

Лабораторной работы №13

Дисциплина “Операционные системы”

Обрезкова А.В.

6 мая 2023

Российский университет дружбы народов, Москва, Россия

ФФМиЕН

Информация

- Обрезкова Анастасия Владимировна
- студентка направления “Математика и механика”
- Российский университет дружбы народов
- 1132226505@pfur.ru

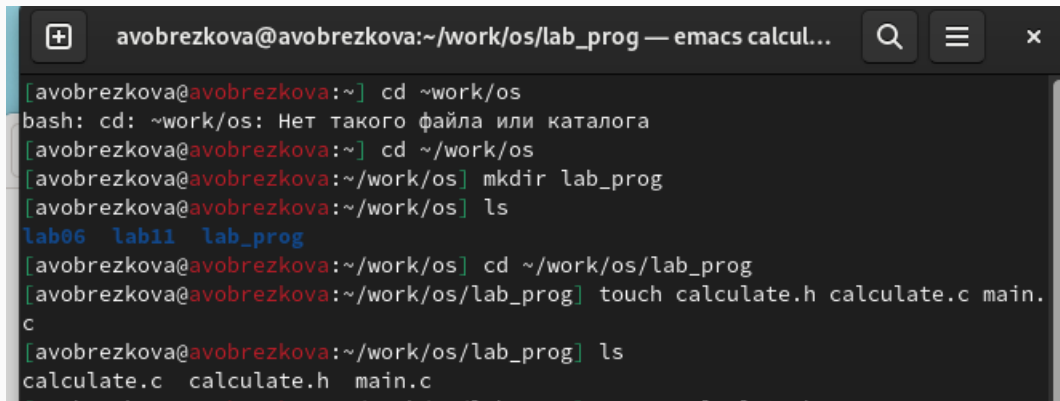
Вводная часть

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.

#Основная часть

Выполнение лабораторной работы

- В домашнем каталоге создаём подкаталог “~/work/os/lab_prog” с помощью команды “mkdir”.



```
avobrezkova@avobrezkova:~/work/os/lab_prog — emacs calcul...
[avobrezkova@avobrezkova:~] cd ~/work/os
bash: cd: ~/work/os: Нет такого файла или каталога
[avobrezkova@avobrezkova:~] cd ~/work/os
[avobrezkova@avobrezkova:~/work/os] mkdir lab_prog
[avobrezkova@avobrezkova:~/work/os] ls
lab06  lab11  lab_prog
[avobrezkova@avobrezkova:~/work/os] cd ~/work/os/lab_prog
[avobrezkova@avobrezkova:~/work/os/lab_prog] touch calculate.h calculate.c main.c
[avobrezkova@avobrezkova:~/work/os/lab_prog] ls
calculate.c  calculate.h  main.c
```



```
////////////////////////////////////  
// calculate.c
```

```
#include <stdio.h>  
#include <math.h>  
#include <string.h>  
#include "calculate.h"
```

```
float
```

```
Calculate(float Numeral, char Operation[4])
```

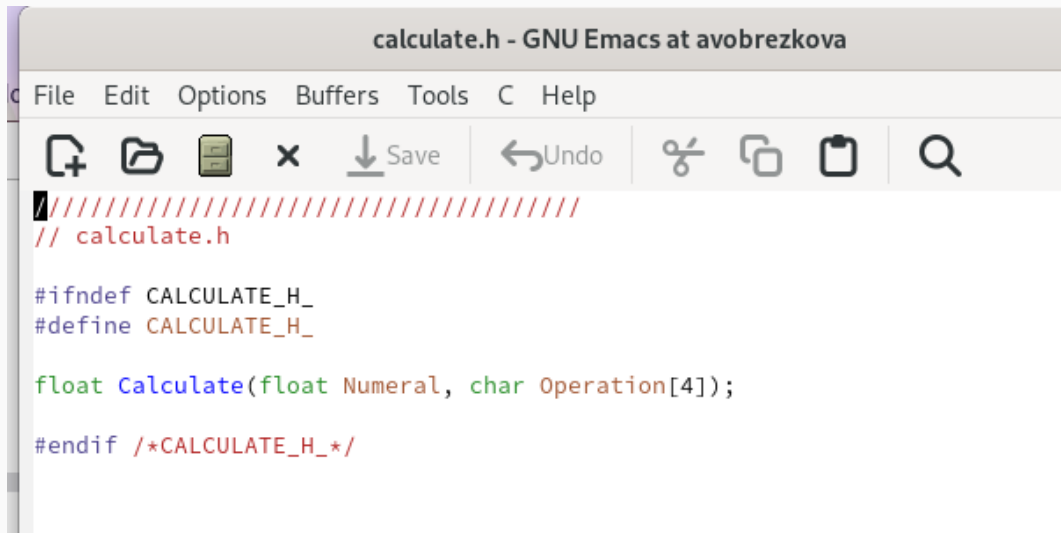
```
{  
    float SecondNumeral;  
    if(strncmp(Operation, "+", 1) == 0)  
    {  
        printf("Второе слагаемое: ");  
        scanf("%f",&SecondNumeral);  
        return(Numeral + SecondNumeral);  
    }  
    else if(strncmp(Operation, "-", 1) == 0)  
    {  
        printf("Вычитаемое: ");  
        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: ");
    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);
}
}

```


Интерфейсный файл calculate.h, описывающий формат вызова функции калькулятора.



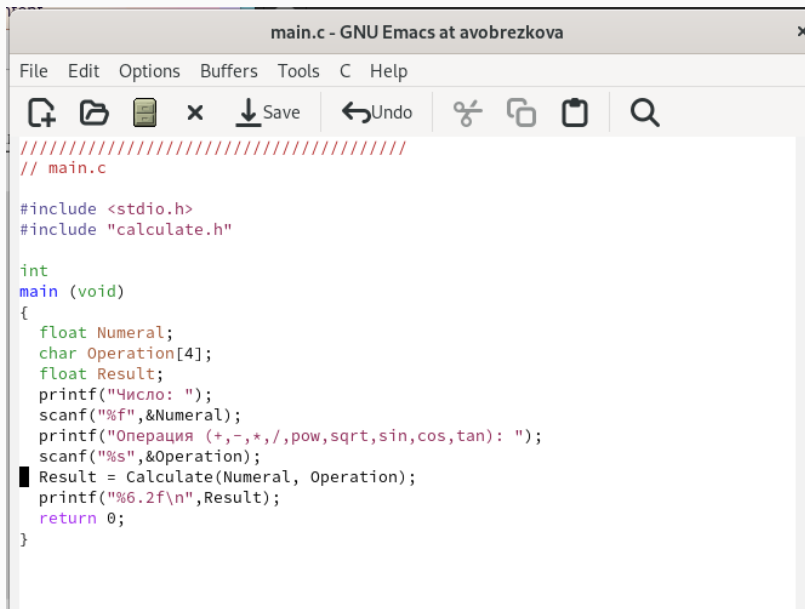
```
calculate.h - GNU Emacs at avobrezkova
File Edit Options Buffers Tools C Help
[Icons: Open, Save, Undo, Redo, Search, etc.]
////////////////////////////////////
// calculate.h

#ifndef CALCULATE_H_
#define CALCULATE_H_

float Calculate(float Numeral, char Operation[4]);

#endif /*CALCULATE_H_*/
```

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



```
main.c - GNU Emacs at avobrezkova
File Edit Options Buffers Tools C Help
[Icons: New, Open, Save, Close, Save All, Undo, Cut, Copy, Paste, Find]

////////////////////
// main.c

#include <stdio.h>
#include "calculate.h"

int
main (void)
{
    float Numeral;
    char Operation[4];
    float Result;
    printf("Число: ");
    scanf("%f",&Numeral);
    printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
    scanf("%s",&Operation);
    Result = Calculate(Numeral, Operation);
    printf("%6.2f\n",Result);
    return 0;
}
```

- Далее выполним компиляцию программы посредством gcc.

```
[avobrezkova@avobrezkova:~/work/os/lab_prog] gcc -c calculate.c  
[avobrezkova@avobrezkova:~/work/os/lab_prog] gcc -c main.c  
[avobrezkova@avobrezkova:~/work/os/lab_prog] gcc calculate.o main.o -o calcul -lm  
[avobrezkova@avobrezkova:~/work/os/lab_prog]
```

- Ошибок не выявлено

makefile - GNU Emacs at avobrezkova

File Edit Options Buffers Tools Makefile Help

Save Undo

```
#
# 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
```



```
#
# Makefile
#

CC = gcc
CFLAGS = -g
LIBS = -lm

calcul: calculate.o main.o
    $(CC) calculate.o main.o -o calcul $(LIBS)

calculate.o: calculate.c calculate.h
    $(CC) -c calculate.c $(CFLAGS)

main.o: main.c calculate.h
    $(CC) -c main.c $(CFLAGS)

clean:
    -rm calcul *.o *~

# End Makefile
```

После выполняем компиляцию файлов.

```
[avobrezkova@avobrezkova:~/work/os/lab_prog] make clean
rm calcul *.o *~
[avobrezkova@avobrezkova:~/work/os/lab_prog] make calculate.o
gcc -c calculate.c -g
[avobrezkova@avobrezkova:~/work/os/lab_prog] make main.o
gcc -c main.c -g
[avobrezkova@avobrezkova:~/work/os/lab_prog] make calcul
gcc calculate.o main.o -o calcul -lm
[avobrezkova@avobrezkova:~/work/os/lab_prog]
```

После этого выполняем gdb отладку программы calcul. Запускаем GDB и загружаем в него программу для отладки, используя команду “gdb ./calcul”

```
[avobrezkova@avobrezkova:~/work/os/lab_prog] gdb ./calcul
GNU gdb (GDB) Fedora 12.1-2.fc36
Copyright (C) 2022 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-redhat-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
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...
(gdb)
```

Далее вводим команду “run” для запуска программы внутри отладчика.

```
(gdb) run
Starting program: /home/avobrezkova/work/os/lab_prog/calcul

This GDB supports auto-downloading debuginfo from the following URLs:
https://debuginfod.fedoraproject.org/
Enable debuginfod for this session? (y or [n]) y
Debuginfod has been enabled.
To make this setting permanent, add 'set debuginfod enabled on' to .gdbinit.
Downloading 7.42 MB separate debug info for /lib64/libc.so.6
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 10
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 14
24.00
[Inferior 1 (process 4575) exited normally]
(gdb)
```


Для постраничного просмотра исходного кода используем команду “list”.

```
(gdb) list
1      //////////////////////////////////////////
2      // main.c
3
4      #include <stdio.h>
5      #include "calculate.h"
6
7      int
8      main (void)
9      {
10         float Numeral;
(gdb)
```

Для просмотра строк с 12 по 15 основного файла используем команду “list 12,15”.

```
(gdb) list 12,15
12      float Result;
13      printf("Число: ");
14      scanf("%f",&Numeral);
15      printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
(gdb) 
```

Для просмотра определённых строк не основного файла используем команду “list calculate.c:20,29”.

```
(gdb) list calculate.c:20,29
20      {
21          printf("Вычитаемое: ");
22          scanf("%f",&SecondNumeral);
23          return(Numeral - SecondNumeral);
24      }
25      else if(strncmp(Operation, "*", 1) == 0)
26      {
27          printf("Множитель: ");
28          scanf("%f",&SecondNumeral);
29          return(Numeral * SecondNumeral);
(gdb)
```

Для установки точки в файле “calculate.c” на строке 21 используем команды “list calculate.c:20,27” и “break 21”.

```
(gdb) break 21  
Breakpoint 1 at 0x40120f: file calculate.c, line 21.  
(gdb)
```

Чтобы вывести информацию об имеющихся точках останова используем команду “info breakpoint”

```
(gdb) info breakpoints
Num      Type           Disp Enb Address            What
1        breakpoint    keep y   0x00000000000040120f in Calculate
                                                at calculate.c:21
(gdb) 
```

Запустим программу внутри отладчика и убедимся, что программа остановилась в момент прохождения точки останова.

```
(gdb) run
Starting program: /home/avobrezkova/work/os/lab_prog/calcul
BFD: warning: /home/avobrezkova/.cache/debuginfod_client/8257ee907646e9b057197533d1e4ac8ede7a9c5c/debuginfo has a section extending past end of file
Error while reading shared library symbols for /lib64/libc.so.6:
`/home/avobrezkova/.cache/debuginfod_client/8257ee907646e9b057197533d1e4ac8ede7a9c5c/debuginfo': can't read symbols: формат файла не распознан.
Число: 12
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -

Breakpoint 1, Calculate (Numeral=12, Operation=0x7fffffffdf34 "-") at calculate.c:21
21         printf("Вычитаемое: ");
(gdb)
```

Посмотрим, чему на этом этапе равно значение переменной Numeral, с помощью команды “print Numeral” и сравним его с результатом вывода на экран после использования команды “display Numeral”. Значения совпадают.

```
(gdb) print Numeral
$1 = 12
(gdb) display Numeral
1: Numeral = 12
(gdb) 
```

Уберём точки останова с помощью команды “d breakpoints”

```
(gdb) info breakpoints
Num      Type           Disp Enb Address                  What
1        breakpoint    keep y   0x00000000000040120f in Calculate
                                at calculate.c:21

    breakpoint already hit 1 time
(gdb) delete 1
(gdb)
```


- С помощью утилиты splint проанализировала коды файлов calculate.c и main.c.

```
avobrezkova@avobrezkova:~/work/os/lab_prog
[avobrezkova@avobrezkova:~/work/os/lab_prog] splint calculate.c
Splint 3.1.2 --- 22 Jan 2022

calculate.h:7:37: Function parameter Operation declared as manifest array (size
                    constant is meaningless)
    A formal parameter is declared as an array with size. The size of the array
    is ignored in this context, since the array formal parameter is treated as a
    pointer. (Use -fixedformalarray to inhibit warning)
calculate.c:10:31: Function parameter Operation declared as manifest array
                    (size constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:16:7: Return value (type int) ignored: scanf("%f", &Sec...
    Result returned by function call is not used. If this is intended, can cast
    result to (void) to eliminate message. (Use -retvalint to inhibit warning)
calculate.c:22:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:28:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:34:7: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:35:10: Dangerous equality comparison involving float types:
                    SecondNumeral == 0
    Two real (float, double, or long double) values are compared directly using
    == or != primitive. This may produce unexpected results since floating point
    representations are inexact. Instead, compare the difference to FLT_EPSILON
    or DBL_EPSILON. (Use -realcompare to inhibit warning)
calculate.c:38:10: Return value type double does not match declared type float:
```

```
[avobrezkova@avobrezkova:~/work/os/lab_prog] splint main.c
```

```
Splint 3.1.2 --- 22 Jan 2022
```

```
calculate.h:7:37: Function parameter Operation declared as manifest array (size  
constant is meaningless)
```

```
A formal parameter is declared as an array with size. The size of the array  
is ignored in this context, since the array formal parameter is treated as a  
pointer. (Use -fixedformalarray to inhibit warning)
```

```
main.c: (in function main)
```

```
main.c:14:3: Return value (type int) ignored: scanf("%f", &Num...
```

```
Result returned by function call is not used. If this is intended, can cast  
result to (void) to eliminate message. (Use -retvalint to inhibit warning)
```

```
main.c:16:14: Format argument 1 to scanf (%s) expects char * gets char [4] *:  
&Operation
```

```
Type of parameter is not consistent with corresponding code in format string.  
(Use -formattype to inhibit warning)
```

```
main.c:16:11: Corresponding format code
```

```
main.c:16:3: Return value (type int) ignored: scanf("%s", &Ope...
```

```
Finished checking --- 4 code warnings
```

```
[avobrezkova@avobrezkova:~/work/os/lab_prog]
```

Заключение

В ходе выполнения данной лабораторной работы я приобрела простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями.