РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Фундаментальная Информатика и Информационные технологии

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ № 13

дисциплина: операционные системы

Узор-Ежикеме Чинечелум А.

НФИбд-03-21

1032215263

Цель работы

Приобрести простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования
С калькулятора с простейшими функциями.

Последовательность выполнения работы

1. В домашнем каталоге создайте подкаталог ~/work/os/lab_prog.

2. Создайте в нём файлы: calculate.h, calculate.c, main.c.

3. Выполните компиляцию программы посредством gcc:

4. При необходимости исправьте синтаксические ошибки.

5. Создайте Makefile со следующим содержанием:

6. С помощью gdb выполните отладку программы calcul (перед использованием gdb исправьте Makefile):

7. С помощью утилиты splint попробуйте проанализировать коды файлов calculate.c и main.c.

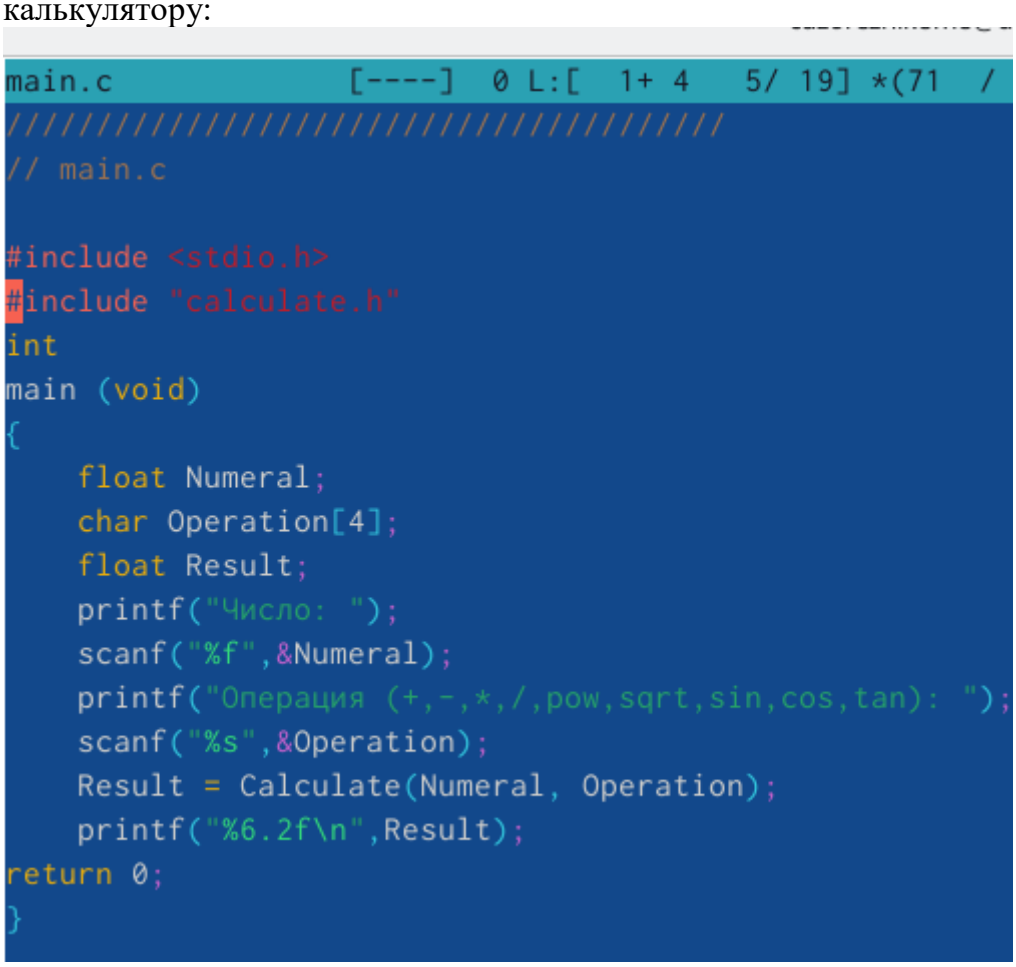
- Создал файлы: calculate.h, calculate.c, main.c. Я написал эскизы, которые будут представлять собой примитивный калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, извлекать квадратный корень, вычислять sin, cos, tan. При запуске он запросит первый номер, операцию, второй номер. После этого программа выведет результат и остановится. Файл интерфейса calculate.h, описывающий формат вызова функции калькулятора:

```
calculate.c        [----] 10 L:[  1+ 8   9/ 60] *(13
#include<stdio.h>
#include<math.h>
#include<string.h>
#include"calculate.h"

float
Calculate(float Numeral, char Operation[4])
{
    float SecondNumeral;
    if(strncmp(Operation, "+", 1) == 0)
<------>{
<------>    printf("Второе слагаемое: ");
<------>    scanf("%f",&SecondNumeral);
<------>    return(Numeral + SecondNumeral);
<------>}
    else if(strncmp(Operation,"-", 1) == 0)
<------>{
<------>    printf("Вычитаемое: ");
<------>    scanf("%f",&SecondNumeral);
<------>    return(Numeral - SecondNumeral);
<------>}
    else if(strncmp(Operation, "*", 1) == 0)
<------>{
<------>    printf("Множитель: ");
<------>    scanf("%f",&SecondNumeral);
<------>    return(Numeral * SecondNumeral);
<------>}
<------>else if(strncmp(Operation,"/", 1) == 0)
<------>{
<------>    printf("Делитель: ");
<------>    scanf("%f",&SecondNumeral);
<------>    if(SecondNumeral == 0)
<------>    {
<------><------>printf("Ошибка: деление на ноль! ");
<------><------>return(HUGE_VAL);
<------>    }
<------>    else
<------><------>return(Numeral / SecondNumeral);
<------>}
    else if(strncmp(Operation, "pow", 3) == 0)
<------>{
<------>    printf("Степень: ");
```

```
<------>else if(strncmp(Operation,"/", 1) == 0)
<------>{
<------>    printf("Делитель: ");
<------>    scanf("%f",&SecondNumeral);
<------>    if(SecondNumeral == 0)
<------>    {
<------><------>printf("Ошибка: деление на ноль! ");
<------><------>return(HUGE_VAL);
<------>    }
<------>    else
<------><------>return(Numeral / SecondNumeral);
<------>}
    else if(strncmp(Operation, "pow", 3) == 0)
<------>{
<------>    printf("Степень: ");
<------>    scanf("%f",&SecondNumeral);
<------>    return(pow(Numeral, SecondNumeral));
<------>}
    else if(strncmp(Operation, "sqrt", 4) == 0)
<------>return(sqrt(Numeral));
    else if(strncmp(Operation, "sin", 3) == 0)
<------>return(sin(Numeral));
    else if(strncmp(Operation, "cos", 3) == 0)
<------>return(cos(Numeral));
    else if(strncmp(Operation,"tan", 3) == 0)
<------>return(tan(Numeral));
    else
<------>{
<------>    printf("Неправильно введено действие ");
<------>    return(HUGE_VAL);
<------>}
}
```

- Основной файл main.c, реализующий интерфейс пользователя к калькулятору:



- Скомпилировал программу с использованием gcc:

```
cuzorezhikeme@dk8n59 ~ $ cd work/os
cuzorezhikeme@dk8n59 ~/work/os $ mcedit calculate.c

cuzorezhikeme@dk8n59 ~/work/os $ mcedit main.c

cuzorezhikeme@dk8n59 ~/work/os $ gcc -c main.c
main.c: В функции «main»:
main.c:15:13: предупреждение: формат «%s» ожидает аргумент типа «char *», но аргумент 2 имеет тип «char (*)[4]» [-Wformat=]
   15 |     scanf("%s",&Operation);
      |            ~^  ~~~~~~~~~~
      |             |  |
      |             |  char (*)[4]
      |             char *
cuzorezhikeme@dk8n59 ~/work/os $ gcc calculate.o main.o -o calcul -lm
cuzorezhikeme@dk8n59 ~/work/os $
```

- Создал Makefile:

```
makefile          [----]  0 L:[  1+20  21/ 21] *
#
# Makefile
#

CC = gcc
CFLAGS =
LIBS = -lm

calcul: calculate.o main.o
<------>gcc calculate.o main.o  -o calcul $(LIBS)
<------>
calculate.o: calculate.c calculate.h
<------>gcc -c calculate.c $(CFLAGS)

main.o: main.c calculate.h
<------>gcc -c main.c $(CFLAGS)
<------>
clean:
<------>-rm calcul *.o *~
<------>
# End Makefile
```

- Использовал gdb для отладки программы calcul.

```
cuzorezhikeme@dk8n59 ~/work/os $ gdb ./calcul
GNU gdb (Gentoo 10.2 vanilla) 10.2
Copyright (C) 2021 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-pc-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://bugs.gentoo.org/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./calcul...
(No debugging symbols found in ./calcul)
(gdb) run
Starting program: /afs/.dk.sci.pfu.edu.ru/home/c/u/cuzorezhikeme/work/os/calcul
Число: list
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): Неправильно введено действие    inf
[Inferior 1 (process 4843) exited normally]
(gdb) list 12,15
No symbol table is loaded.  Use the "file" command.
(gdb) run
Starting program: /afs/.dk.sci.pfu.edu.ru/home/c/u/cuzorezhikeme/work/os/calcul
Число: list 12,15
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): Неправильно введено действие    inf
[Inferior 1 (process 5001) exited normally]
(gdb) list calculate.c:20,27
No symbol table is loaded.  Use the "file" command.
(gdb) info breakpoints
No breakpoints or watchpoints.
(gdb) run
Starting program: /afs/.dk.sci.pfu.edu.ru/home/c/u/cuzorezhikeme/work/os/calcul
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Вычитаемое: backtrace
  5.00
[Inferior 1 (process 5248) exited normally]
(gdb) print Numeral
No symbol table is loaded.  Use the "file" command.
```

Контрольные вопросы

1. Как получить информацию о возможностях программ gcc, make, gdb и др.?
- с помощью команды help.

2. Что такое суффикс в контексте языка программирования? Приведите примеры использования.
- Суффикс - это условное обозначение добавления одного или нескольких символов к имени файла

3. Каково основное назначение компилятора языка C в UNIX?
- Компилятор берет рецепт (код) для новой программы (написанной на языке высокого уровня) и преобразует этот код в новый язык (машинный язык), который может быть понят самим компьютером.

4. Для чего предназначена утилита make?
- Инструмент, который автоматически определяет, какие исходные файлы программы необходимо перекомпилировать и/или связать.

5. Назовите и дайте основную характеристику основным командам отладчика gdb.
- b main - Устанавливает точку останова в начале программы
b - Устанавливает точку останова в текущей строке
b N - Ставит точку останова на номер строки
b +N - Устанавливает точку останова на N строк ниже текущей строки.
bfn - Ставит точку останова в начале функции "fn"
d N - Удалить номер точки останова N
info break - список точек останова
r - запускайте программу до тех пор, пока не будет достигнута точка останова или ошибка
c - Продолжает выполнение программы до следующей точки останова или ошибки
f - Выполнять до тех пор, пока текущая функция не будет завершена
s - запускает следующую строку программы
s N - Запускает следующие N строк программы
n - Лайков, но он не переходит в функции
u N - Выполняйте до тех пор, пока не получите N строк перед текущей строкой
p var - Выводит текущее значение переменной "var"
bt - Выводит трассировку стека
u - Поднимается на уровень выше в стеке
d - Опускается на уровень ниже в стеке
q - Завершает работу gdb

6. Прокомментируйте реакцию компилятора на синтаксические ошибки в программе при его первом запуске.

- В принципе, если есть ошибка компиляции, то вы не можете ее запустить. Синтаксическая ошибка означает, что компилятор / интерпретатор говорит: "Я не знаю, что вы имеете в виду под этим", поэтому он не может генерировать какой-либо код или выполнять какие-либо команды, пока вы не исправите это.

7. Назовите основные средства, повышающие понимание исходного кода программы.
- Самые основные инструменты - это редактор исходного кода и компилятор или интерпретатор, которые используются повсеместно и постоянно.

8. Каковы основные задачи, решаемые программой splint?
- Split особенно хорош при проверке типов назначений переменных и функций, эффективности, неиспользуемых переменных и идентификаторов функций, недоступного кода и возможных утечек памяти.