|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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: Вольняга Максим Юрьевич ИУ7-16Б | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (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: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*2021г*

Оглавление

[Постановка задачи 3](#_Toc90849718)

[Особенности реализации 5](#_Toc90849719)

[Тестовые случаи 8](#_Toc90849720)

[Ограничения по безопасности 12](#_Toc90849722)

[Инструкции пользователя 13](#_Toc90849723)

[Тихий режим 13](#_Toc90849724)

[Интерактивный режим 14](#_Toc90849725)

[Листинг исходного кода 15](#_Toc90849726)

# 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 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 объем каждого временного 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

В начале кода происходит объявление переменных расширений, рабочей директории по умолчанию и пользовательской программы. Затем эти переменные передаются в массив для удобства в перезаписи.

* Пример объявления переменных :

config\_temp="\*.log"

config\_work="\*.py"

config\_work\_dir="./"

config\_programm="grep def\* program.py >last.log"

*# Массив содержащий конфигурации по умолчанию*

config=( "$config\_temp" "$config\_work" "$config\_work\_dir" "$config\_programm" )

Запрещается выполнение скрипта от имени администратора. У пользователя root идентификатор UID равен 0 .

* Пример проверки UID:

**if** [[ $UID == 0 ]]; **then**

echo "Вы зашли под админом, вход запрещен"

exit

**fi**

Далее происходит работа с файлом “con.myconfig”. Скрипт ищет файл “con.myconfig” в той директории, в которой лежит скрипт. Если такого файла не существует, то создается новый файл, в который записываются переменные по умолчанию.

* Пример поиска файла “con.myconfig” :

**if** [[ -f "./con.myconfig" ]]; **then**

i=0

**while** read line; **do**

config\_now[$i]="$line"

i=**$((**$i+1**))**

**done** < ./con.myconfig

**else**

config\_rec "${config[@]}"

**fi**

* Пример файла “.myconfig” по умолчанию :

\*.log

\*.py

./

grep def\* program.py >last.log

Далее в скрипте объявляются функции, которые будут использоваться в интерактивном и тихом режиме

Список функций и их назначение :

|  |  |
| --- | --- |
| Название функции | Назначение функции |
| view\_edit\_temporary \_rec | Просмотр и изменение рабочих файлов |
| view\_edit\_workdir | Просмотр и изменение рабочей папки |
| view\_edit\_programm | Просмотр и изменение программы |
| apostrophe | Просмотр всех строк, ограниченных апострофами, во всех рабочих файлах |
| trash | Просмотр объёма каждого мусорного файла |
| del\_temp | Удаление определенного расширения из списка временных файлов |
| del\_work | Удаление определенного расширения из списка рабочих файлов |

Функции del\_work, del\_temp требуют аргумента – строку, остальные нет. Все функции кроме (apostrophe, trash) после выполнения вызывают функцию config\_rec, которая перезаписывает “con.myconfig”.

* Пример функции config\_rec

config\_rec(){

IFS\_OLD=$IFS

IFS=""

echo -n > ./con.myconfig

**for** i in "$@"; **do**

echo $i >> ./con.myconfig

**done**

IFS=$IFS\_OLD

Тихий режим выполняется с помощью позиционных параметров, передаваемых пользователем. После ввода “-s ” в зависимости от ключа произойдет отработка различных действий (все ключи в инструкции пользователя)

* Пример тихого режима :

**if** ! [ $# -eq 0 ]; **then**

**if** [ "$1" == "-s" ]; **then**

**if** [ "$2" == "-v\_temp" ]; **then**

echo "${config\_now[0]}"

Если тихий режим не выбран пользователем, запускается интерактивный режим. Он выполнен с помощью бесконечного цикла, из которого пользователь может выйти при помощи выбора меню – 0.

* Пример интерактивного меню и его вывод :

**while** :

**do**

echo " "

echo "Введите [0-9] для выбора меню

1. Просмотреть или задать заново список временных файлов.

2. Добавить или удалить конкретное расширение из списка временных файлов.

3. Просмотреть или задать заново список рабочих файлов.

4. Добавить или удалить конкретное расширение из списка рабочих файлов.

5. Просмотреть, изменить или задать заново рабочую папку скрипта.

6. Удалить временные файлы.

7. Выполнить или изменить записанную команду.

8. Просмотреть все строки, ограниченные апострофами, во всех рабочих файлах

9. Просмотреть объём каждого мусорного файла.

0. Выйти

"

read menu

# Тестовые случаи

* Генерация конфигурационного файла при первом запуске:

max@max-VirtualBox:~/indi/r$ ls

scrip.sh

max@max-VirtualBox:~/indi/r$ ./scrip.sh -s -v\_tmp

max@max-VirtualBox:~/indi/r$ ls

con.myconfig scrip.sh

max@max-VirtualBox:~/indi/r$

## Тихий режим:

1. Просмотр и изменение временых файлов

max@max-VirtualBox:~/indi$ ./scrip.sh -s -v\_temp

\*.log .ff

max@max-VirtualBox:~/indi$ ./scrip.sh -s -d\_temp .ff

max@max-VirtualBox:~/indi$ ./scrip.sh -s -v\_temp

\*.log

1. Удаление конкретного расширения из списка временных файлов

max@max-VirtualBox:~/indi$ ./scrip.sh -s -v\_temp

\*.log .ff

max@max-VirtualBox:~/indi$ ./scrip.sh -s -d\_temp .ff

max@max-VirtualBox:~/indi$ ./scrip.sh -s -v\_temp

\*.log

1. Просмотр и изменение рабочих файлов

max@max-VirtualBox:~/indi$ ./scrip.sh -s -e\_work .txt .py .pdf

max@max-VirtualBox:~/indi$ ./scrip.sh -s -v\_work

.txt .py .pdf

1. Добавление и удаление конкретного расширения из списка рабочих файлов

max@max-VirtualBox:~/indi$ ./scrip.sh -s -v\_work

\*.py

max@max-VirtualBox:~/indi$ ./scrip.sh -s -a\_work .txt

max@max-VirtualBox:~/indi$ ./scrip.sh -s -v\_work

\*.py .txt

max@max-VirtualBox:~/indi$ ./scrip.sh -s -d\_work .txt

max@max-VirtualBox:~/indi$ ./scrip.sh -s -v\_work

\*.py

1. Просмотр и изменение рабочей папки скрипта

max@max-VirtualBox:~/indi$ ./scrip.sh -s -v\_workdir

./

max@max-VirtualBox:~/indi$ ./scrip.sh -s -e\_workdir ./indi/r

max@max-VirtualBox:~/indi$ ./scrip.sh -s -v\_workdir

./indi/r

1. Удаление всех временных файлы

max@max-VirtualBox:~/indi$ ls

1.log 2.log 3.log r scrip.sh

max@max-VirtualBox:~/indi$ ./scrip.sh -s -del\_all\_temp

max@max-VirtualBox:~/indi$ ls

con.myconfig r scrip.sh

1. Выполнение и изменение программы

max@max-VirtualBox:~/indi$ ls

con.myconfig program.py r scrip.sh

max@max-VirtualBox:~/indi$ ./scrip.sh -s -prog

program.py:def abc

program.py:def cde

program.py:def ggg

max@max-VirtualBox:~/indi$ ./scrip.sh -s -e\_prog du

max@max-VirtualBox:~/indi$ ./scrip.sh -s -prog

76 ./r

104 .

1. Просмотр всех строки, ограниченные апострофами

max@max-VirtualBox:~/indi$ ./scrip.sh -s -apostrophe

'erfrffr'

max@max-VirtualBox:~/indi$ cat program.py

def abc

def cde

def ggg

"efefreferfref"

'erfrffr'

""ewrferef

"erfreerfrf"

1. Просмотр объёма каждого мусорного файла.

max@max-VirtualBox:~/indi$ ./scrip.sh -s -trash

0 ./111.log

4,0K ./11.log

4,0K ./1.log

Поведение скрипта при работе в интерактивном режиме идентично работе в тихом режиме, поэтому рассмотрим лишь некоторые пункты:

* Пример использования интерактивного режима :
  1. Добавить конкретное расширение из списка рабочих файлов
  2. Удалить конкретное расширение из списка рабочих файлов
  3. Просмотр списока рабочих файлов

max@max-VirtualBox:~/indi$ ./scrip.sh

Введите [0-9] для выбора меню

1. Просмотреть или задать заново список временных файлов.

2. Добавить или удалить конкретное расширение из списка временных файлов.

3. Просмотреть или задать заново список рабочих файлов.

4. Добавить или удалить конкретное расширение из списка рабочих файлов.

5. Просмотреть, изменить или задать заново рабочую папку скрипта.

6. Удалить временные файлы.

7. Выполнить или изменить записанную команду.

8. Просмотреть все строки, ограниченные апострофами, во всех рабочих файлах

9. Просмотреть объём каждого мусорного файла.

0. Выйти

4

Введите [1-2] для выбора

1. Добавить конкретное расширение

2. Удалить конкретное расширение

1

Текущий список рабочих расширени: program.py

Введите расширение: .txt

Введите [0-9] для выбора меню

1. Просмотреть или задать заново список временных файлов.

2. Добавить или удалить конкретное расширение из списка временных файлов.

3. Просмотреть или задать заново список рабочих файлов.

4. Добавить или удалить конкретное расширение из списка рабочих файлов.

5. Просмотреть, изменить или задать заново рабочую папку скрипта.

6. Удалить временные файлы.

7. Выполнить или изменить записанную команду.

8. Просмотреть все строки, ограниченные апострофами, во всех рабочих файлах

9. Просмотреть объём каждого мусорного файла.

0. Выйти

4

Введите [1-2] для выбора

1. Добавить конкретное расширение

2. Удалить конкретное расширение

2

Текущий список рабочих расширени: program.py .txt

Введите расширение: .txt

Введите [0-9] для выбора меню

1. Просмотреть или задать заново список временных файлов.

2. Добавить или удалить конкретное расширение из списка временных файлов.

3. Просмотреть или задать заново список рабочих файлов.

4. Добавить или удалить конкретное расширение из списка рабочих файлов.

5. Просмотреть, изменить или задать заново рабочую папку скрипта.

6. Удалить временные файлы.

7. Выполнить или изменить записанную команду.

8. Просмотреть все строки, ограниченные апострофами, во всех рабочих файлах

9. Просмотреть объём каждого мусорного файла.

0. Выйти

3

Введите [1-2] для выбора

1. Просмотреть список рабочих файлов

2. Задать заново список рабочих файлов

1

program.py

# Ограничения по безопасности

Для использование скрипта в тихом режиме необходимо передать в качестве первого аргумента ключ *“-s”.* В тихом режиме можно вызвать только одну команду через пробел. Например, *“*./scrip.sh -s -prog*”*

В случае если ключ тихого режима не был передан, скрипт завершает работу.

При изменении рабочего каталога у пользователя должны быть права на чтение и открытие новой директории, которая уже должна существовать. Работа с папкой происходит не рекурсивно

Не валидные запросы, при которых поведение скрипта не определено :

1. Записанная пользовательская команда содержит ошибки, имеет недопустимое выполнение или нехватку прав.
2. Путь к рабочей директории недоступен к чтению или указан неверно.
3. Временные или рабочие расширения записаны не через пробел

# Инструкции пользователя

1. Для выполнения тихого режима, передайте ключ

***“-s”*** при вызове скрипта.

1. Затем передайте один из предложенных ниже ключей, для выбора операции
2. Для работы в интерактивном режиме, просто совершите вызов[0-9] скрипта без передачи ключей.
3. Для удаления расширений (-d\_temp, - d\_work) необходимо вводить названия самих расширений.
4. В пунктах 1-4, имеется возможность вводить множество расширений разделяя их пробелом.
5. Для изменения пути к рабочей папке, указывается относительный или абсолютный путь. Работа с папкой происходит не рекурсивно.

## Тихий режим

|  |  |
| --- | --- |
| Ключ | Название |
| -v\_temp | Просмотр временных файлов |
| -e\_temp | Изменение временых файлов |
| -a\_temp | Добавление конкретного расширения из списка временных файлов |
| -d\_temp | Просмотр всех строк, ограниченных апострофами, во всех рабочих файлах |
| -v\_work | Просмотр изменение рабочих файлов |
| -e\_work | Удаление определенного расширения из списка временных файлов |
| -a\_work | Добавление конкретного расширения из списка рабочих файлов |
| -d\_work | Удаление конкретного расширения из списка рабочих файлов |
| -v\_workdir | Просмотр рабочей папки скрипта |
| -e\_workdir | Изменение рабочей папки скрипта |
| -del\_all\_temp | Удаление всех временных файлы |
| -prog | Выполнение программы |
| -e\_prog | Изменение программы |
| -apostrophe | Просмотр всех строки, ограниченные апострофами |
| -trash | Просмотр объёма каждого мусорного файла. |

## Интерактивный режим

Меню с описанием команд :

1. Просмотреть или задать заново список временных файлов.

2. Добавить или удалить конкретное расширение из списка временных файлов.

3. Просмотреть или задать заново список рабочих файлов.

4. Добавить или удалить конкретное расширение из списка рабочих файлов.

5. Просмотреть, изменить или задать заново рабочую папку скрипта.

6. Удалить временные файлы.

7. Выполнить или изменить записанную команду.

8. Просмотреть все строки, ограниченные апострофами, во всех рабочих файлах

9. Просмотреть объём каждого мусорного файла.

0. Выйти

На вход принимается соответствующий номер команды [0-9]. В некоторых пунктах (1,2,3,4,5) потребуется дополнительный ввод, например номер подкоманды или текстовое значение.

# Листинг исходного кода

*#!/bin/bash*

*# Вольняга Максим ИУ7-16Б (ВАР-2)*

*# Выход из скрипта, если пользователь - админ*

**if** [[ $UID == 0 ]]; **then**

echo "Вы зашли под админом, вход запрещен"

exit

**fi**

work\_dir=**$(**dirname "$0"**)** *# Расположение скрипта*

cd $work\_dir *# Изменяю положение пользователя*

config\_temp="\*.log"

config\_work="\*.py"

config\_work\_dir="./"

config\_programm="grep def\* program.py >last.log"

*# Массив содержащий конфигурации по умолчанию*

config=( "$config\_temp" "$config\_work" "$config\_work\_dir" "$config\_programm" )

config\_now=("${config[@]}") *# Текущая конфигурация*

*# Запись конфигурации в con.myconfig*

config\_rec(){

IFS\_OLD=$IFS

IFS=""

echo -n > ./con.myconfig

**for** i in "$@"; **do**

echo $i >> ./con.myconfig

**done**

IFS=$IFS\_OLD

}

*# Если файл конфигурации существует, запоминаем конфигурацию, иначе создаем новый файл*

**if** [[ -f "./con.myconfig" ]]; **then**

i=0

**while** read line; **do**

config\_now[$i]="$line"

i=**$((**$i+1**))**

**done** < ./con.myconfig

**else**

config\_rec "${config[@]}"

**fi**

*# Просмотр и изменение временных файлов*

**function** view\_edit\_temporary {

echo "Введите [1-2] для выбора

1. Посмотреть список временных файлов

2. Задать заново"

read choice

**if** [ $choice -eq 1 ]; **then**

echo " "

echo **${**config\_now[0]**}**

**elif** [ $choice -eq 2 ]; **then**

config\_now[0]=""

echo -n "Введите расширения через пробел: "

read -a arr\_extension *# Ввод массива состоящий из расширений*

config\_now[0]=**${**arr\_extension[@]**}** *# Изменяем текущую конфигурацию для расшишений*

config\_rec "${config\_now[@]}" *# Перезаписываем конфигурацию в con.myconfig*

**else**

echo "Вы ввели не верный номер "

**fi**

**return**

}

*# Просмотр и изменение рабочих файлов*

**function** view\_edit\_work {

echo "Введите [1-2] для выбора

1. Просмотреть список рабочих файлов

2. Задать заново список рабочих файлов"

echo " "

read choice

**if** [ $choice -eq 1 ]; **then**

echo " "

echo **${**config\_now[1]**}**

**elif** [ $choice -eq 2 ]; **then**

config\_now[1]=""

echo "Введите расширения через пробел: "

read -a arr\_work *# Ввод массива состоящий из расширений*

config\_now[1]=**${**arr\_work[@]**}**

config\_rec "${config\_now[@]}"

**else**

echo "Вы ввели не верный номер "

**fi**

**return**

}

*# Просмотр и изменение рабочей папки*

**function** view\_edit\_workdir {

echo "Введите [1-2] для выбора

1. Посмотреть рабочую папку скрипта

2. Задать заново рабочую папку скрипта "

echo " "

read choice

**if** [ $choice -eq 1 ]; **then**

echo " "

echo **${**config\_now[2]**}**

**elif** [ $choice -eq 2 ]; **then**

echo "Введите новый путь к рабочей папке: "

read path\_new *# Считываем строку - путь к рабочей папке*

config\_now[2]=$path\_new

config\_rec "${config\_now[@]}"

**else**

echo "Вы ввели не верный номер "

**fi**

**return**

}

*# Просмотр и изменение программы*

**function** view\_edit\_programm {

echo "Введите [1-2] для выбора

1. Выполнить команду

2. Изменить команду"

echo " "

read choice

**if** [ $choice -eq 1 ]; **then**

eval "${config\_now[3]}" *# Выполняем команду*

**elif** [ $choice -eq 2 ]; **then**

echo "Введите новую команду: "

read command\_new *# Считываем команду - строка*

config\_now[3]=$command\_new

config\_rec "${config\_now[@]}"

**else**

echo "Вы ввели не верный номер "

**fi**

**return**

}

*# Просмотр всех строк, ограниченных апострофами, во всех рабочих файлах*

**function** apostrophe() {

**for** i in "${config\_now[1]}"; **do**

grep "^'[[:print:]]\*'$" **${**config\_now[2]**}**$i *# При помощи регулярного выражения выводим нужные строки*

**done**

**return**

}

*# Просмотр объёма каждого мусорного файла*

**function** trash {

**for** i in "${config\_now[0]}"; **do**

du -h **${**config\_now[2]**}**$i *# При помощи команды du выводим объем мусорных файлов*

**done**

**return**

}

*# Удаление определенного расширения из списка временных файлов*

del\_temp() {

str\_extension\_new="" *# Строка, которая хранит все расширения кроме удаляемой*

**for** i in **${**config\_now[0]**}**; **do**

**if** [[ "$i" != "$str\_extension" ]]; **then**

str\_extension\_new+=" "$i *# Добавляем все расширения кроме удаляемой*

**fi**

**done**

config\_now[0]=$str\_extension\_new

config\_rec "${config\_now[@]}"

**return**

}

*# Удаление определенного расширения из списка рабочих файлов*

del\_work() {

str\_work\_new=""

**for** i in **${**config\_now[1]**}**; **do**

**if** [[ "$i" != "$str\_work" ]]; **then**

str\_work\_new+=" "$i

**fi**

**done**

config\_now[1]=$str\_work\_new

config\_rec "${config\_now[@]}"

**return**

}

*# Тихий режим*

**if** ! [ $# -eq 0 ]; **then**

**if** [ "$1" == "-s" ]; **then**

**if** [ "$2" == "-v\_temp" ]; **then** *# Просмотр временных файлов*

echo "${config\_now[0]}"

**elif** [ "$2" == "-e\_temp" ]; **then** *# Изменение временных файлов*

config\_now[0]=""

count=0

**for** i in "$@"; **do**

**if** [[ $count > 1 ]]; **then** *# Записываем расширения в 1 эл массива*

config\_now[0]+=" ""$i"

config\_rec "${config\_now[@]}" *# Обновляем конфиг*

**fi**

count=**$((**count+1**))**

**done**

**elif** [ "$2" == "-a\_temp" ]; **then** *# Добавление конкретного расширения из списка временных файлов*

config\_now[0]+=" ""$3" *# Изменяем текущую конфигурацию для расширений*

config\_rec "${config\_now[@]}"

**elif** [ "$2" == "-d\_temp" ]; **then** *# Удаление конкретного расширения из списка временных файлов*

str\_extension="$3"

del\_temp "$str\_extension"

**elif** [ "$2" == "-v\_work" ]; **then** *# Просмотр рабочих файлов*

echo **${**config\_now[1]**}**

**elif** [ "$2" == "-e\_work" ]; **then** *# Изменение рабочих файлов*

config\_now[1]=""

count=0

**for** i in "$@"; **do**

**if** [[ $count > 1 ]]; **then**

config\_now[1]+=" ""$i"

config\_rec "${config\_now[@]}"

**fi**

count=**$((**count+1**))**

**done**

**elif** [ "$2" == "-a\_work" ]; **then** *# Добавление конкретного расширения из списка рабочих файлов*

config\_now[1]+=" ""$3"

config\_rec "${config\_now[@]}"

**elif** [ "$2" == "-d\_work" ]; **then** *# Удаление конкретного расширения из списка временных файлов*

str\_work="$3"

del\_work "$str\_work"

**elif** [ "$2" == "-v\_workdir" ]; **then** *# Просмотр рабочей папки скрипта*

echo **${**config\_now[2]**}**

**elif** [ "$2" == "-e\_workdir" ]; **then** *# Изменение рабочей папки скрипта*

config\_now[2]="$3"

config\_rec "${config\_now[@]}"

**elif** [ "$2" == "-del\_all\_temp" ]; **then** *# Удаление всех временных файлы*

**for** i in **${**config\_now[0]**}**; **do**

rm -f **${**config\_now[2]**}**/**${**i**}**

**done**

**elif** [ "$2" == "-prog" ]; **then** *# Выполнение записанной команды.*

**${**config\_now[3]**}**

**elif** [ "$2" == "-e\_prog" ]; **then** *# Изменяем записанную команду*

config\_now[3]="$3"

config\_rec "${config\_now[@]}"

**elif** [ "$2" == "-apostrophe" ]; **then** *# Просмотр всех строки, ограниченные апострофами*

apostrophe

**elif** [ "$2" == "-trash" ]; **then** *# Просмотр объёма каждого мусорного файла.*

trash

**fi**

**fi**

exit

**fi**

*# Меню*

**while** :

**do**

echo " "

echo "Введите [0-9] для выбора меню

1. Просмотреть или задать заново список временных файлов.

2. Добавить или удалить конкретное расширение из списка временных файлов.

3. Просмотреть или задать заново список рабочих файлов.

4. Добавить или удалить конкретное расширение из списка рабочих файлов.

5. Просмотреть, изменить или задать заново рабочую папку скрипта.

6. Удалить временные файлы.

7. Выполнить или изменить записанную команду.

8. Просмотреть все строки, ограниченные апострофами, во всех рабочих файлах

9. Просмотреть объём каждого мусорного файла.

0. Выйти

"

read menu

**if** [ $menu -eq 0 ]; **then** *# Выход из программы*

echo "Bye-bye"

exit

**fi**

*# 1. Просмотреть или задать заново список временных файлов.*

**if** [ $menu -eq 1 ]; **then**

view\_edit\_temporary

*# 2. Добавить или удалить конкретное расширение из списка временных файлов*

**elif** [ $menu -eq 2 ]; **then**

echo "Введите [1-2] для выбора

1. Добавить конкретное расширение

2. Удалить конкретное расширение "

echo " "

read choice

**if** [ $choice -eq 1 ]; **then**

echo "Текущий список временных расширении: ${config\_now[0]}"

echo -n "Введите расширение: "

read str\_extension *# Ввод строки расширения*

config\_now[0]+=" "$str\_extension

config\_rec "${config\_now[@]}"

**elif** [ $choice -eq 2 ]; **then**

echo "Текущий список временных расширении: ${config\_now[0]}"

echo -n "Введите расширение: "

read str\_extension

del\_temp "$str\_extension"

config\_now[0]=$str\_extension\_new

config\_rec "${config\_now[@]}"

**else**

echo "Вы ввели не верный номер "

**fi**

*# 3. Просмотреть или задать заново список рабочих файлов.*

**elif** [ $menu -eq 3 ]; **then**

view\_edit\_work

*# 4. Добавить или удалить конкретное расширение из списка рабочих файлов.*

**elif** [ $menu -eq 4 ]; **then**

echo "Введите [1-2] для выбора

1. Добавить конкретное расширение

2. Удалить конкретное расширение "

echo " "

read choice

**if** [ $choice -eq 1 ]; **then**

echo "Текущий список рабочих расширении: ${config\_now[1]}"

echo -n "Введите расширение: "

read str\_work

config\_now[1]+=" "$str\_work

config\_rec "${config\_now[@]}"

**elif** [ $choice -eq 2 ]; **then**

echo "Текущий список рабочих расширении: ${config\_now[1]}"

echo -n "Введите расширение: "

read str\_work

del\_work "$str\_work"

**else**

echo "Вы ввели не верный номер "

**fi**

*# 5. Просмотреть, изменить или задать заново рабочую папку скрипта.*

**elif** [ $menu -eq 5 ]; **then**

view\_edit\_workdir

*# 6. Удалить временные файлы.*

**elif** [ $menu -eq 6 ]; **then**

**for** i in **${**config\_now[0]**}**; **do**

rm -f **${**config\_now[2]**}**/**${**i**}** *# f Удаление без вопросов*

**done**

*# 7. Выполнить или изменить записанную команду.*

**elif** [ $menu -eq 7 ]; **then**

view\_edit\_programm

*# 8. Просмотреть все строки, ограниченные апострофами, во всех рабочих файлах*

**elif** [ $menu -eq 8 ]; **then**

apostrophe

*# 9. Просмотреть объём каждого мусорного файла.*

**elif** [ $menu -eq 9 ]; **then**

trash

**else**

echo "

Вы ввели неверное значения, для выбора меню

Введите заново"

**fi**

**done**