



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ \_\_\_\_\_ Информатика и системы управления \_\_\_\_\_

КАФЕДРА \_ Программное обеспечение ЭВМ и информационные технологии \_

## ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ

Студент \_\_\_\_\_ Фам Минь Хиеу \_\_\_\_\_

Группа \_\_\_\_\_ ИУ7-12Б \_\_\_\_\_

Тип пратики: \_\_\_\_\_ проектно - технологическая пратика \_\_\_\_\_

Название предприятия: \_\_\_\_\_ МГТУ им. Н. Э. Баумана \_\_\_\_\_

Студент \_\_\_\_ Фам Минь Хиеу \_\_\_\_\_

\_\_\_\_\_  
(Группа)  
(И.О.Фамилия)

\_\_\_\_\_  
(Подпись, дата)

Проверяющий :

\_ Кострицкий Александр Сергеевич \_

\_\_\_\_\_  
(Подпись, дата)

Проверяющий

\_ Ломовской Игорь Владимирович \_

\_\_\_\_\_  
(Подпись, дата)

Москва 2021г.

# Оглавление

|                                     |  |
|-------------------------------------|--|
| 1. Подстановка задачи.....          |  |
| 2. Особенности реализации.....      |  |
| 3. Ограничения по безопасности..... |  |
| 4. Тестовые случаи.....             |  |
| 5. Инструкции пользователя.....     |  |
| 6. Листинг исходного кода.....      |  |

# Подстановка задачи

Разработать скрипт командной оболочки для обработки текстовых файлов.

1. У скрипта есть список расширений временных файлов. По умолчанию список состоит из «\*.log».

2. У скрипта есть список расширений рабочих файлов. По умолчанию список состоит из «\*.py».

3. У скрипта есть рабочая папка, в которой выполняется вся работа скрипта.

По умолчанию это папка самого скрипта.

4. Настройки скрипта сохраняются в файле «.myconfig» рядом со скриптом.

Если файл при запуске нельзя обнаружить, генерируется файл настроек по умолчанию.

5. У скрипта есть записанная пользователем в виде строки команда. По умолчанию это «grep def\* program.py >last.log».

Скрипт должен предоставлять пользователю с помощью меню и текстового

интерфейса следующие возможности:

1. Возможность просмотреть или задать заново список расширений временных файлов.

2. Возможность добавлять или удалять конкретное расширение из списка расширений временных файлов. Достаточно реализовать удаление по

номеру.

3. Возможность просмотреть или задать заново список расширений рабочих файлов.

4. Возможность добавлять или удалять конкретное расширение из списка расширений рабочих файлов. Достаточно реализовать удаление по номеру.

5. Возможность просмотреть, изменить или задать заново рабочую папку скрипта.

6. Возможность удалить временные файлы.

7. Возможность выполнить или изменить записанную команду.

8. Возможность просмотреть все строки, ограниченные апострофами, во всех рабочих файлах.

9. Возможность просмотреть объём каждого временного файла.

## Ограничения по безопасности

В сценарии предусмотрена проверка на запуск от имени администратора. Пользователь при желании может прекратить работу

Пример

```
#проверка на запуск от имени администратора
if [ "$EUID" -eq 0 ]; then
    echo "запуск от имени администратора"
    echo -n "вы хотите продолжить? (y/n): "
    read -r ans
    if [ "$ans" == "n" ]; then
        exit 1
    fi
fi
```

# Особенность реализации

**Скрипт имеет 2 режима работы:**

*1. Взаимодействие с пользователем происходит в виде текстного меню:*

Для выбора необходимо написать число соответственно. Если пользователь вводит не корректное число, то скрипт будет ему сообщать об ошибках.

**Пример сообщения о неправильном вводе**

```
1. просмотреть список расширений временных файлов.
2. задать заново список расширений временных файлов.
3. добавлять конкретное расширение временных файлов.
4. удалять конкретное расширение из списка расширений временных файлов.
5. просмотреть список расширений рабочих файлов.
6. задать заново список расширений рабочих файлов.
7. добавлять конкретное расширение рабочих файлов.
8. удалять конкретное расширение из списка расширений рабочих файлов.
9. просмотреть рабочую папку скрипта
10. задать заново рабочую папку скрипта
11. удалить временные файлы
12. просмотреть записанную команду
13. выполнить записанную команду.
14. изменить записанную команду.
15. просмотреть все строки, ограниченные апострофами, во всех рабочих файлах.
16. просмотреть объём каждого временного файла
0. Завершить
какой выбор: 7623
выбор должен быть от 0 до 9: 24
выбор должен быть от 0 до 9: -4
выбор должен быть от 0 до 9:
```

Номер пункта начиная от 0 до 16 ( 0 для завершения работы )

Скрипт написан для отработки с файлами ( временными и рабочими ). По умолчанию временные файлы записываются расширением .log, рабочие файлы расширением .ру . Директория рабочей папки, в случае новой рабочей папки. Путь может быть как относительным, так и абсолютным. Выбранная папка должна существовать в файловом пространстве.

В начале скрипта происходит табличка, в которой все созданы по умолчанию.

### Пример 1.2

```
temp_file= .log
work_file= .py
working_dir=/c/Users/Admin
commanda:grep def* program.py
```

Скрипт позволяет записывать и выполнять любую записанную команду. Пользователь должен узнать, что будет делать команда перед выполнением, потому что скрипт совсем не знает или проверяет не ее безопасность. Например команда `rm -rf [файл]`. Скрипт позволяет изменить путь, чтобы иметь возможность обрабатывать файлы в другом каталоге, а не в том же скрипте. Скрипт также позволяет искать во всех файлах строки, ограниченные апострофами.

### Пример 1.3

```
1. просмотреть список расширений временных файлов.
2. задать заново список расширений временных файлов.
3. добавлять конкретное расширение временных файлов.
4. удалять конкретное расширение из списка расширений временных файлов.
5. просмотреть список расширений рабочих файлов.
6. задать заново список расширений рабочих файлов.
7. добавлять конкретное расширение рабочих файлов.
8. удалять конкретное расширение из списка расширений рабочих файлов.
9. просмотреть рабочую папку скрипта
10. задать заново рабочую папку скрипта
11. удалить временные файлы
12. просмотреть записанную команду
13. выполнить записанную команду.
14. изменить записанную команду.
15. просмотреть все строки, ограниченные апострофами, во всех рабочих файлах.
16. просмотреть объем каждого временного файла
0. Завершить
какой выбор: 15
'papa'
'mama'
```

Скрипт также может определить размер временных файлов.

### Пример 1.4

```
1. просмотреть список расширений временных файлов.
2. задать заново список расширений временных файлов.
3. добавлять конкретное расширение временных файлов.
4. удалять конкретное расширение из списка расширений временных файлов.
5. просмотреть список расширений рабочих файлов.
6. задать заново список расширений рабочих файлов.
7. добавлять конкретное расширение рабочих файлов.
8. удалять конкретное расширение из списка расширений рабочих файлов.
9. просмотреть рабочую папку скрипта
10. задать заново рабочую папку скрипта
11. удалить временные файлы
12. просмотреть записанную команду
13. выполнить записанную команду.
14. изменить записанную команду.
15. просмотреть все строки, ограниченные апострофами, во всех рабочих файлах.
16. просмотреть объём каждого временного файла
0. Завершить
какой выбор: 16
676 /c/Users/Admin/ererer - Copy (2).log
676 /c/Users/Admin/ererer - Copy (3).log
676 /c/Users/Admin/ererer - Copy (4).log
676 /c/Users/Admin/ererer - Copy (5).log
676 /c/Users/Admin/ererer - Copy.log
676 /c/Users/Admin/ererer.log
0 /c/Users/Admin/last.log
```

2.Скрипт также позволяет работу в тихом режиме. Он активируется автоматически если пользователь передаст параметр. Первый параметр – это номер пункта, который вам нравится, данше второй параметр, будет числом или строкой зависит от того, что этой функции нужно. Чтобы не допустить досадных ошибок, внимательно просмотрите таблицу в разделе инструкций.

# Тестовые случаи

## 1.Тест добавления нового расширения временных файлов

```
temp_file= .txt .dat
work_file= .py .txt
working_dir=/c/Users/Admin
commanda:grep def* program.py
```

1. просмотреть список расширений временных файлов.
  2. задать заново список расширений временных файлов.
  3. добавлять конкретное расширение временных файлов.
  4. удалять конкретное расширение из списка расширений временных файлов.
  5. просмотреть список расширений рабочих файлов.
  6. задать заново список расширений рабочих файлов.
  7. добавлять конкретное расширение рабочих файлов.
  8. удалять конкретное расширение из списка расширений рабочих файлов.
  9. просмотреть рабочую папку скрипта
  10. задать заново рабочую папку скрипта
  11. удалить временные файлы
  12. просмотреть записанную команду
  13. выполнить записанную команду.
  14. изменить записанную команду.
  15. просмотреть все строки, ограниченные апострофами, во всех рабочих файлах.
  16. просмотреть объем каждого временного файла
0. Завершить  
какой выбор: 3  
введите новое расширение (пример .log): .abc

```
temp_file= .txt .dat .abc
work_file= .py .txt
working_dir=/c/Users/Admin
commanda:grep def* program.py
```

## 2.Тест удаления временных файлов

1. просмотреть список расширений временных файлов.
2. задать заново список расширений временных файлов.
3. добавлять конкретное расширение временных файлов.
4. удалять конкретное расширение из списка расширений временных файлов.
5. просмотреть список расширений рабочих файлов.
6. задать заново список расширений рабочих файлов.
7. добавлять конкретное расширение рабочих файлов.
8. удалять конкретное расширение из списка расширений рабочих файлов.
9. просмотреть рабочую папку скрипта
10. задать заново рабочую папку скрипта
11. удалить временные файлы
12. просмотреть записанную команду
13. выполнить записанную команду.



14. изменить записанную команду.  
15. просмотреть все строки, ограниченные апострофами, во всех рабочих файлах.  
16. просмотреть объём каждого временного файла  
0. Завершить  
какой выбор: 11  
/c/Users/Admin/ahihi.txt  
/c/Users/Admin/ds.txt  
введите название файла для удаления:(например aa.log, \*.log): \*.txt

### 3.Тест вводить новую команду

1. просмотреть список расширений временных файлов.  
2. задать заново список расширений временных файлов.  
3. добавлять конкретное расширение временных файлов.  
4. удалять конкретное расширение из списка расширений временных файлов.  
5. просмотреть список расширений рабочих файлов.  
6. задать заново список расширений рабочих файлов.  
7. добавлять конкретное расширение рабочих файлов.  
8. удалять конкретное расширение из списка расширений рабочих файлов.  
9. просмотреть рабочую папку скрипта  
10. задать заново рабочую папку скрипта  
11. удалить временные файлы  
12. просмотреть записанную команду  
13. выполнить записанную команду.  
14. изменить записанную команду.  
15. просмотреть все строки, ограниченные апострофами, во всех рабочих файлах.  
16. просмотреть объём каждого временного файла  
0. Завершить  
какой выбор: 14  
введите новую команду: echo helloeveryone  
  
temp\_file= .txt .dat .abc .ooo  
work\_file= .py .txt  
working\_dir=/c/Users/Admin  
commanda:echo helloeveryone

#### 4.Тест найти во всех рабочих файлов строки, ограниченные апострофами

```
1. просмотреть список расширений временных файлов.
2. задать заново список расширений временных файлов.
3. добавлять конкретное расширение временных файлов.
4. удалять конкретное расширение из списка расширений временных файлов.
5. просмотреть список расширений рабочих файлов.
6. задать заново список расширений рабочих файлов.
7. добавлять конкретное расширение рабочих файлов.
8. удалять конкретное расширение из списка расширений рабочих файлов.
9. просмотреть рабочую папку скрипта
10. задать заново рабочую папку скрипта
11. удалить временные файлы
12. просмотреть записанную команду
13. выполнить записанную команду.
14. изменить записанную команду.
15. просмотреть все строки, ограниченные апострофами, во всех рабочих файлах.
16. просмотреть объём каждого временного файла
0. Завершить
какой выбор: 15
'пара'
'тата'

temp_file= .txt .dat .abc .ooo
work_file= .py .txt
working_dir=/c/Users/Admin
commanda:echo helloeveryone
```

#### 5. Тест удаления расширения рабочих файлов

```
1. просмотреть список расширений временных файлов.
2. задать заново список расширений временных файлов.
3. добавлять конкретное расширение временных файлов.
4. удалять конкретное расширение из списка расширений временных файлов.
5. просмотреть список расширений рабочих файлов.
6. задать заново список расширений рабочих файлов.
7. добавлять конкретное расширение рабочих файлов.
8. удалять конкретное расширение из списка расширений рабочих файлов.
9. просмотреть рабочую папку скрипта
10. задать заново рабочую папку скрипта
11. удалить временные файлы
12. просмотреть записанную команду
13. выполнить записанную команду.
14. изменить записанную команду.
15. просмотреть все строки, ограниченные апострофами, во всех рабочих файлах.
16. просмотреть объём каждого временного файла
0. Завершить
какой выбор: 8
1.py
2.txt
введите номер, который необходимо удалить: 2

temp_file= .txt .dat .abc .ooo
work_file= .py
```

```
working_dir=/c/Users/Admin  
commanda:echo helloeveryone
```

## Инструкция пользователя

Для запуска скрипта в директории с ним должен находиться файл `.config` (при первом запуске из директории он будет создан скриптом со значениями по умолчанию), в котором находятся 4 строки: первая список расширений временных файлов, вторая список расширений рабочих файлов, третья путь к рабочей папке и четвертая записанная команда, которая должна выполнять

```
temp_file= .log  
work_file= .py .txt  
working_dir=/c/Users/Admin  
commanda:grep def* program.py
```

Вот так выглядит файл `.myconfig`

При запуске программы, если аргументов нет, программа перейдет в интерактивный режим с пользователем. Нужно ввести номер, который вам нравится.

1. просмотреть список расширений временных файлов.
  2. задать заново список расширений временных файлов.
  3. добавлять конкретное расширение временных файлов.
  4. удалять конкретное расширение из списка расширений временных файлов.
  5. просмотреть список расширений рабочих файлов.
  6. задать заново список расширений рабочих файлов.
  7. добавлять конкретное расширение рабочих файлов.
  8. удалять конкретное расширение из списка расширений рабочих файлов.
  9. просмотреть рабочую папку скрипта
  10. задать заново рабочую папку скрипта
  11. удалить временные файлы
  12. просмотреть записанную команду
  13. выполнить записанную команду.
  14. изменить записанную команду.
  15. просмотреть все строки, ограниченные апострофами, во всех рабочих файлах.
  16. просмотреть объем каждого временного файла
  0. Завершить
- какой выбор:

В общем, все функции работают с помощью функции `take_var` и функции `update_file`. Функция `take_var` ищет в файле `.myconfig` соответствующие данные, отделенные от имени знаком `=`, и возвращает необходимые значения, такие как массив `temp_file` массив `work_file` переменную `commanda` и переменную `directory`.

```
# возвращаются значение temp_file, work_file ,w_dir, cmline
take_var () {
    temp=$(grep 'temp_file' .myconfig)
    temp_file=( )
    IFS=" "
    for i in $temp; do
        if ! [[ $i == "temp_file=" ]]; then
            temp_file+=( "$i" )
        fi
    done
    work=$(grep "work_file" .myconfig)
    work_file=( )
    IFS=" "
    for i in $work; do
        if ! [[ $i == "work_file=" ]]; then
            work_file+=( "$i" )
        fi
    done
    cmline=$(grep "^commanda" .myconfig)
    IFS=":"
    for i in $cmline; do
        if ! [[ $i == "commanda" ]]; then
            com=$i
        fi
    done
    x=$(grep "working_dir" .myconfig)
    IFS="="
    for i in $x; do
        if ! [[ $x == "working_dir" ]]; then
            w_dir=$i
        fi
    done
}
```

Функция `update_file` с помощью команды `echo` и перенаправления выводов переписывается все данные, один из которых изменился.

```
# Обновление (перезапись) файла .myconfig
```

```
update_file () {  
    echo -e "temp_file= \"${temp_file[@]}\""\nwork_file=  
"${work_file[@]}\""\nworking_dir=\"${w_dir}\""\ncommanda:\"$com" > .myconfig  
}
```

Первая функция, после ввода цифры 1 вы будете смотреть все расширения временных файлов. Аналогично чтобы увидеть все расширения рабочих файлов, пути и команды нажмите 5, 9, 12. Вы можете задать заново и расширения временных рабочих файлов, путь к рабочей папке и записанную команду когда нажаты кнопки 2, 6, 10, 14

Если нажать 2, после чего требуется ввести новые расширения через пробел и нажать Enter для завершения ввода

```
15. просмотреть все строки, ограниченные апострофами, во всех рабочих  
файлах.  
16. просмотреть объём каждого временного файла  
0. Завершить  
какой выбор: 2  
введите новые расширения через пробел( пример .log .dat .bin ):
```

Еще вы можете добавлять какое-то расширение или удалить конкретное расширение. Для добавления введите номер пункта и дальше введите расширение. Для удаления также введите номер пунктов и выберите какой номер расширения, который необходимо удалить.

```
15. просмотреть все строки, ограниченные апострофами, во всех рабочих  
файлах.  
16. просмотреть объём каждого временного файла  
0. Завершить  
какой выбор: 4  
1.a  
2.b  
3.c  
4.d  
5.e  
введите номер, который необходимо удалить:
```

16-ая функция, нажмите 16 для просмотра объем каждого временного файла

```
1. просмотреть список расширений временных файлов.
2. задать заново список расширений временных файлов.
3. добавлять конкретное расширение временных файлов.
4. удалять конкретное расширение из списка расширений временных файлов.
5. просмотреть список расширений рабочих файлов.
6. задать заново список расширений рабочих файлов.
7. добавлять конкретное расширение рабочих файлов.
8. удалять конкретное расширение из списка расширений рабочих файлов.
9. просмотреть рабочую папку скрипта
10. задать заново рабочую папку скрипта
11. удалить временные файлы
12. просмотреть записанную команду
13. выполнить записанную команду.
14. изменить записанную команду.
15. просмотреть все строки, ограниченные апострофами, во всех рабочих файлах.
16. просмотреть объем каждого временного файла
0. Завершить
какой выбор: 16
676 /c/Users/Admin/ererer - Copy (2).log
676 /c/Users/Admin/ererer - Copy (3).log
676 /c/Users/Admin/ererer - Copy (4).log
676 /c/Users/Admin/ererer - Copy (5).log
676 /c/Users/Admin/ererer - Copy.log
676 /c/Users/Admin/ererer.log
0 /c/Users/Admin/last.log
```

В режиме тихого, понадобится передать аргументы.

Вот так на пример

Конкретная информация представлена в следующей таблице.

| Номер пункта | Количество параметров | Пример                       |
|--------------|-----------------------|------------------------------|
| 1            | 1                     | bash script.sh 1             |
| 2            | 2                     | bash script.sh 2 ‘.php .txt’ |
| 3            | 2                     | bash script.sh 3 ‘.aaa’      |
| 4            | 2                     | bash script.sh 4 ‘.aaa’      |

|    |   |                                |
|----|---|--------------------------------|
| 5  | 1 | bash script.sh 5               |
| 6  | 2 | bash script.sh 6 '.py .c .h'   |
| 7  | 2 | bash script.sh 7 '.bin'        |
| 8  | 2 | bash script.sh 8 '.tst'        |
| 9  | 1 | bash script.sh 9               |
| 10 | 2 | bash script.sh 10 '/home/'     |
| 11 | 2 | bash script.sh 11 '*.txt'      |
| 12 | 1 | bash script.sh 12              |
| 13 | 1 | bash script.sh 13              |
| 14 | 2 | bash script.sh 14 'echo hello' |
| 15 | 1 | bash script.sh 15              |
| 16 | 1 | bash script.sh 16              |

## Листинг исходного кода

```
#!/bin/bash

#проверка на запуск от имени администратора
if [ "$EUID" -eq 0 ]; then
    echo "запуск от имени администратора"
    echo -n "вы хотите продолжить? (y/n): "
    read -r ans
    if ! [ "$ans" == "y" ]; then
        exit 1
    fi
fi

# возвращаются значение temp_file, work_file ,w_dir, cmdline
take_var () {
    temp=$(grep 'temp_file' "$ST"/.myconfig)
    temp_file=( )
    IFS=" "
    for i in $temp; do
        if ! [[ $i == "temp_file=" ]]; then
            temp_file+=( "$i" )
        fi
    done
    work=$(grep "work_file" "$ST"/.myconfig)
    work_file=( )
```

```

IFS=" "
for i in $work; do
    if ! [[ $i == "work_file=" ]]; then
        work_file+=( "$i" )
    fi
done
cmline=$(grep "^commanda" "$ST"/.myconfig)
IFS=":"
for i in $cmline; do
    if ! [[ $i == "commanda" ]]; then
        com=$i
    fi
done

x=$(grep "working_dir" "$ST"/.myconfig)
IFS="="
for i in $x; do
    if ! [[ $x == "working_dir" ]]; then
        w_dir=$i
    fi
done

}

# печать меню
menu () {
echo
take_var
echo -e "temp_file= "${temp_file[@]}"\nwork_file=
"${work_file[@]}"\nworking_dir="${w_dir}"\ncommanda:""$com"
echo
echo 1. просмотреть список расширений временных файлов.
echo 2. задать заново список расширений временных файлов.
echo 3. добавлять конкретное расширение временных файлов.
echo 4. удалять конкретное расширение из списка расширений временных файлов.
echo 5. просмотреть список расширений рабочих файлов.
echo 6. задать заново список расширений рабочих файлов.
echo 7. добавлять конкретное расширение рабочих файлов.
echo 8. удалять конкретное расширение из списка расширений рабочих файлов.
echo 9. просмотреть рабочую папку скрипта
echo 10. задать заново рабочую папку скрипта
echo 11. удалить временные файлы
echo 12. просмотреть записанную команду
echo 13. выполнить записанную команду.
echo 14. изменить записанную команду.
echo 15. просмотреть все строки, ограниченные апострофами, во всех рабочих
файлах.
echo 16. просмотреть объём каждого временного файла
echo 0. Завершить

```



```

}

# Обновление(перезапись) файла .myconfig
update_file () {
    echo -e "temp_file= ""${temp_file[@]}""\nwork_file=
""${work_file[@]}""\nworking_dir=""$w_dir""\ncommanda:""$com" > .myconfig
}

# проверка, существует ли файл .myconfig
if ! [[ -e ./myconfig ]]; then
    touch .myconfig
    temp_file=( .log )
    work_file=( .py )
    w_dir=$(pwd)
    ST=$w_dir
    com="grep def* program.py"
    update_file
fi

# проверяет правильный ли выбор
check_choice () {
    menu
    echo -n "какой выбор: "
    read -r c
    while ! [[ $c -ge 0 && $c -le 16 ]]; do
        echo -n "выбор должен быть от 0 до 9: "
        read -r c
    done
}

# просмотреть список расширений временных файлов
view_temp_files () {
    take_var
    echo "${temp_file[@]}"
}

# задать заново список расширений временных файлов
new_temp_files () {
    take_var
    temp_file=( "$REPLY" )
    update_file
}

# добавлять конкретное расширение временных файлов
add_tfile () {
    take_var
    temp_file+=( "$REPLY" )
}

```

```

    update_file
}

# удалять конкретное расширение из списка расширений временных файлов
remove_tfile () {
    take_var
    num=$((REPLY-1))
    temp_file[num]=""
    update_file
}

# просмотреть список расширений рабочих файлов
view_work_files () {
    take_var
    echo "${work_file[@]}"
}

# задать заново список расширений рабочих файлов
new_work_files () {
    take_var
    work_file=( "$REPLY" )
    update_file
}

# добавлять конкретное расширение рабочих файлов.
add_wfile () {
    take_var
    work_file+=( "$REPLY" )
    update_file
}

# удалять конкретное расширение из списка расширений рабочих файлов.
remove_wfile () {
    take_var
    num=$((REPLY-1))
    work_file[num]=""
    update_file
}

# просмотреть рабочую папку скрипта.
view_working_dir () {
    take_var
    echo "$w_dir"
    check_config
}

# задать заново рабочую папку скрипта.
new_working_dir () {
    take_var
    w_dir=$REPLY

```

```

echo "$w_dir"
if ! [[ -e $w_dir && -d $w_dir ]]; then
    echo "введите неверно!. Введите например ./Documents/ : "
else
    cd "$w_dir"
    update_file 2> /dev/null
fi
}

# удалить временные файлы.
remove_tempfile () {
    take_var
    rm -rf "$REPLY"
    update_file
}

# просмотреть записанную команду
view_cmd () {
    take_var
    echo "$com"
}

# выполнить записанную команду
executecmd () {
    take_var
    echo "$com" > execfile
    chmod +x execfile
    ./execfile 2> /dev/null
    rm execfile
}

# изменить записанную команду
changecmd () {
    take_var
    com=$REPLY
    update_file
}

# просмотреть все строки, ограниченные апострофами, во всех рабочих файлах.
find_strings () {
    take_var
    for extension in "${work_file[@]}"; do
        for file in "$w_dir"/*; do
            if echo "$file" | grep -q "$extension$"; then
                less "$file" | grep -o "'.*'"
            fi
        done
    done
}

```

```

# просмотреть объём каждого временного файла
view_size () {
    take_var
    for extension in "${temp_file[@]}"; do
        for file in "$w_dir"/*; do
            if echo "$file" | grep -q "$extension$"; then
                echo "$(less "$file" | wc -c) $file"
            fi
        done
    done
}

if [ $# -eq 0 ]; then
    check_choice
    while true; do
        if [ "$c" -eq 1 ]; then
            view_temp_files
            check_choice
        elif [ "$c" -eq 2 ]; then
            read -r -p "введите новые расширения через пробел( пример .log
.dat .bin ): "
            new_temp_files
            check_choice
        elif [ "$c" -eq 3 ]; then
            read -r -p "введите новое расширение (пример .log): "
            add_tfile
            check_choice
        elif [ "$c" -eq 4 ]; then
            take_var
            count=1
            for file in "${temp_file[@]}"; do
                echo "$count$file"
                count=$((count+1))
            done
            read -r -p "введите номер, который необходимо удалить: "
            remove_tfile
            check_choice
        elif [ "$c" -eq 5 ]; then
            view_work_files
            check_choice
        elif [ "$c" -eq 6 ]; then
            read -r -p "введите новые расширения через пробел( пример .py .cpp
.txt ): "
            new_work_files
            check_choice
        elif [ "$c" -eq 7 ]; then
            read -r -p "введите новое расширение (пример .log): "
            add_wfile

```

```

        check_choice
    elif [ "$c" -eq 8 ]; then
        take_var
        count=1
        for file in "${work_file[@]}"; do
            echo "$count$file"
            count=$((count+1))
        done
        read -r -p "введите номер, который необходимо удалить: "
        remove_wfile
        check_choice
    elif [ "$c" -eq 9 ]; then
        view_working_dir
        check_choice
    elif [ "$c" -eq 10 ]; then
        read -r -p "введите новую рабочую папку(например /): "
        new_working_dir
        check_choice
    elif [ "$c" -eq 11 ]; then
        take_var
        count=1
        for extension in "${temp_file[@]}"; do
            for file in "$w_dir"/*; do
                if echo "$file" | grep -q "$extension$"; then
                    echo "$file"
                    count=$((count+1))
                fi
            done
        done
        read -r -p "введите название файла для удаления:(например aa.log, *.log): "
        remove_tempfilie
        check_choice
    elif [ "$c" -eq 12 ]; then
        view_cmd
        check_choice
    elif [ "$c" -eq 13 ]; then
        executecmd
        check_choice
    elif [ "$c" -eq 14 ]; then
        read -r -p "введите новую команду: "
        changecmd
        check_choice
    elif [ "$c" -eq 15 ]; then
        find_strings
        check_choice
    elif [ "$c" -eq 16 ]; then
        view_size
        check_choice
    else

```

```

        exit

    fi
done
else
    if [ "$1" -eq 1 ]; then
        view_temp_files
    elif [ "$1" -eq 2 ]; then
        REPLY=$2
        new_temp_files
    elif [ "$1" -eq 3 ]; then
        REPLY=$2
        add_tfile
    elif [ "$1" -eq 4 ]; then
        REPLY=$2
        remove_tfile
    elif [ "$1" -eq 5 ]; then
        view_work_files
    elif [ "$1" -eq 6 ]; then
        REPLY=$2
        new_work_files
    elif [ "$1" -eq 7 ]; then
        REPLY=$2
        add_wfile
    elif [ "$1" -eq 8 ]; then
        REPLY=$2
        remove_wfile
    elif [ "$1" -eq 9 ]; then
        view_working_dir
    elif [ "$1" -eq 10 ]; then
        REPLY=$2
        new_working_dir
    elif [ "$1" -eq 11 ]; then
        REPLY=$2
        remove_tempfile
    elif [ "$1" -eq 12 ]; then
        view_cmd
    elif [ "$1" -eq 13 ]; then
        executecmd
    elif [ "$1" -eq 14 ]; then
        REPLY=$2
        changecmd
    elif [ "$1" -eq 15 ]; then
        find_strings
    elif [ "$1" -eq 16 ]; then
        view_size
    fi
fi

```

