

```

echo 'Hello, world!'

# Проверка структуры файла конфигурации
checkConfig() {
    echo "config check not implemented"
    return
}

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

# Проверка существования и корректности файла конфигурации
startupConfig() {
    if [ -e "conf.myconfig" ]; then
        echo "config file exists"
        if [ $(checkConfig) ]; then
            echo "config file is OK"
            wd=$(sed -n '4p' 'conf.myconfig')
            cd $wd
        else
            echo "config file is damaged, restoring default state"
            initConfig
        fi
    else
        echo "no config file"
        initConfig
    fi
}

# Проверка корректности ввода номера команды
# params: comm
commCorrect() {
    if ! echo "$comm" | grep -Eq "^[0-9][0-9]*$"; then
        return 1
    elif [ "$comm" -ge 1 -a "$comm" -le 15 ]; then
        return 0
    else
        return 1
    fi
}

```

```

# 1 Показывает список расширений временных файлов
showTmp() {
    sed -n '1p' 'conf.myconfig'
}

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

# 3 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'
}

# 4 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'
}

# 5 Показать список расширений рабочих файлов
showWorking() {
    sed -n '2p' 'conf.myconfig'
}

# 6 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'
}

```

```

# 7 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'
}

# 8 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'
}

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

# 10 Изменяет текущую рабочую директорию скрипта
changeCWD() {
    prev_wd=$(sed -n '4p' 'conf.myconfig')
    echo "Текущий полный путь: $prev_wd"
    echo -n 'Введите полный или относительный путь новой директории (
    read newwd
    cd $newwd
    cp "$prev_wd/conf.myconfig" "$(PWD)/conf.myconfig"
    sed -i '' "4s/$prev_wd/$newwd/g" 'conf.myconfig'
}

# 11 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
}

# 12 Исполняет команду в рабочей директории
execProg() {
    prog=$(sed -n '3p' 'conf.myconfig')
    exec $prog
}

```

13 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'  
}
```

14 Показывает кол-во строк и слов в текстовом файле

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

15 Для каждого файла, подходящего по расширению как "рабочий", по

```
showForEveryWorking() {  
    extensions=$(sed -n '2p' 'conf.myconfig')  
    for ext in $extensions; do  
        files=$(find . -type f -name "$ext")  
        for file in $files; do  
            filename=$file  
            showLinesAndWords  
        done  
    done  
}
```

16 Для каждого файла, подходящего по расширению как "временный",

```
showJunkSize() {  
    extensions=$(sed -n '1p' 'conf.myconfig')  
    for ext in $extensions; do  
        files=$(find . -type f -name "$ext")  
        for file in $files; do  
            du $file  
        done  
    done  
}
```

```
printmenu() {  
    echo "0. Завершение программы"  
    echo "1. Показать список расширений ВРЕМЕННЫХ файлов"  
    echo "2. Перезадать список расширений ВРЕМЕННЫХ файлов"  
    echo "3. Добавить расширение в список расширений ВРЕМЕННЫХ файлов"  
    echo "4. Удалить расширение из списка расширений ВРЕМЕННЫХ файлов"
```

```

echo "5. Показать список расширений РАБОЧИХ файлов"
echo "6. Перезадать список расширений РАБОЧИХ файлов"
echo "7. Добавить расширение в список расширений РАБОЧИХ файлов"
echo "8. Удалить расширение из списка расширений РАБОЧИХ файлов (
echo "9. Показать текущую рабочую директорию"
echo "10. Сменить рабочую директорию (! конфигурация копируется в
echo "11. Удалить все ВРЕМЕННЫЕ файлы"
echo "12. Исполнить команду из файла конфигурации"
echo "13. Изменить выполняемую команду"
echo "14. Показать кол-во слов и строк для каждого РАБОЧЕГО файла
echo "15. Показать размер каждого ВРЕМЕННОГО файла"
}

```

```

silentArgsCorrect() {
    echo "not implemented"
}

```

```

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

```

```

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 ./$name --action [1-15]"
    echo "           либо ./$name --help , чтобы показать это сообщени
    printmenu
elif [ "$1" == "--interactive" ]; then
    startupConfig
    printmenu
    echo -n "Введите команду: "
    read comm
    while [ "$comm" -ne 0 ]; do
        until commCorrect; do
            echo -n 'Неверная команда, повторите ввод: '
            read comm
        done
        commandSelector $comm
        printmenu
        echo -n "Введите команду: "
        read comm
    done
    echo "Завершение."
elif [ "$1" == "--action" ]; then # Работа в тихом режиме
    startupConfig
    comm=$2
    if commCorrect; then
        commandSelector $comm
        echo "Завершение"
    else
        echo "Неверный номер команды" >&2
        exit 1
    fi
else # неправильный первый параметр
    echo "Использование: ./$name --interactive" >&2
    echo "           либо ./$name --action [1-15]" >&2
    echo "           либо ./$name --help , чтобы показать это сообщени

```

```
    exit 1  
fi
```