 2>&1 | tail -n1) | awk -F': ' '{print $NF}'))
40         if [[ $chk_video =~ ^[+-]?[0-9]+\.?[0-9]*$ ]] && [ $chk_video != "0.040000" ]; then
41             video_time=$((bc <<< "scale=2; $chk_video/60") Min)
```

Рис 2.1 Текст программы

```

36     else
37         types=("${files_name#*.}")
38         chk_video=$((ffprobe -v error -show_entries format=duration -of \
39             default=noprint_wrappers=1:nokey=1 "$line" 2>&1 | tail -n1) | awk -F': ' '{print $NF}')
40         if [[ $chk_video =~ ^[+-]?[0-9]+\.[0-9]*$ ]] && [ $chk_video != "0.040000" ]; then
41             video_time=$((bc <<< "scale=2; $chk_video/60") Min)
42         else
43             video_time="-"
44         fi
45     fi
46
47     out_string+="${files_name}; out_string+=","
48     out_string+="${datetime}; out_string+=","
49     out_string+="${sizes}; out_string+=","
50     out_string+="${types}; out_string+=","
51     out_string+="${video_time};
52
53     spin
54     echo -e $out_string >> $output.xls"
55     # printf "\b${line}"
56 done < <(find "$folder" -type f -not -name '.*')
57
58 endspin
59 printf "\nWrite into xls file end\n"
60 # echo $string

```

Рис 2.2 Текст программы

В результате выполнения программы создается *xls* файл со следующим содержимым:

	A	B	C	D	E
1	Name	Create data	Size	Extension	Time for video and music
2	blumba	2019-09-24	0	unknow	-
3	out.xls	2019-09-24	8.0K	xls	-
4	dfwdfwf/wdef	2019-09-24	512	unknow	-
5	gorillaz-feel-good-inc.mp3	2019-09-24	8.5M	mp3	3.71 Min
6	g_t	2019-09-24	512	unknow	-
7	hk	2019-09-24	0	unknow	-
8	mu.mp3	2019-09-24	0	mp3	-
9	p.txt	2019-09-30	512	txt	-
10	second	2019-09-24	0	unknow	-
11	sim_test.mp4	2019-09-24	0	mp4	-
12	test	2019-09-24	0	unknow	-
13	test.mp4	2019-09-24	16M	mp4	2.04 Min
14	6a6yk	2019-09-24	0	unknow	-
15	666yk	2019-09-24	0	unknow	-

Рис 3. Таблица с результатом выполнения программы