

# Операционные системы

Лабораторная работа № 13. Средства, применяемые при разработке программного обеспечения в ОС типа UNIX/Linux.

---

Абдеррахим Мугари.

7 мая 2023

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Абдеррахим Мугари
- Студент
- Российский университет дружбы народов
- 1032215692@pfur.ru
- <https://github.com/iragoum>

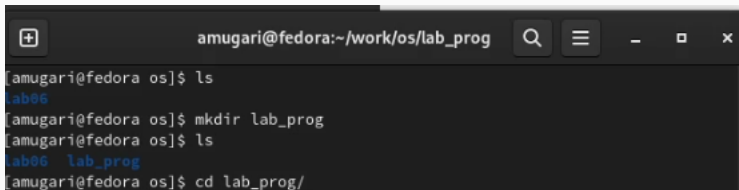


- Приобрести простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования C калькулятора с простейшими функциями.

- Терминал Unix.
- Файловая система.
- Текстовой редактор emacs .
- vi редактор



- Сначала в домашнем каталоге я создал подкаталог `~/work/os/lab_prog`

A terminal window with a dark background. The title bar shows the user 'amugari@fedora' and the current directory '~/work/os/lab\_prog'. The terminal contains the following commands and output:

```
[amugari@fedora os]$ ls
lab06
[amugari@fedora os]$ mkdir lab_prog
[amugari@fedora os]$ ls
lab06  lab_prog
[amugari@fedora os]$ cd lab_prog/
```

Рис. 1: создание подкаталога

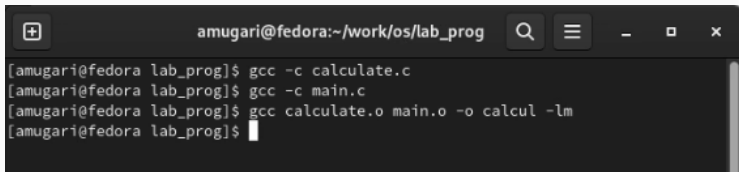
- затем я создал в нем три файла: `calculate.h`, `calculate.c`, `main.c` и поместил в них код

```
[amugari@fedora os]$ cd lab_prog/  
[amugari@fedora lab_prog]$ emacs calculate.h  
[amugari@fedora lab_prog]$ emacs calculate.c  
[amugari@fedora lab_prog]$ emacs main.c  
[amugari@fedora lab_prog]$
```

Рис. 2: создание файлов программы



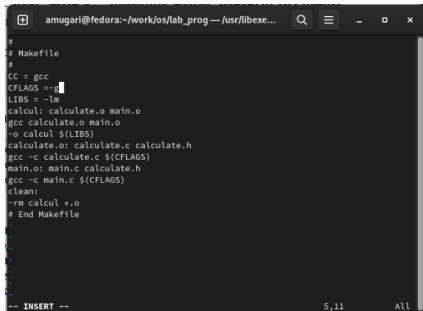
- затем я скомпилировал программу, используя gcc

A terminal window with a dark background. The title bar shows the user 'amugari' on a 'fedora' machine in the directory '~/work/os/lab\_prog'. The terminal contains four lines of text: the first three are compilation commands using 'gcc' to compile 'calculate.c', 'main.c', and then link them into an executable named 'calcul' with the '-lm' flag. The fourth line is a prompt with a cursor.

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

Рис. 3: компиляция файлов

- затем я создал Makefile и добавил его скрипт

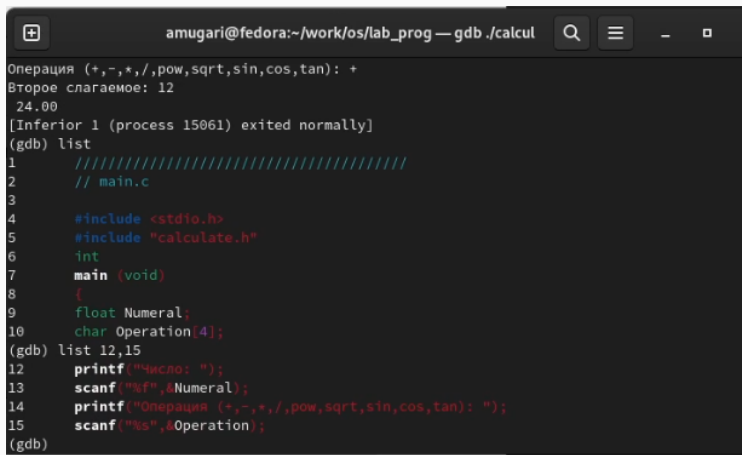


```
amugari@fedora:~/work/os/lab_prog — /usr/libexe...  
#  
# Makefile  
#  
CC = gcc  
CFLAGS = -g  
LIBS = -lm  
calcul: calculate.o main.o  
gcc calculate.o main.o  
-o calcul $(LIBS)  
calculate.o: calculate.c calculate.h  
gcc -c calculate.c $(CFLAGS)  
main.o: main.c calculate.h  
gcc -c main.c $(CFLAGS)  
clean:  
-rm calcul *.o  
# End Makefile  
  
-- INSERT -- 5,11 All
```

Рис. 4: создание Makefile

- после этого я запустил программу и отобразил заголовок `main.c`, а также отобразил строки между 12 и 15 из `main.c`

## запуск программы и управление кодом :



```
amugari@fedora:~/work/os/lab_prog — gdb ./calcul
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 12
24.00
[Inferior 1 (process 15061) exited normally]
(gdb) list
1      //////////////////////////////////////
2      // main.c
3
4      #include <stdio.h>
5      #include "calculate.h"
6      int
7      main (void)
8      {
9          float Numeral;
10         char Operation[4];
(gdb) list 12,15
12         printf("Число: ");
13         scanf("%f",&Numeral);
14         printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
15         scanf("%s",&Operation);
(gdb)
```

Рис. 5: запуск программы и управление кодом

## добавление точки останова :

- затем я проверил, что в файле calculate.c нет точек остановки, а затем добавил точку остановку в строку 21

```
(gdb) info breakpoints
No breakpoints or watchpoints.
(gdb) list calculate.c:20,27
20     scanf("%f",&SecondNumeral);
21     return(Numeral - SecondNumeral);
22 }
23 else if(strncmp(Operation, "*", 1) == 0)
24 {
25     printf("Множитель: ");
26     scanf("%f",&SecondNumeral);
27     return(Numeral * SecondNumeral);
(gdb) break 21
Breakpoint 6 at 0x401234: file calculate.c, line 21.
(gdb)
```

В МОМЕНТ ПРОХОЖДЕНИЯ ТОЧКИ ОСТАНОВА:

Рис. 6: добавление точки останова

## проверка функционирования точки останова :

- после этого я повторно запустил файл и убедился, что программа остановится на 21 строке, и это сработало

```
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/amugari/work/os/lab_prog/calcul
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 21
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Вычитаемое: 12

Breakpoint 6, Calculate (Numeral=21, Operation=0x7fffffffda34 "-") at calculate.c:21
21      return (Numeral - SecondNumeral);
(gdb) |
```

Рис. 7: проверка функционирования точки останова

## проверка значения операнда :

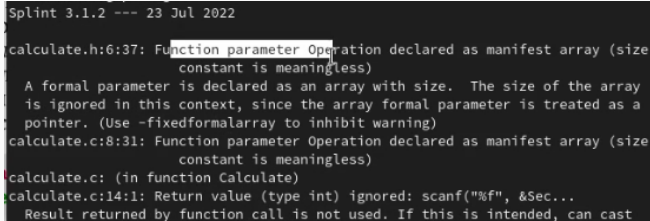
- после этого я проверил правильность значения первого операнда этой операции

```
(gdb) print Numeral
$1 = 21
(gdb) display Numeral
1: Numeral = 21
(gdb) info breakpoints
Num   Type             Disp Enb Address            What
6      breakpoint      keep y   0x0000000000401234 in Calculate at calculate.c:21
      breakpoint already hit 1 time
(gdb) delete 6
(gdb) info breakpoints
No breakpoints or watchpoints.
(gdb)
```

Рис. 8: проверка значения операнда

## роанализировать коды файлов calculate.c и main.c :

- наконец, используя утилиту splint, попытался проанализировать коды файлов calculate.c и main.c.



```
Splint 3.1.2 --- 23 Jul 2022
calculate.h:6:37: 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:8:31: Function parameter Operation declared as manifest array (size
constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:14:1: Return value (type int) ignored: scanf("%f", &Sec...
Result returned by function call is not used. If this is intended, can cast
```

Рис. 9: роанализировать коды файлов calculate.c



## роанализировать коды файлов calculate.c и main.c :

```
[amugari@fedora lab_prog]$ splint main.c
Splint 3.1.2 --- 23 Jul 2022

calculate.h:6:37: 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:13:1: 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:15:12: 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:15:9: Corresponding format code
main.c:15:1: Return value (type int) ignored: scanf("%s", &Ope...

Finished checking --- 4 code warnings
[amugari@fedora lab_prog]$
```

Рис. 10: роанализировать коды файлов calculate.c

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

Выводы, согласованные с целью  
работы:

---

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