



**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» (ИУ)

КАФЕДРА

«Программное обеспечение ЭВМ и информационные технологии» (ИУ7)

## **ОТЧЁТ ПО УЧЕБНОЙ ПРАКТИКЕ**

Тип практики: Проектно-технологическая

Название предприятия: НУК ИУ МГТУ им. Н. Э. Баумана

Студент:

Кузнецов Денис Евгеньевич, группа ИУ7-13Б

\_\_\_\_\_  
(подпись, дата)

Руководитель от предприятия:

Старший преподаватель ИУ-7

Ломовской Игорь Владимирович

\_\_\_\_\_  
(подпись, дата)

Руководитель от кафедры:

Ассистент кафедры ИУ-7

Кострицкий Александр Сергеевич

\_\_\_\_\_  
(подпись, дата)

Оценка: \_\_\_\_\_

<b>1. ПОСТАНОВКА ЗАДАЧИ .....</b>	<b>3</b>
<b>2. ОСОБЕННОСТИ РЕАЛИЗАЦИИ .....</b>	<b>5</b>
<b>КОНФИГ .....</b>	<b>9</b>
<b>РАБОТА СКРИПТА В ТИХОМ РЕЖИМЕ:.....</b>	<b>10</b>
<b>3. ТЕСТОВЫЕ СЛУЧАИ.....</b>	<b>13</b>
<b>4. ИНСТРУКЦИЯ ПОЛЬЗОВАТЕЛЯ .....</b>	<b>18</b>
<b>5.     ЛИСТИНГ КОДА .....</b>	<b>19</b>

# 1. Постановка 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 рабочих файлов. По умолчанию список состоит из «\*.c».
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 error\* program.c >last.log».

Скрипт должен предоставлять пользователю с помощью меню и текстового интерфейса следующие возможности:

1. Возможность просмотреть, изменить или задать заново список временных файлов.
2. Возможность просмотреть, изменить или задать заново список рабочих файлов.
3. Возможность просмотреть, изменить или задать заново рабочую папку скрипта.
4. Возможность удалить временные файлы.
5. Возможность выполнить или изменить записанную команду.
6. Возможность просмотреть все целые числа во всех рабочих файлах с указанием спецфлага: shorti для чисел от -32000 до 32000, regi для чисел не-shorti от -2000000000 до 2000000000. Достаточно вывести таблицу с указанием местоположений таких констант и их флагов.
7. Возможность просмотреть объём каждого мусорного файла.

Скрипт должен иметь возможность запуска в тихом режиме (без меню), для чего следует использовать позиционные аргументы. Скрипт не должен позволять запуск от имени администратора.

# Особенности реализации

При запуске скрипта сначала делается проверка на запуск от имени администратора. В случае, если произошел запуск скрипта от имени администратора, то скрипт завершает работу и выводит сообщение об ошибке.

```
if [ $EUID -eq 0 ]; then
    echo "Запрещается запускать скрипт суперпользователем"
    exit
fi
```

Листинг 1

Название функции	Назначение функции
menu	Функция для вывода значений переменных по умолчанию из файла setting.myconfig. (работает во время интерактивного режима)
print_list_files	Функция для вывода списка расширений, который пришел ей на вход в виде позиционного аргумента. (работает во время тихого и интерактивного режима)
print_work_dir_script	Функция для вывода абсолютного пути до рабочей директории. (работает во время тихого и интерактивного режима)
complete_command	Функция для выполнения записанной команды, в переменной command записанной в settings.myconfig. (работает во

	время тихого и интерактивного режима)
print_command	Функция для вывода значения переменной command. (работает во время тихого и интерактивного режима)
print_volume_temp_files	Функция для вывода объема всех рабочих файлов в удобном для пользователя формате. (работает во время тихого и интерактивного режима)
update_list_files	Функция для обновления списка расширений, который приходит ей на вход в виде позиционного аргумента. (работает во время интерактивного режима)
add_extension_files	Функция для добавления расширения в список расширений, который приходит ей на вход в виде позиционного аргумента. (работает во время интерактивного режима)
delete_extension_files	Функция для удаления расширения (по номеру), номер расширения и список расширений передаются на вход функции в виде позиционных аргументов. (работает во время итерактивного режима)

change_work_dir_script	Функция для изменения директории рабочей папки. (работает во время интерактивного режима)
delete_temp_files	Функция для удаления временного файла (по имени), имя файла передается на вход в виде позиционного аргумента.
change_command	Функция для изменения значения переменной command записанной в settings.myconfig
print_int_numbers	Функция для просмотра в виде таблицы просмотреть всех целых чисел во всех рабочих файлах с указанием спецфлага: shorti для чисел от -32000 до 32000, regi для чисел не-shorti от -2000000000 до 2000000000. (работает во время тихого и интерактивного режима)
update_list_files_s	Функция для обновления списка расширений, который передается на вход функции в виде позиционного аргумента (работает только во время тихого режима)
add_extension_files_s	Функция для добавления расширения в список расширений, который приходит ей на вход в виде позиционного аргумента (работает только во время тихого режима)

delete_temp_files_s	Функция для удаления временного файла по имени, которое приходит ей на вход в виде позиционного аргумента (работает только во время тихого режима)
delete_extension_files_s	Функция для удаления расширения из списка расширений, который приходит ей на вход в виде позиционного аргумента (работает только во время тихого режима)

Скрипт можно запустить в двух режимах: интерактивном, тихом. Чтобы перейти в тихий режим пользователь должен ввести после названия скрипта первый позиционный аргумент – «-s». Если ввести какое-либо другое значение первого позиционного аргумента, то скрипт продолжит свою работу в интерактивном режиме. Для выхода из интерактивного режима нужно ввести «х». Ниже представлен список дополнительных функций, которые проверяют корректность данных, которые ввел пользователь.

Функция	Назначение функции
check_extension	Функция для проверки, что расширение введено в нужном формате
settings_values	Функция для записи новых значений переменных в конфиг
settings_default_config	Функция для создания конфига с настройками по умолчанию



# Конфиг

В конфиге хранятся 5 переменных:

1. `list_extensions_temp_files` – список расширенный временных файлов
2. `list_extensions_work_files` – список расширений рабочих файлов
3. `wommand` – команда, которую может выполнить скрипт
4. `working_dir` – текущая папка
5. `config_dir` – папка, в которой находится конфиг

```
6. setting_values() {
7.     echo -n "list_extensions_temp_files=\"$1\"
8. list_extensions_work_files=\"$2\"
9. working_dir=\"$3\"
10. command=\"$4\"
11. config_dir=\"$5\" > "$config_dir"
12. }
13.
14. # Создаем конфиг с настройками по умолчанию (В случае, если файл
    отсутствует, он будет создан автоматически)
15. setting_default_config() {
16.     config_dir="$(pwd)/settings.myconfig"
17.     working_dir=$(pwd)
18.     setting_values ".log" ".c" "$(pwd)" "grep error*
    program.c>last.log" "$(pwd)/settings.myconfig"
19. }
20.
21. if [ ! -f settings.myconfig ]; then
22.     touch settings.myconfig
23.     setting_default_config
24. fi
25.
26.
27. # Обновление конфига
28. update_config() {
29.     setting_values "$list_extensions_temp_files"
    "$list_extensions_work_files" "$working_dir" "$command"
    "$config_dir"
30. }
```

Листинг 1

Сначала скрипт заносит в конфиг значения переменных по умолчанию, в случае, если файл конфига отсутствует, то он будет создан автоматически. Функция `update_config` обновляет значения переменных в файле `settings.myconfig`, если значение какой-то из них изменяется.

У скрипта существует два режима работы: интерактивный и тихий. В интерактивном режиме скрипт выводит в консоль сообщения о том, что должен ввести пользователь, а тихий работает без вывода сообщений о том, что должен ввести пользователь. Рассмотрим работу тихого режима.

## Работа скрипта в тихом режиме:

Для запуска скрипта в тихом режиме пользователь в консоли должен ввести полный путь до файла, затем опцию включающую тихий режим, номер исполняемой функции и позиционные аргументы, которые требуются для корректной работы функции.

`'working_dir' -s [1-16] [arg]*` - обобщенная форма записи команды при работе в тихом режиме.

```
if [ $1 = "-s" 2>/dev/null ]; then

    if [ $2 = 1 2>/dev/null ]; then
        print_list_files "$list_extensions_temp_files"
    elif [ $2 = 2 2>/dev/null ]; then
        shift
        shift
        update_list_files_s "$@"
        list_extensions_temp_files=$new_extensions
    elif [ $2 = 3 2>/dev/null ]; then
        add_extension_files_s "$list_extensions_temp_files $3"
        list_extensions_temp_files=$new_extensions
    elif [ $2 = 4 2>/dev/null ]; then
        delete_extensions_files_s "$list_extensions_temp_files $3"
        list_extensions_temp_files=$new_extensions
    fi
fi
```

```

elif [ $2 = 5 2>/dev/null ]; then
    print_list_files "$list_extensions_work_files"
elif [ $2 = 6 2>/dev/null ]; then
    update_list_files_s "$@"
    list_extensions_work_files=$new_extensions
elif [ $2 = 7 2>/dev/null ]; then
    add_extension_files_s "$list_extensions_work_files $3"
    list_extensions_work_files=$new_extensions
elif [ $2 = 8 2>/dev/null ]; then
    delete_extension_files_s "$list_extensions_work_files $3"
    list_extensions_work_files=$new_extensions
elif [ $2 = 9 2>/dev/null ]; then
    print_work_dir_script
elif [ $2 = 10 2>/dev/null ]; then
    change_work_dir_script_s "$3"
    $working_dir=$new_working_dir
elif [ $2 = 11 2>/dev/null ]; then
    delete_temp_files_s "$list_extensions_temp_files $3"
elif [ $2 = 12 2>/dev/null ]; then
    print_command "$command"
elif [ $2 = 13 2>/dev/null ]; then
    complete_command "$command"
elif [ $2 = 14 2>/dev/null ]; then
    change_command "$command"
    command=$new_command
elif [ $2 = 15 2>/dev/null ]; then
    print_int_numbers "$list_extensions_work_files"
elif [ $2 = 16 2>/dev/null ]; then
    print_volume_temp_files "$list_extensions_temp_files"
elif [ $2 = "x" 2>/dev/null ]; then
    exit
else
    echo "Неверно введен второй параметр скрипта!"
fi
update_config
exit

```

Листинг 2

Скрипт проверяет, если первый позиционный аргумент равен ключу, который должен ввести пользователь, чтобы перейти в тихий режим, то

скрипт запускается в тихом. Чтобы скрипт выполнил номер той команды, которая нужна пользователю, то для этого после первого позиционного аргумента (-s) пользователь должен ввести число соответствующее номеру функции, которую надо выполнить. В зависимости от полученного второго позиционного аргумента пользователь должен либо ввести необходимые позиционные аргументы после номера функции, либо нажать “Enter”, и скрипт отработает без дополнительных позиционных параметров.

Пример запуска тихого режима без дополнительных позиционных аргументов:

```
$ ./script.sh -s 1  
Список расширений временных файлов: .log
```

*Пример 1*

Пример запуска тихого режима с дополнительными позиционными аргументами:

```
$ ./script.sh -s 2 .log .t .k .ttt .t  
$ ./script.sh -s 1  
Список расширений временных файлов: .k .log .t .ttt
```

*Пример 2*

В первой строке мы вводим новый список расширений временных файлов, а во второй строчке вызываем функцию (уже без дополнительных позиционных аргументов)

Примечание: дополнительными позиционными аргументами считаются те, которые вводятся после номера выполняемой функции.

В данном скрипте также реализованы функции, если пользователь вводит некорректные данные. В случае, если расширения в списке повторяются, то скрипт автоматически удаляет дубликаты, также удаляются расширения которые имеют форму записи отличную от '.расширение'.

\*Расширения должны обязательно вводиться в форме ‘.расширение’

В данном скрипте не реализована защита, на случай если введенное пользователем расширение, уже есть во другом списке расширений, поэтому пользователь сам должен отслеживать какие именно расширения он вводит.

## Тестовые случаи

Ниже представлены примеры работы различных функций в тихом режиме:

```
$ ./script.sh -s 16
284K      file3.log
0         last.log
```

*Пример 3*

В примере 3 представлена работа функции, которая выводит размеры всех временных файлов.

```
$ ./script.sh -s 15
file3.c:2:125      shorti
file3.c:3:23       shorti
file3.c:4:44       shorti
file3.c:5:6        shorti
file3.c:6:7        shorti
file3.c:6:57       shorti
file3.c:6:45       shorti
file3.c:6:34       shorti
```

*Пример 4*

В примере 4 представлена работа функции, которая выводит все числа в рабочих файлах в формате: название файла: строка файла: число: [shorti/regi]  
Shorti – числа от -32000 до 32000

Regi – числа от -2000000000 до 2000000000, которые при это не shorti

```
$ ./script.sh -s 12
grep error* program.c>last.log
```

```

$ ./script.sh -s 14 "ls -l"
$ ./script.sh -s 12
ls -l
$ ./script.sh -s 13
total 300
-rwxrwxrwx 1 denis denis      0 Dec 19 04:26 file1.c
-rwxrwxrwx 1 denis denis      0 Dec 15 01:22 file2.c
-rwxrwxrwx 1 denis denis    47 Dec 18 01:48 file3.c
-rwxrwxrwx 1 denis denis 289432 Dec 19 21:22 file3.log
-rwxrwxrwx 1 denis denis      0 Dec 19 03:09 grep
-rwxrwxrwx 1 denis denis      0 Dec 19 21:40 last.log
-rwxrwxrwx 1 denis denis 14750 Dec 19 21:39 script.sh
-rwxrwxrwx 1 denis denis    285 Dec 19 21:42
settings.myconfig
Команда успешно выполнена!
$ ./script.sh -s 11 last.log
$ ./script.sh -s 13
total 300
-rwxrwxrwx 1 denis denis      0 Dec 19 04:26 file1.c
-rwxrwxrwx 1 denis denis      0 Dec 15 01:22 file2.c
-rwxrwxrwx 1 denis denis    47 Dec 18 01:48 file3.c
-rwxrwxrwx 1 denis denis 289432 Dec 19 21:22 file3.log
-rwxrwxrwx 1 denis denis      0 Dec 19 03:09 grep
-rwxrwxrwx 1 denis denis 14750 Dec 19 21:39 script.sh
-rwxrwxrwx 1 denis denis    285 Dec 19 21:42
settings.myconfig
Команда успешно выполнена!

```

*Пример 5*

В примере 5 представлена работа функций 11, 12, 13, 14.

Функция 12 выводит значение команды, которую надо выполнить.

С помощью функции 14 изменяем значение команды, которую может выполнить скрипт.

Примечание: следует писать команду всегда в кавычках, чтобы программа всегда распознавала это как один позиционный параметр. Если команда будет состоять из нескольких слов и не будет взята в кавычки, то работа программы не определена. С помощью функции 13 выполняем записанную команду (`ls -l`), из списка выведенных файлов выбираем временный файл и удаляем его с помощью функции 14. В функции 14 реализована проверка на удаление именно временного файла, в случае если файл не временный, то выведется сообщение, что введенное имя некорректное.

Ниже представлены примеры работы в интерактивном режиме:

```
./script.sh
.t .ttt

ls -l
/mnt/c/Users/Denis/Desktop/Лабораторки по проге/project
/mnt/c/Users/Denis/Desktop/Лабораторки по проге/project/settings.myconfig
1) Посмотреть список расширений временных файлов
2) Задать заново список расширений временных файлов
3) Добавить расширение в список расширений временных файлов
4) Удалить расширение из списка временных файлов
5) Посмотреть список расширений рабочих файлов
6) Задать заново список расширений рабочих файлов
7) Добавить расширение в список расширений временных файлов
8) Удалить расширение из списка временных файлов
9) Посмотреть рабочую папку скрипта
10) Задать заново рабочую папку скрипта
11) Удалить временный файл
12) Посмотреть записанную команду
13) Выполнить записанную команду
14) Изменить записанную команду
15) Анализ целых чисел в рабочих файлах
16) Посмотреть объем каждого временного файла
Введите номер функции: 1
Список расширений временных файлов: .t .ttt
Введите номер функции: 6
Вводите расширения в виде: .расширение
```

```
Введите новый список расширений: .с
Введите номер функции: 5
Список расширений временных файлов: .с
Введите номер функции: 8
Введите номер расширения, которое вы хотите удалить: 1
Введите номер функции:
```

*Пример 6*

В данном примере мы видим работу скрипта в интерактивном режиме, первые 5 строк вывода – значения переменных записанных в settings.myconfig. Следующие 16 строк – функции, которые может выполнять скрипт, затем идут примеры исполнения функций скриптом. При вводе расширений выводится подсказка для пользователя, в каком именно формате нужно вводить расширения. Функция 8 реализует удаление расширения временных файлов по номеру, в случае неверно введенного номера расширения реализована проверка, если пользователь неверно вводит номер, то программа выводит соответствующее сообщение об ошибке. В пункте при изменении выполняемой команды не реализованы проверки по безопасности, при команды пользователь должен корректно вводить команду, иначе работа программы не определена. Также пользователь должен понимать смысл введенной команды, например команда `rm script.sh` удаляет весь скрипт, соответственно не следует вводить команды, которые могут навредить работоспособности скрипта.

```
./script.sh
Спиок расширений временных файлов: .log .ttt
Список расширений рабочих файлов: .с
Команда: ls -l
Рабочая директория: /mnt/c/Users/Denis/Desktop/Лабораторки по
проге/project
Директория конфига: /mnt/c/Users/Denis/Desktop/Лабораторки по
проге/project/settings.myconfig
1) Посмотреть список расширений временных файлов
```



- 2) Задать заново список расширений временных файлов
- 3) Добавить расширение в список расширений временных файлов
- 4) Удалить расширение из списка временных файлов
- 5) Посмотреть список расширений рабочих файлов
- 6) Задать заново список расширений рабочих файлов
- 7) Добавить расширение в список расширений рабочих файлов
- 8) Удалить расширение из списка рабочих файлов
- 9) Посмотреть рабочую папку скрипта
- 10) Задать заново рабочую папку скрипта
- 11) Удалить временный файл
- 12) Посмотреть записанную команду
- 13) Выполнить записанную команду
- 14) Изменить записанную команду
- 15) Анализ целых чисел в рабочих файлах
- 16) Посмотреть объем каждого временного файла

Введите номер функции: 1

Список расширений временных файлов: .log .ttt

Введите номер функции: 3

Введите расширение в виде: .расширение

Введите расширение, которое вы хотите добавить: .tt

Введите номер функции: 1

Список расширений временных файлов: .log .tt .ttt

Введите номер функции: 4

Введите номер расширения, которое вы хотите удалить: 2

Введите номер функции: 1

Список расширений временных файлов: .log .ttt

Введите номер функции: 9

Рабочая папка скрипта: /mnt/c/Users/Denis/Desktop/Лабораторки по проге/project

Введите номер функции:

*Пример 7*

В примере 7 представлена реализация добавления и удаления расширения из представленного списка (в данном случае список расширений временных файлов). Также представлена работа 9 функции, которая выводит полный путь до рабочей папки.

## 4. Инструкция пользователя

- 1) Для работы в тихом режиме укажите первый позиционный аргумент –s
- 2) Для исполнения нужной функции введите второй позиционный аргумент (число соответствующее номеру функции из меню)

1	Вывод списка расширений временных файлов
2	Задать заново список расширений временных файлов
3	Добавить расширение в список расширений временных файлов
4	Удалить расширение из списка расширений временных файлов
5	Посмотреть список расширений рабочих файлов
6	Задать заново список расширений рабочих файлов
7	Добавить расширение в список расширений рабочих файлов
8	Удалить расширение из списка расширений рабочих файло
9	Посмотреть рабочую папку скрипта
10	Задать заново рабочую папку скрипта
11	Удалить временный файл
12	Посмотреть записанную команду
13	Выполнить записанную команду
14	Изменить записанную команду
15	Анализ целых числе в рабочих файлах
16	Посмотреть объем каждого временного файла

- 3) При вводе расширений обязательно вводите их в форме: .расширение

- 4) При задавании новой команды обязательно берите команду в кавычки, чтобы bash не распознал ее как несколько разных позиционных аргументов
- 5) При добавлении расширения можно добавить только одно расширение за раз.
- 6) При удалении расширения можно удалить только одно расширение за раз.
- 7) При удалении временного файла можно удалить только один файл за раз (при этом реализована проверка, что будет удален именно временный файл)
- 8) Перед тем, как выполнить команду записанную в переменной command следует вывести ее (с помощью функции номер 12), чтобы убедиться, что эта команда не навредит компьютеру или исполняемому скрипту.
- 9) Объем каждого временного файла выводится в удобных для пользователя единицах.
- 10) При выводе целых чисел на экран сначала выводится название файла, затем строка, в которой расположено число, затем само число, а в конце строки стоит флаг (ключевое слово), по которому можно понять к какому диапазону относится число.
- 11) При работе в интерактивном режим пользователя сопровождают сообщения о том, что следует ввести пользователю.
- 12) Для выхода из интерактивного режима следует ввести вместо номера команды "x".

## 5. Листинг кода

```
#!/bin/bash

if [ $EUID -eq 0 ]; then
    echo "Запрещается запускать скрипт суперпользователем"
    exit
fi

# Вывод меню
```

```

menu() {
    echo "Спиок расширений временных файлов: $list_extensions_temp_files"
    echo "Список расширений рабочих файлов: $list_extensions_work_files"
    echo "Команда: $command"
    echo "Рабочая директория: $working_dir"
    echo "Директория конфига: $config_dir"
    echo ""1) Посмотреть список расширений временных файлов
2) Задать заново список расширений временных файлов
3) Добавить расширение в список расширений временных файлов
4) Удалить расширение из списка временных файлов
5) Посмотреть список расширений рабочих файлов
6) Задать заново список расширений рабочих файлов
7) Добавить расширение в список расширений рабочих файлов
8) Удалить расширение из списка рабочих файлов
9) Посмотреть рабочую папку скрипта
10) Задать заново рабочую папку скрипта
11) Удалить временный файл
12) Посмотреть записанную команду
13) Выполнить записанную команду
14) Изменить записанную команду
15) Анализ целых чисел в рабочих файлах
16) Посмотреть объем каждого временного файла""
}

# ===== функции работающие в тихом и обычном режиме =====

# Функция для проверки существования файла в текущей директории
check_extension() {
    if echo $1 | grep -Eq "^\.[:alnum:][:alnum:]*$"; then
        return 0
    fi
    return 1
}

# функция для обновления переменных в файле
setting_values() {
    echo -n "list_extensions_temp_files=\"$1\"
list_extensions_work_files=\"$2\"
working_dir=\"$3\"
command=\"$4\"
config_dir=\"$5\" > "$config_dir"
}

```

```

# Создаем конфиг с настройками по умолчанию (В случае, если файл
отсутствует, он будет создан автоматически)
setting_default_config(){
    config_dir="$(pwd)/settings.myconfig"
    working_dir=$(pwd)
    setting_values ".log" ".c" "$(pwd)" "grep error* program.c>last.log"
    "$(pwd)/settings.myconfig"
}

if [ ! -f settings.myconfig ]; then
    touch settings.myconfig
    setting_default_config
fi

# Обновление конфига
update_config(){
    setting_values "$list_extensions_temp_files"
    "$list_extensions_work_files" "$working_dir" "$command" "$config_dir"
}

# Функция для просмотра списка расширений
print_list_files(){
    echo "Список расширений временных файлов: $1"
}

# Функция для просмотра рабочей папки скрипта
print_work_dir_script(){
    echo "Рабочая папка скрипта: $working_dir"
}

# Функция для выполнения записанной команды
complete_command(){
    if eval $1 2>/dev/null; then
        echo "Команда успешно выполнена!"
    else
        echo "Невозможно выполнить команду. Во время выполнения возникли
ошибки."
    fi
}

```

```

# Функция для просмотра команды
print_command(){
    echo "Команда: $1"
}

# Вывод объема временных файлов
print_volume_temp_files(){
    for mask in $1; do
        du -h *$mask 2>/dev/null
    done
}

# ===== /// функции работающие в тихом и обычном режиме /// =====

# ===== функции для обычного режима =====

# Функция для обновления списка расширений
update_list_files(){
    echo "Вводите расширения в виде: .расширение"
    read -p "Введите новый список расширений: " start_extensions
    new_extensions=""
    start_extensions=$(echo $start_extensions | xargs -n1 | sort -u |
xargs | sed "s# # #g")
    for extension in $start_extensions; do
        if check_extension "$extension"; then
            new_extensions="$new_extensions $extension"
        fi
    done
}

# Функция для добавления расширения в список расширений
add_extension_files(){
    echo "Введите расширение в виде: .расширение"
    read -p "Введите расширение, которое вы хотите добавить: " extension
    if ! check_extension "$extension"; then
        printf "Некорректный ввод расширения!"
    fi
    new_extensions="$1 $extension"
    new_extensions=$(echo $new_extensions | xargs -n1 | sort -u | xargs |
sed "s# # #g")
}

```

```

# Функция для удаления расширения из списка расширений
delete_extension_files(){
    read -p "Введите номер расширения, которое вы хотите удалить: "
number

    if ! echo $number | grep -Eq "[1-9][0-9]*"; then
        echo "Некорректно введен номер расширения!"
        return
    fi
    is_number="false"
    count=1
    new_extensions=""
    for extension in $1; do
        if [ $number -ne $count ]; then
            new_extensions="$new_extensions $extension"
        else
            is_number="true"
        fi
        count=$((count+1))
    done
    if [ is_number = "false" ]; then
        echo "Расширения с таким номером в списке нет!"
    fi
}

# Функция для изменения рабочей папки скрипта
change_work_dir_script(){
    read -p "Введите новый путь до рабочей папки (можно ввести
относительный или абсолютный путь): " new_working_dir
    if eval cd $new_working_dir 2>/dev/null; then
        return
    else
        echo "Некорректно введен новый путь до рабочей папки!"
    fi
}

# Функция для удаления временных файлов
delete_temp_files(){
    read -p "Введите название временного файла, который нужно удалить: "
name
    flag="false"
    for i in $1; do

```

```

        if ! echo $name | grep -Eq "$1"; then
            flag="false"
        else
            flag="true"
            break
        fi
    done
    if [ flag = "false" ]; then
        echo "Файла с таким именем не найдено!"
    else
        rm $name
    fi
}

# Функция для изменения записанной команды
change_command() {
    read -p "Введите новую команду: " new_command
}

# Функция для анализа целых чисел в рабочих файлах
print_int_numbers() {
    for mask in $1; do
        files=$(ls *$mask)
        for file in $files; do
            positions=$(grep -EoinH [-+]?[0-9]+ $file)
            numbers=$(grep -EoiH [-+]?[0-9]+ $file)
            for index in ${!numbers[@]}; do
                number=${numbers[$index]}
                position=${positions[$index]}
                if [ $number -ge -32000 -a $number -le 32000 2>/dev/null
]; then
                    printf "%-30s%-10s\n" $position "shorti"
                    elif [ $number -ge -2000000000 -a $number -le 2000000000
2>/dev/null ]; then
                    printf "%-30s%-10s\n" $position "regi"
                fi
            done
        done
    done
done

# ===== /// функции для обычного режима /// =====

```



```

# ===== Функции для тихого режима =====

# Обновление списка расширений файлов (тихий режим)
update_list_files_s() {
    new_extensions=""
    for extension in $@; do
        new_extensions="$new_extensions $extension"
    done
    new_extensions=$(echo $new_extensions | xargs -n1 | sort -u | xargs |
sed "s# # #g")
}

# Функция для изменения рабочей директории (тихий режим)
change_work_dir_script_s() {
    if eval cd $3 2>/dev/null; then
        return
    else
        echo "Некорректно введен новый путь до рабочей папки!"
    fi
}

# Добавление нового расширения (тихий режим)
add_extension_files_s() {
    new_extensions="$1 $2"
    new_extensions=$(echo $new_extensions | xargs -n1 | sort -u | xargs |
sed "s# # #g")
    echo $new_extensions
}

# Удаление файла из списка расширений (тихий режим)
delete_temp_files_s() {
    flag="false"
    for i in $1; do
        if ! echo $2 | grep -Eq "$i"; then
            flag="false"
        else
            flag="true"
            break
        fi
    done
    if [ flag = "false" ]; then
        echo "Файла с таким именем не найдено!"
    else

```

```

        rm $2
    fi
}

# Удаление элемента из списка расширений (тихий режим)
delete_extension_files_s() {
    count=1
    flag="false"
    new_extensions=""
    for extension in $1; do
        if [ $count -ne $2 ]; then
            new_extensions="$new_extensions $extension"
        else
            flag="true"
        fi
    done
    if [ flag = "false" ]; then
        echo "Некорректно введен номер расширения!"
    fi
}

# Изменение команды (тихий режим)
change_command_s() {
    new_command=$2
}

. settings.myconfig

# Выполнение основного скрипта
if [ $1 = "-s" 2>/dev/null ]; then

    if [ $2 = 1 2>/dev/null ]; then
        print_list_files "$list_extensions_temp_files"
    elif [ $2 = 2 2>/dev/null ]; then
        shift
        shift
        update_list_files_s "$@"
        list_extensions_temp_files=$new_extensions
    elif [ $2 = 3 2>/dev/null ]; then
        add_extension_files_s "$list_extensions_temp_files" "$3"
        list_extensions_temp_files=$new_extensions
    elif [ $2 = 4 2>/dev/null ]; then

```

```

        delete_extensions_files_s "$list_extensions_temp_files" "$3"
        list_extensions_temp_files=$new_extensions
    elif [ $2 = 5 2>/dev/null ]; then
        print_list_files "$list_extensions_work_files"
    elif [ $2 = 6 2>/dev/null ]; then
        update_list_files_s "$@"
        list_extensions_work_files=$new_extensions
    elif [ $2 = 7 2>/dev/null ]; then
        add_extension_files_s "$list_extensions_work_files" "$3"
        list_extensions_work_files=$new_extensions
    elif [ $2 = 8 2>/dev/null ]; then
        delete_extension_files_s "$list_extensions_work_files" "$3"
        list_extensions_work_files=$new_extensions
    elif [ $2 = 9 2>/dev/null ]; then
        print_work_dir_script
    elif [ $2 = 10 2>/dev/null ]; then
        change_work_dir_script_s "$3"
        $working_dir=$new_working_dir
    elif [ $2 = 11 2>/dev/null ]; then
        delete_temp_files_s "$list_extensions_temp_files" "$3"
    elif [ $2 = 12 2>/dev/null ]; then
        print_command "$command"
    elif [ $2 = 13 2>/dev/null ]; then
        complete_command "$command"
    elif [ $2 = 14 2>/dev/null ]; then
        change_command_s "$command" "$3"
        command=$new_command
    elif [ $2 = 15 2>/dev/null ]; then
        print_int_numbers "$list_extensions_work_files"
    elif [ $2 = 16 2>/dev/null ]; then
        print_volume_temp_files "$list_extensions_temp_files"
    elif [ $2 = "x" 2>/dev/null ]; then
        exit
    else
        echo "Неверно введен второй параметр скрипта!"
    fi
    update_config
    exit
else
    menu
    while true; do

```

```

read -p 'Введите номер функции: ' choice
if [ $choice = 1 2>/dev/null ]; then
    print_list_files "$list_extensions_temp_files"
elif [ $choice = 2 2>/dev/null ]; then
    update_list_files
    list_extensions_temp_files=$new_extensions
elif [ $choice = 3 2>/dev/null ]; then
    add_extension_files "$list_extensions_temp_files"
    list_extensions_temp_files=$new_extensions
elif [ $choice = 4 2>/dev/null ]; then
    delete_extension_files "$list_extensions_temp_files"
    list_extensions_temp_files=$new_extensions
elif [ $choice = 5 2>/dev/null ]; then
    print_list_files "$list_extensions_work_files"
elif [ $choice = 6 2>/dev/null ]; then
    update_list_files
    list_extensions_work_files=$new_extensions
elif [ $choice = 7 2>/dev/null ]; then
    add_extension_files "$list_extensions_work_files"
    list_extensions_work_files=$new_extensions
elif [ $choice = 8 2>/dev/null ]; then
    delete_extension_files "$list_extensions_temp_files"
    list_extensions_temp_files=$new_extensions
elif [ $choice = 9 2>/dev/null ]; then
    print_work_dir_script
elif [ $choice = 10 2>/dev/null ]; then
    change_work_dir_script
    $working_dir=$new_working_dir
elif [ $choice = 11 2>/dev/null ]; then
    delete_temp_files
elif [ $choice = 12 2>/dev/null ]; then
    print_command "$command"
elif [ $choice = 13 2>/dev/null ]; then
    complete_command "$command"
elif [ $choice = 14 2>/dev/null ]; then
    change_command "$command"
    command=$new_command
elif [ $choice = 15 2>/dev/null ]; then
    print_int_numbers "$list_extensions_work_files"
elif [ $choice = 16 2>/dev/null ]; then
    print_volume_temp_files "$list_extensions_temp_files"
elif [ $choice = "x" 2>/dev/null ]; then

```

```
        exit
    else
        echo 'Некорректно введен номер функции!'
    fi
    update_config
done
fi
```

*Листинг всего кода*