

# Отчет о выполненном проекте

---

Выполнил Тихоненко Артём, ИУ7-15Б

Вариант 1

Проверяющие:

Кострицкий Александр Сергеевич

Ломовской Игорь Владимирович

## Содержание

---

- [Постановка задачи](#)
- [Особенности реализации](#)
- [Тестовые случаи](#)
- [Инструкции пользователя](#)
- [Листинг исходного кода](#)

## Постановка задачи

---

Скрипт командной оболочки для обработки текстовых файлов.

У скрипта есть список расширений временных файлов, список расширений рабочих файлов, команда для выполнения. Все эти данные записаны в файле «conf.myconfig».

По умолчанию conf.myconfig выглядит так:

```
*.log
*.txt
grep "error*" last.txt >last.log
```

- список расширений временных файлов состоит из «\*.log».
- список расширений рабочих файлов состоит из «\*.txt».
- команда выглядит как «grep "error\*" last.txt > last.log».

Скрипт предоставляет пользователю с помощью меню и текстового интерфейса следующие возможности:

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 или изменить рабочую папку скрипта.
6. Удалить временные файлы.
7. Выполнить или изменить записанную команду.
8. Просмотреть число строк и слов в каждом рабочем файле.
9. Просмотреть объём каждого временного файла.

Скрипт имеет три режима работы:

- Интерактивный `./project.sh --interactive`
- «Тихий» `./project.sh --action [0-16]? [0-16]*`, позволяет выполнить последовательность команд, вводя в интерактивном режиме только аргументы функций.
- Справка `./project.sh --help`, печатает список режимов и команд

## Особенности реализации

---

- НЕ работает выполнение некоторых команд, например, дефолтной. `ls -la` работает. Кажется, это связано с перенаправлениями вывода.
- Все операции с файлом конфигурации через `sed`
- При каждой операции файл конфигурации читается заново
- Есть проверка безопасности только на (не)запуск от имени администратора
- Есть проверка номера команды в интерактивном и тихом режимах

- Выход из интерактивного режима с помощью команды `0`
- Тихий режим не истинно "тихий", но можно передать сразу последовательность команд

## Тестовые случаи

---

Протестирована каждая команда в интерактивном режиме, также протестирован переход в другую директорию и работа всех команд в ней.

Т.к. список расширений временных/рабочих файлов не связан строго с записанной командой, выполнение команды тестировалось без привязки к спискам расширений.

В ходе тестирования выявлено, что выполнение некоторых команд не работает. При этом, если эти же команды ввести вручную в терминале, всё работает. Кажется, это связано с перенаправлениями вывода, т.к. `grep` ругается на `no such file ">"`.

Тихий режим тестировался менее внимательно, т.к. он просто обертка над теми же функциями, что есть в интерактивном.

## Инструкции пользователя

---

Программа имеет три режима работы:

1. Интерактивный - запуск через `./project.sh --interactive`
2. Тихий (номер(а) команд передаются в аргументах) - запуск через `./project.sh --action [0-16]? [0-16]*`
3. Показ справки - запуск через `./project.sh --help`

Список команд:

0. Завершение программы

Работа с временными файлами

1. Перезадать список расширений ВРЕМЕННЫХ файлов
2. Добавить расширение в список расширений ВРЕМЕННЫХ файлов
3. Удалить расширение из списка расширений ВРЕМЕННЫХ файлов (по имени расширения)

Работа с рабочими файлами

4. Перезадать список расширений РАБОЧИХ файлов
5. Добавить расширение в список расширений РАБОЧИХ файлов
6. Удалить расширение из списка расширений РАБОЧИХ файлов (по имени расширения)

Директория

7. Сменить рабочую директорию

Работа с файлами

8. Удалить все ВРЕМЕННЫЕ файлы
9. Исполнить команду из файла конфигурации
10. Изменить выполняемую команду
11. Показать кол-во слов и строк для каждого РАБОЧЕГО файла
12. Показать размер каждого ВРЕМЕННОГО файла
13. Показать записанную команду для выполнения

Скрыты в интерактивном режиме, но доступны в тихом

14. Показать список расширений временных файлов
15. Показать список расширений рабочих файлов
16. Показать рабочую директорию

## Листинг исходного кода

---

```
#!/bin/bash
```

```
# Проверка структуры файла конфигурации
```

```

checkConfig() {
    echo "config check not implemented"
    return
}

# Создание нового файла конфигурации, если его не было или он повре
initConfig() {
    touch conf.myconfig
    echo '*.log ' > conf.myconfig
    echo '*.txt ' >> conf.myconfig
    echo 'grep "error*" last.txt > last.log' >> conf.myconfig
}

# Проверка существования и корректности файла конфигурации
startupConfig() {
    if [ -e "conf.myconfig" ]; then
        if [ checkConfig ]; then
            echo "$(tput setaf 2)config file found $(tput sgr 0)"
        else
            echo "config file is damaged, restoring default state"
            initConfig
        fi
    else
        echo "$(tput setaf 2) config file created $(tput sgr 0)"
        initConfig
    fi
}

# Проверка корректности ввода номера команды
# params: comm
commCorrect() {
    if ! echo "$comm" | grep -Eq "^[0-9][0-9]*$"; then
        return 1
    elif [ "$comm" -ge 0 -a "$comm" -le 16 ]; then
        return 0
    else
        return 1
    fi
}

# Показывает список расширений временных файлов
showTmp() {
    extensions=$(sed -n '1p' "conf.myconfig")
    echo "Расширения временных файлов: $extensions"
}

```

```

# Вбить список расширений временных файлов заново
redefineTmp() {
    echo -n 'Введите новый список расширений временных файлов: '
    read new_tmp
    new_tmp=$new_tmp" "
    sed -i '' "1s/.*/$new_tmp/g" 'conf.myconfig'
}

# Добавить расширение в список расширений временных файлов
addTmp() {
    prev_tmp=$(sed -n '1p' 'conf.myconfig')
    echo "Список расширений сейчас: $prev_tmp"
    echo -n 'Введите расширение, которое надо добавить: '
    read new_tmp
    tmp=$prev_tmp$new_tmp" "
    sed -i '' "1s/.*/$tmp/g" 'conf.myconfig'
}

# Удалить расширение из списка расширений временных файлов
removeTmp() {
    prev_tmp=$(sed -n '1p' 'conf.myconfig')
    echo "Список расширений сейчас: $prev_tmp"
    echo -n 'Введите расширение, которое надо удалить: '
    read del_tmp
    del_tmp=$del_tmp" "
    sed -i '' "1s/$del_tmp//g" 'conf.myconfig'
}

# Показать список расширений рабочих файлов
showWorking() {
    extensions=$(sed -n '2p' 'conf.myconfig')
    echo "Расширения рабочих файлов: $extensions"
}

# Перезадать список расширений рабочих файлов
redefineWorking() {
    echo -n 'Введите новый список расширений временных файлов: '
    read new_wrk
    new_wrk=$new_wrk" "
    sed -i '' "2s/.*/$new_wrk/g" 'conf.myconfig'
}

```

```

# Добавить расширение в список расширений рабочих файлов
addWorking() {
    prev_wrk=$(sed -n '2p' 'conf.myconfig')
    echo "Список расширений сейчас: $prev_wrk"
    echo -n 'Введите расширение, которое надо добавить: '
    read new_wrk
    wrk=$prev_wrk$new_wrk "
    sed -i '' "2s/./$wrk/g" 'conf.myconfig'
}

# Удалить расширение из списка расширений рабочих файлов
removeWorking() {
    prev_wrk=$(sed -n '2p' 'conf.myconfig')
    echo "Список расширений сейчас: $prev_wrk"
    echo -n 'Введите расширение, которое надо удалить: '
    read del_wrk
    del_wrk=$del_wrk" "
    sed -i '' "2s/$del_wrk//g" 'conf.myconfig'
}

# Показывает текущую рабочую директорию скрипта
showCWD() {
    echo "Текущий каталог: $(PWD)"
}

# Изменяет текущую рабочую директорию скрипта
changeCWD() {
    prev_wd=$(PWD)
    echo "Текущий полный путь: $prev_wd"
    echo -n 'Введите полный или относительный путь новой директории (
    read newwd
    cd $newwd
    cp -n "$prev_wd/conf.myconfig" "$(PWD)/conf.myconfig"
}

# Удаляет все файлы, подходящие по расширению как "временные"
cleanupTmp() {
    extensions=$(sed -n '1p' 'conf.myconfig')
    for ext in $extensions; do
        find . -type f -name "$ext" -delete
    done
}

```

*# Исполняет команду в рабочей директории*

```
execProg() {  
    prog=$(sed -n '3p' 'conf.myconfig')  
    $prog  
}
```

*# Изменяет исполняемую команду*

```
changeProg() {  
    prog=$(sed -n '3p' 'conf.myconfig')  
    echo "Текущая команда: $prog"  
    echo -n "Введите новую команду: "  
    read newcomm  
    sed -i '' "3s/.*/$newcomm/g" 'conf.myconfig'  
}
```

*# (вспомогательная) Показывает кол-во строк и слов в текстовом файле*

```
showLinesAndWords() {  
    linesdirty=$(wc -l $filename)  
    lines=($linesdirty)  
    wordsdirty=$(wc -w $filename)  
    words=($wordsdirty)  
    echo "$filename words: $words lines: $lines"  
}
```

*# Для каждого файла, подходящего по расширению как "рабочий", показ*

```
showForEveryWorking() {  
    extensions=$(sed -n '2p' 'conf.myconfig')  
    for ext in $extensions; do  
        files=$(find . -type f -name "$ext")  
        if [ -z "$files" ]; then  
            echo "Файлов с расширением $ext не нашлось"  
        fi  
        for file in $files; do  
            filename=$file  
            showLinesAndWords  
        done  
    done  
}
```

*# Для каждого файла, подходящего по расширению как "временный", пок*

```
showJunkSize() {  
    extensions=$(sed -n '1p' 'conf.myconfig')  
    for ext in $extensions; do  
        files=$(find . -type f -name "$ext")  
        if [ -z "$files" ]; then
```



```

        echo "Файлов с расширением $ext не нашлось"
    fi
    for file in $files; do
        du $file
    done
done
}

# Показать записанную программу
showProg() {
    prog=$(sed -n '3p' 'conf.myconfig')
    echo "Записанная команда: $prog"
}

printmenu() {
    echo "${tput setaf 6}1.${tput setaf 3) Перезадать список расшир
    echo "${tput setaf 6}2.${tput setaf 3) Добавить расширение в спи
    echo "${tput setaf 6}3.${tput setaf 3) Удалить расширение из спи
    echo "${tput setaf 6}4.${tput setaf 3) Перезадать список расшире
    echo "${tput setaf 6}5.${tput setaf 3) Добавить расширение в спи
    echo "${tput setaf 6}6.${tput setaf 3) Удалить расширение из спи
    echo "${tput setaf 6}7.${tput setaf 3) Сменить рабочую директори
    echo "${tput setaf 6}8.${tput setaf 3) Удалить все ВРЕМЕННЫЕ фай
    echo "${tput setaf 6}9.${tput setaf 3) Исполнить команду из файл
    echo "${tput setaf 6}10.${tput setaf 3) Изменить выполняемую ком
    echo "${tput setaf 6}11.${tput setaf 3) Показать кол-во слов и с
    echo "${tput setaf 6}12.${tput setaf 3) Показать размер каждого
    echo "${tput setaf 6}13.${tput setaf 3) Показать записанную кома
    echo "${tput setaf 1}0.${tput setaf 3) Завершение программы ${tp
}

printFullMenu() {
    echo "1. Перезадать список расширений ВРЕМЕННЫХ файлов"
    echo "2. Добавить расширение в список расширений ВРЕМЕННЫХ файлов
    echo "3. Удалить расширение из списка расширений ВРЕМЕННЫХ файлов
    echo "4. Перезадать список расширений РАБОЧИХ файлов"
    echo "5. Добавить расширение в список расширений РАБОЧИХ файлов"
    echo "6. Удалить расширение из списка расширений РАБОЧИХ файлов (
    echo "7. Сменить рабочую директорию"
    echo "8. Удалить все ВРЕМЕННЫЕ файлы"
    echo "9. Исполнить команду из файла конфигурации"
    echo "10. Изменить выполняемую команду"
    echo "11. Показать кол-во слов и строк для каждого РАБОЧЕГО файла"
    echo "12. Показать размер каждого ВРЕМЕННОГО файла"
    echo "13. Показать записанную команду"
    echo "14. Показать список ВРЕМЕННЫХ файлов"
    echo "15. Показать список РАБОЧИХ файлов"
}

```

```
    echo "16. Показать рабочую директорию"
    echo "0. Завершение программы"
}
```

```
commandSelector() {
    commm=$1
    if [ $commm -eq 1 ]; then
        redefineTmp
    elif [ $commm -eq 2 ]; then
        addTmp
    elif [ $commm -eq 3 ]; then
        removeTmp
    elif [ $commm -eq 4 ]; then
        redefineWorking
    elif [ $commm -eq 5 ]; then
        addWorking
    elif [ $commm -eq 6 ]; then
        removeWorking
    elif [ $commm -eq 7 ]; then
        changeCWD
    elif [ $commm -eq 8 ]; then
        cleanupTmp
    elif [ $commm -eq 9 ]; then
        execProg
    elif [ $commm -eq 10 ]; then
        changeProg
    elif [ $commm -eq 11 ]; then
        showForEveryWorking
    elif [ $commm -eq 12 ]; then
        showJunkSize
    elif [ $commm -eq 13 ]; then
        showProg
    elif [ $commm -eq 14 ]; then
        showTmp
    elif [ $commm -eq 15 ]; then
        showWorking
    elif [ $commm -eq 16 ]; then
        showCWD
    elif [ $commm -eq 0 ]; then
        echo "Завершение"
        exit
    fi
}
```

```
# Проверка, от чьего имени запущена программа
```

```

if [ "$(id -u)" -eq 0 ]; then
    echo 'Данный скрипт нельзя запустить от имени администратора' >&2
    exit 1
fi

name=`basename $0`

# Работа в интерактивном режиме
if [ "$1" == "--help" ]; then # вывод справки
    echo "Использование: ./$name --interactive"
    echo "        либо ./$name --action [0-16]? [0-16]*"
    echo "        либо ./$name --help , чтобы показать это сообщение"
    printFullMenu
elif [ "$1" == "--interactive" ]; then
    startupConfig
    printmenu
    showTmp
    showWorking
    showCWD

    echo -n "$(tput setaf 4)Введите команду: $(tput sgr 0)"
    read comm
    until commCorrect; do
        echo -n "$(tput setaf 1)Неверная команда,$(tput sgr 0) $(tput s
        read comm
    done

    while [ "$comm" -ne 0 ]; do
        commandSelector $comm
        printmenu
        showTmp
        showWorking
        showCWD
        echo -n "$(tput setaf 4)Введите команду: $(tput sgr 0)"
        read comm
        until commCorrect; do
            echo -n "$(tput setaf 1)Неверная команда,$(tput sgr 0) $(tput
            read comm
        done
    done
    echo "$(tput setaf 2)Завершение. $(tput sgr 0)"
elif [ "$1" == "--action" ]; then # Работа в тихом режиме
    startupConfig
    comm=$2
    until [ "$comm" == "" ]; do
        if commCorrect; then
            echo "$(tput setaf 3)Выполнение команды $comm$(tput sgr 0)"
            commandSelector $comm
        else

```

```

        echo "${tput setaf 1}Неверный номер команды:${tput sgr 0} $co
        exit 1
    fi
    shift
    comm=$2
done
else # неправильный первый параметр
    echo "${tput setaf 1}Использование: ./$name --interactive" >&2
    echo "        либо ./$name --action [0-16]? [0-16]*" >&2
    echo "        либо ./$name --help , чтобы показать это сообщени
    exit 1
fi

```