

Отчёт по лабораторной работе №13

Дисциплина “Операционные системы”

Батова Ирина Сергеевна, НММбд-01-22

02 мая 2023

Российский университет дружбы народов, Москва, Россия

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

- Вводим скрипт для реализации функций калькулятора в файле calculate.c

Скрипт в calculate.c

```
1 //////////////////////////////////////
2 // calculate.c
3
4 #include <stdio.h>
5 #include <math.h>
6 #include <string.h>
7 #include "calculate.h"
8
9 float
10 Calculate(float Numeral, char Operation[4])
11 {
12     float SecondNumeral;
13     if(strncmp(Operation, "+", 1) == 0)
14     {
15         printf("Второе слагаемое: ");
16         scanf("%f",&SecondNumeral);
17         return(Numeral + SecondNumeral);
18     }
19     else if(strncmp(Operation, "-", 1) == 0)
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);
30     }
31     else if(strncmp(Operation, "/", 1) == 0)
32     {
33         printf("Делитель: ");
34         scanf("%f",&SecondNumeral);
35         if(SecondNumeral == 0)
36     {
```

- Вводим скрипт в интерфейсный файл calculate.h, описывающий формат вызова функции-калькулятора

```
1 //////////////////////////////////////
2 // calculate.h
3
4 #ifndef CALCULATE_H_
5 #define CALCULATE_H_
6
7 float Calculate(float Numeral, char Operation[4]);
8
9 #endif /*CALCULATE_H_*/
10
```

- Вводим скрипт в основной файл main.c, реализующий интерфейс пользователя к калькулятору

```
1 ////////////////////////////////////////////////////
2 // main.c
3
4 #include <stdio.h>
5 #include "calculate.h"
6
7 int
8 main (void)
9 {
10 float Numeral;
11 char Operation[4];
12 float Result;
13 printf("Число: ");
14 scanf("%f",&Numeral);
15 printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
16 scanf("%s",&Operation);
```



```
[isbatova@fedora lab_prog]$ gcc -c calculate.c  
[isbatova@fedora lab_prog]$ gcc -c main.c  
[isbatova@fedora lab_prog]$ gcc calculate.o main.o -o calcul -lm  
[isbatova@fedora lab_prog]$
```

Makefile

- Файл используется для автоматической компиляции main.c, calculate.c и создание из них исполняемого файла calcul. Помимо этого, в файле также есть функция 'clean', используемая для удаления всех файлов. В начале скрипта также вводятся переменные: CC - команда для компиляции gcc, CFLAGS - опции к команде компиляции, LIBS - опции при создании исполняемого файла.

```
1 #
2 # Makefile
3 #
4
5 CC = gcc
6 CFLAGS =
7 LIBS = -lm
8
9 calcul: calculate.o main.o
10     gcc calculate.o main.o -o calcul $(LIBS)
11
12 calculate.o: calculate.c calculate.h
13     gcc -c calculate.c $(CFLAGS)
```

```
[isbatova@fedora lab_prog]$ make clean
rm calcul *.o *~
[isbatova@fedora lab_prog]$ make calculate.o
gcc -c calculate.c
[1]+  Завершён          emacs
[isbatova@fedora lab_prog]$ make main.o
gcc -c main.c
[isbatova@fedora lab_prog]$ make calcul
gcc calculate.o main.o -o calcul -lm
```

```
[isbatova@fedora lab_prog]$ gdb ./calcul
GNU gdb (GDB) Fedora 12.1-4.fc37
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...
```

Запуск программы внутри отладчика GDB

```
(gdb) run
Starting program: /home/isbatova/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]) n
Debuginfod has been disabled.
To make this setting permanent, add 'set debuginfod enabled off' to .gdbinit.
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 3
      8.00
[Inferior 1 (process 4937) exited normally]
Missing separate debuginfos, use: dnf debuginfo-install glibc-2.36-7.fc37.x86_
(gdb)
```

```
(gdb) list calculate.c:20,27
20      {
21      printf("Вычитаемое: ");
22      scanf("%f",&SecondNumeral);
23      return(Numeral - SecondNumeral);
24      }
25      else if(strncmp(Operation, "*", 1) == 0)
26      {
27      printf("Множитель: ");
(gdb) break 21
Breakpoint 1 at 0x40120f: file calculate.c, line 21.
```

Запуск программы с точками останова

```
(gdb) run
Starting program: /home/isbatova/work/os/lab_prog/calcul
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -

Breakpoint 1, Calculate (Numeral=5, Operation=0x7fffffffdf04 "-") at calculate.c:21
21      printf("Вычитаемое: ");
(gdb) backtrace
#0  Calculate (Numeral=5, Operation=0x7fffffffdf04 "-") at calculate.c:21
#1  0x00000000004014eb in main () at main.c:17
```

```
(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
```


Вывод

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