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

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
```

2. 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
```

3. 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
```

4. 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
```

5. 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
```

6. 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
```

7. 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 .
```

8. 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"
```

9. 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 списка 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.
2) Затем передайте один из предложенных ниже ключей, для выбора операции
3) Для работы в интерактивном режиме, просто совершите вызов[0-9] скрипта без передачи ключей.
4) Для удаления расширений (-d_temp, - d_work) необходимо вводить названия самих расширений.
5) В пунктах 1-4, имеется возможность вводить множество расширений разделяя их пробелом.
6) 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
```