**ОГЛАВЛЕНИЕ**

- ВВЕДЕНИЕ........................................................................................................... 2

+ Задание...................................................................................................................2

+ Цель работы ……………………………………………………………………..2

- ОПИСАНИЕ СКРИПТОВ.....................................................................................3

+ Скрипт build\_debug.sh..........................................................................................4

+ Скрипт build\_debug\_asan.sh.................................................................................5

+ Скрипт build\_debug\_msan.sh................................................................................6

+ Скрипт build\_debug\_ubsan.sh...............................................................................7

+ Скрипт build\_release.sh ........................................................................................8

+ Скрипт check\_scripts.sh ........................................................................................9

+ Скрипт clean.sh .....................................................................................................9

+ Скрипт collect\_coverage.sh ................................................................................10

+ Скрипт comparator.sh..........................................................................................11

+ Скрипт func\_tests.sh ...........................................................................................14

+ Скрипт neg\_case.sh ............................................................................................ 16

+ Скрипт pos\_case.sh............................................................................................. 17

- ЗАКЛЮЧЕНИЕ .................................................................................................. 18

+ разработан........................................................................................................... 18

- СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .........................................19

**ВВЕДЕНИИ**

**Цель работы**

Целью данной работы является автоматизация процессов сборки и тестирования.

**Задание**

Требуется:  
1. Реализовать скрипты отладочной и релизной сборок.  
2. Реализовать скрипты отладочной сборки с санитайзерами. 3. Реализовать скрипт очистки побочных файлов.

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

5. Реализовать компаратор для сравнения содержимого двух текстовых файлов, располагающегося после первого вхождения подстроки «Result:␣».

6. Реализовать скрипт pos\_case.sh для проверки позитивного тестового случая по определённым далее правилам.

7. Реализовать скрипт neg\_case.sh для проверки негативного тестового случая по определённым далее правилам.

8. Обеспечить автоматизацию функционального тестирования.

**ОПИСАНИЕ СКРИПТОВ**

Организации проекта:

/lab\_ LL \_ PP \_ CC /

main.c

build\_debug.sh

build\_release.sh

collect\_coverage.sh

clean.sh

/func\_tests/ readme.md /scripts/

func\_tests.sh

neg\_case.sh

pos\_case.sh

comparator.sh

/data/

pos\_01\_in.txt

pos\_01\_out.txt

pos\_02\_in.txt

...

pos\_05\_in.txt

pos\_05\_out.txt

neg\_01\_in.txt

...

где LL— номер лабораторной, PP — номер задачи, CC — вариант студента

**ОПИСАНИЕ СКРИПТОВ**

**Скрипт build\_debug.sh**

#!/bin/bash

if gcc -c -std=c99 -Wall -Werror -Wpedantic -Wextra -Wfloat-equal -Wfloat-conversion main.c; then

echo "Compilation successful"

if gcc main.o -o app.exe -lm; then

echo "Linking successful"

else

echo "Linking failed" >&2

exit 1

fi

else

echo "Compilation failed" >&2

exit 1

fi

Предназначен для компиляции программ с отладочной информацией

**Скрипт build\_debug\_asan.sh**

#!/bin/bash

if ! clang -std=c99 -Wall -Werror -Wpedantic -Wextra -Wfloat-equal -Wfloat-conversion -fsanitize=address -fno-omit-frame-pointer -g main.c -c -o main.o; then

echo "Compilation failed. Exiting..."

exit 1

else

echo "Compilation successful"

fi

if clang main.o -o app.exe -lm; then

echo "Linking successful"

else

echo "Linking failed. Exiting..."

exit 1

fi

Предназначен для компиляции программ с отладочной информацией

**Скрипт build\_debug\_msan.sh**

#!/bin/bash

if ! clang -std=c99 -Wall -Werror -Wpedantic -Wextra -Wfloat-equal -Wfloat-conversion -fsanitize=memory -fPIE -pie -fno-omit-frame-pointer -g main.c -o app\_ms.exe; then

echo "Compilation failed. Exiting..."

exit 1

else

echo "Compilation successful"

fi

if ! clang main.c -o app.exe -lm; then

echo "Linking failed. Exiting..."

exit 1

else

echo "Linking successful"

fi

Предназначен для компиляции программ с отладочной информацией

**Скрипт build\_debug\_ubsan.sh**

#!/bin/bash

if clang -std=c99 -Wall -Werror -Wpedantic -Wextra -Wfloat-equal -Wfloat-conversion -fsanitize=undefined -fno-omit-frame-pointer -g main.c; then

echo "Compilation successful"

else

echo "Compilation failed. Exiting..."

exit 1

fi

if clang main.c -o app.exe -lm; then

echo "Linking successful"

else

echo "Linking failed. Exiting..."

exit 1

fi

Предназначен для компиляции программ с отладочной информацией

**Скрипт build\_release.sh**

#!/bin/bash

if gcc -c -std=c99 -Wall -Werror -Wpedantic -Wextra -Wfloat-equal -Wfloat-conversion -Wvla -O3 main.c -o main.o; then

echo "Compilation successful"

if gcc main.o -o app.exe -lm; then

echo "Linking successful"

else

echo "Linking failed" >&2

exit 1

fi

else

echo "Compilation failed" >&2

exit 1

fi

Предназначен для компилирование программ при выпуске.

**Скрипт check\_scripts.sh**

#!/bin/bash

find . -name "\*.sh" -print0 | xargs -0 shellcheck

Предназначен для проверки shellcheck.

**Скрипт clean.sh**

#!/bin/bash

find . -maxdepth 1 -type f -name '.exe' -o -name '.o' -o -name '.out' -o -name '.gcno' -o -name '.gcda' -o -name '.gcov' -delete

**Скрипт collect\_coverage.sh**

#!/bin/bash

if gcc -c -std=c99 -Wall -Werror -Wpedantic -Wextra -Wfloat-equal -Wfloat-conversion main.c --coverage -O0; then

echo "Compilation successful"

else

echo "Compilation failed. Exiting..."

exit 1

fi

if gcc main.o -o app.exe -lm --coverage; then

echo "Linking successful"

else

echo "Linking failed. Exiting..."

exit 1

fi

cd ./func\_tests/scripts || { echo "Directory not found. Exiting..."; exit 1; }

./func\_tests.sh > ../func\_tests\_output.txt || { echo "Functional tests failed. Exiting..."; exit 1; }

cat func\_tests\_output.txt

cd ../../

# Generate gcov report

gcov main.c || { echo "gcov failed. Exiting..."; exit 1; }

Предназначен для собрание информации о покрытии кода.

**Скрипт comparator.sh**

#!/bin/bash

# Check number of parameters

if [[ $# -lt 2 || $# -gt 3 ]]; then

echo "Wrong parameter count"

exit 2

fi

# Assign filenames to variables

file1=$1

file2=$2

# Define regular expression to match floating point numbers

regex="^[+-]?[0-9]\*\.[0-9]+$"

# Initialize end of file flags

eof1=0

eof2=0

# Loop through both files simultaneously

for (( i=0; eof1!=1 || eof2!=1; i++ )); do

# Read one byte at a time from file1 until a floating point number is found

num1=""

while read -n 1 -r -u 3 byte; do

if [[ "$byte" =~ $regex ]]; then

num1="$num1$byte"

elif [[ "$num1" != "" ]]; then

break

fi

done

# Read one byte at a time from file2 until a floating point number is found

num2=""

while read -n 1 -r -u 4 byte; do

if [[ "$byte" =~ $regex ]]; then

num2="$num2$byte"

elif [[ "$num2" != "" ]]; then

break

fi

done

# Compare the two numbers

if [[ "$num1" =~ $regex && "$num2" =~ $regex ]]; then

if [[ "$3" = "-v" ]]; then

comparison="Comparing $num1 & $num2"

echo "$comparison"

fi

if [[ "$num1" != "$num2" ]]; then

if [[ "$3" = "-v" ]]; then

echo "file: $file1 != file: $file2"

fi

exit 1

fi

fi

# Set end of file flags

if ! read -n 1 -r -u 3 && [[ ${PIPESTATUS[0]} -ne 0 ]]; then

eof1=1

fi

if ! read -n 1 -r -u 4 && [[ ${PIPESTATUS[0]} -ne 0 ]]; then

eof2=1

fi

done 3< "$file1" 4< "$file2"

# Check if files are equal

if [[ "$file1" != "$file2" ]]; then

if [[ "$3" = "-v" ]]; then

echo "file: $file1 != file: $file2"

fi

exit 1

fi

exit 0

Предназначен для выбора компаратора, сравнивающий по определенному признаку. Это может быть компаратор, сравнивающий только целые числа, только вещественные числа

**Скрипт func\_tests.sh**

#!/bin/bash

failed=0

# Test for positive cases

for in\_file in ../data/pos\_\*\_in.txt; do

test\_num=${in\_file##\*pos\_} # Remove all characters before the last occurrence of "pos\_"

test\_num=${test\_num%%\_in.txt} # Remove all characters after "\_in.txt"

if [ -z "$test\_num" ]; then

echo "There are no positive tests!"

break

fi

out\_file="../data/pos\_${test\_num}\_out.txt"

if ./pos\_case.sh "$in\_file" "$out\_file" ; then

echo "test\_pos\_$test\_num: PASSED"

else

((failed++))

echo "test\_pos\_$test\_num: FAILED"

fi

done

# Test for negative cases

for in\_file in ../data/neg\_\*\_in.txt; do

test\_num=${in\_file##\*neg\_} # Remove all characters before the last occurrence of "neg\_"

test\_num=${test\_num%%\_in.txt} # Remove all characters after "\_in.txt"

if [ -z "$test\_num" ]; then

echo "There are no negative tests!"

break

fi

out\_file="../data/neg\_${test\_num}\_out.txt"

if ./neg\_case.sh "$in\_file" "$out\_file" ; then

echo "test\_neg\_$test\_num: PASSED"

else

((failed++))

echo "test\_neg\_$test\_num: FAILED"

fi

done

echo "Total failed tests: $failed"

Предназначен для тестирование позитивных и негативных входных данных.

**Скрипт neg\_case.sh**

#!/bin/bash

input\_file="$1"

if test -f "$1"; then

../../app.exe < "$1" > /dev/null 2>&1

if [ $? -eq 0 ]; then

exit 1

else

exit 0

fi

else

echo "Error: file '$1' not found" >&2

exit 1

fi

exit "$input\_file"

Предназначен для тестирование негативных входных данных.

**Скрипт pos\_case.sh**

#!/bin/bash

if [[ $# -lt 2 || $# -gt 3 ]]; then

echo "comparator: Wrong parameter count"

exit 2

fi

file1=$1

file2=$2

regex="^[+-]?[0-9]\*\.[0-9]+$"

tempfile=$(mktemp)

../../app.exe < "$file1" > "$tempfile"

if ./comparator.sh "$tempfile" "$file2"; then

rm "$tempfile" && exit 0 || rm "$tempfile" && exit 1

fi

rm "$tempfile"

Предназначен для тестирование позитивных входных данных.

**ЗАКЛЮЧЕНИЕ**

Скрипт командной оболочки для:  
1. Реализация отладочной и релизной сборок.  
2. Реализация отладочной сборки с санитайзерами. 3. Реализация очистки побочных файлов.

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

5. Реализация компаратор для сравнения содержимого двух текстовых файлов, располагающегося после первого вхождения подстроки «Result:␣».

6. Реализация pos\_case.sh для проверки позитивного тестового случая по определённым правилам.

7. Реализация скрипт neg\_case.sh для проверки негативного тестового случая по определённым правилам.

8. Обеспечения автоматизацию функционального тестирования. разработан.

**разработан**

Задание выполнено!

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. материал по линкус из первого семестра
2. видео лекции и презентации