

Темирканов Кантемир ИУ7-23Б

Отчет по заданию №1 в рамках вычислительного практикума

Автоматизация функционального тестирования

1. Реализованы скрипты отладочной и релизной сборок.

```
#!/bin/bash

gcc -std=c99 -Wall -Werror -Wpedantic -Wextra -Wfloat-equal -Wfloat-
conversion -g3 -c ./main.c
gcc -o ./main.exe ./main.c -lm
```

```
#!/bin/bash

gcc -std=c99 -Wall -Werror -Wpedantic -Wextra -Wfloat-equal -Wfloat-
conversion -c ./main.c
gcc -o ./main.exe ./main.o -lm
```

2. Реализован скрипт очистки побочных файлов.

```
#!/bin/bash

rm ./main.exe ./main.o ./main.c.gcov ./main.gcno ./main.gcda
```

3. Реализован компаратор для сравнения последовательностей действительных чисел, располагающихся в двух текстовых файлах, с игнорированием остального содержимого.

```
#!/bin/bash

f1=$1
f2=$2
touch f1_num.txt f2_num.txt
grep -oE '[+-]?[0-9]+([.][0-9]+)?' "$f1" > f1_num.txt
grep -oE '[+-]?[0-9]+([.][0-9]+)?' "$f2" > f2_num.txt
if cmp f1_num.txt f2_num.txt; then
    exit 1
else
    exit 0
fi
```

4. Реализован компаратор для сравнения содержимого двух текстовых файлов, располагающегося после первого вхождения подстроки «Result: _».

```
#!/bin/bash

f1=$1
f2=$2
touch f1_num.txt f2_num.txt
if grep -o -i 'Result: ' "$f1" | wc -l -ne 1
then
    exit 1
else
    grep -oE "Result: .*" "$f1" > f1_num.txt
    grep -oE "Result: .*" "$f2" > f2_num.txt
    if cmp f1_num.txt f2_num.txt; then
        exit 0
    fi
fi
```

```
        else
            exit 1
        fi
    fi
```

5. Реализован скрипт pos_case.sh для проверки позитивного тестового случая по определённым далее правилам. С содержанием внутри опционального запуска приложения в оболочке valgrind для проверки тестовых случаев.

```
#!/bin/bash

if [ "$1" = "USE_VALGRIND" ]; then #Проверка на то , поднят ли глобальный
флаг USE_VALGRIND,если да, то все тестовые прогоны проводятся в оболочке
valgrind, иначе – как обычно.
    file_stream_in=$2
    file_stream_out_expect=$3
    touch pos_out.txt
    if [ -n "$4" ]; then
        file_app_args=$4
    else
        file_app_args=""
    fi
    valgrind --log-file=log.txt --quiet ../../main.exe "$file_app_args" <
"$file_stream_in" > pos_out.txt
    codv=$?
else
    file_stream_in=$1
    file_stream_out_expect=$2
    touch pos_out.txt
    if [ -n "$3" ]; then
        file_app_args=$3
    else
        file_app_args=""
    fi
    ../../main.exe "$file_app_args" < "$file_stream_in" > pos_out.txt
    codv=$?
fi

if [ "$codv" != "0" ]; then
    exit 1
else
    ./comparator.sh pos_out.txt "$file_stream_out_expect"
    point=$?
    if [ "$point" != "0" ]; then
        exit 1
    else
        exit 0
    fi
fi
```

6. Реализован скрипт neg_case.sh для проверки негативного тестового случая по определённым далее правилам. С содержанием внутри опционального запуска приложения в оболочке valgrind для проверки тестовых случаев.

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

```

if [ "$1" = "--USE-VALGRIND" ]; then #Проверка на то , поднят ли глобальный
    флаг USE_VALGRIND,если да, то все тестовые прогоны проводятся в оболочке
    valgrind, иначе – как обычно.
    file_stream_in=$2
    touch neg_out.txt
    if [ -n "$3" ]; then
        file_app_args=$3
    else
        file_app_args=""
    fi
    valgrind --log-file=log.txt --quiet ../../main.exe "$file_app_args" <
"$file_stream_in" > neg_out.txt
    codv=$?
else
    file_stream_in=$1
    touch pos_out.txt
    if [ -n "$2" ]; then
        file_app_args=$2
    else
        file_app_args=""
    fi
    ../../main.exe "$file_app_args" < "$file_stream_in" > neg_out.txt
    codv=$?
fi

if [ "$codv" != "0" ]; then
    exit 1
else
    exit 0
fi

```

7. Обеспечение автоматизации функционального тестирования.

```

#!/bin/bash
# запись имен тестовых файлов в массивы чтобы потом бегать по массиву и
извлекать данные
arr_pos_in=()
while IFS=' ' read -r line; do arr_pos_in+=("$line"); done < <(ls
../data/pos_[0-9][1-9]_in.txt)

arr_pos_out=()
while IFS=' ' read -r line; do arr_pos_out+=("$line"); done < <(ls
../data/pos_[0-9][1-9]_out.txt)

arr_neg_in=()
while IFS=' ' read -r line; do arr_neg_in+=("$line"); done < <(ls
../data/neg_[0-9][1-9]_in.txt)

success_pos_count=0
failure_pos_count=0

success_neg_count_memory=0
failure_neg_count_memory=0

success_pos_count_memory=0
failure_pos_count_memory=0

count_pos_elem=${#arr_pos_in[@]}
count_neg_elem=${#arr_neg_in[@]}

```

```

success_neg_count=0
failure_neg_count=0
# Проверка на флаг Валгринд
if [ "$1" == "USE_VALGRIND" ]; then
    # Проходимся по тестовым файлам и передаем их в скрипты для тестирования
    for (( item=0 ; item<count_pos_elem ; item++))
    do
        ./pos_case.sh "${arr_pos_in[item]}" "${arr_pos_out[item]}"
        codv=$?
        if [ $codv -ne 0 ]; then
            success_pos_count=$((success_pos_count + 1))
        else
            failure_pos_count=$((failure_pos_count + 1))
        fi
        if [ -s log.txt ]; then
            failure_pos_count_memory=$((failure_pos_count_memory + 1))
        else
            success_pos_count_memory=$((success_pos_count_memory + 1))
        fi
    done

    for (( item=0 ; item<count_neg_elem ; item++))
    do
        ./neg_case.sh "${arr_neg_in[item]}"
        codv=$?
        if [ $codv -ne 0 ]; then
            success_neg_count=$((success_neg_count + 1))
        else
            failure_neg_count=$((failure_neg_count + 1))
        fi

        if [ -s log.txt ]; then
            failure_neg_count_memory=$((failure_neg_count_memory + 1))
        else
            success_neg_count_memory=$((success_neg_count_memory + 1))
        fi
    done

else
    for (( item=0 ; item<count_pos_elem ; item++))
    do
        ./pos_case.sh "${arr_pos_in[item]}" "${arr_pos_out[item]}"
        codv=$?
        if [ $codv -ne 0 ]; then
            success_pos_count=$((success_pos_count + 1))
        else
            failure_pos_count=$((failure_pos_count + 1))
        fi
    done

    for (( item=0 ; item<count_neg_elem ; item++))
    do
        ./neg_case.sh "${arr_neg_in[item]}"
        codv=$?
        if [ $codv -ne 0 ]; then
            success_neg_count=$((success_neg_count + 1))
        else

```

```

        failure_neg_count=$((failure_neg_count + 1))
    fi

done

fi

#Вывод результатов тестирующей системы
if [ "$1" == "USE_VALGRIND" ]; then

    echo "Positive tests: correct $success_pos_count of $count_pos_elem;
memory passed $success_pos_count_memory of $count_pos_elem"
    if [ "$count_neg_elem" -eq 0 ]; then
        echo "There are no negative 'in' files"
    else
        echo "Negative tests: correct $success_neg_count of $count_neg_elem;
memory passed $success_neg_count_memory of $count_neg_elem"
    fi
else
    echo "Positive tests: correct $success_pos_count of $count_pos_elem"

    if [ "$count_neg_elem" -eq 0 ]; then
        echo "There are no negative 'in' files"
    else
        echo "Negative tests: correct $success_neg_count of $count_neg_elem"
    fi
fi
fi

```

9. Автоматизация получения статистики полноты данных при тестировании.

```

#!/bin/bash

gcc -std=c99 -Wall -Werror main.c --coverage -O0 -o main.exe -lm
cd func_tests/scripts || exit
./func_tests.sh > /dev/null #ВЫВОД В КОРЗИНУ
cd ../..
gcov main.exe

```