

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

Дисциплина: Операционные системы

НВЕ МАНГЕ ХОСЕ ХЕРСОН МИКО

3 мая 2023

Группа НКАбд-03-22

Российский университет дружбы народов, Москва, Россия

Информация

- НВЕ МАНГЕ ХОСЕ ХЕРСОН МИКО
- Студент 1-го курса, группа НКАбд-03-22
- Математика и Компьютера науки
- Российский университет дружбы народов
- Ссылка на репозиторий НВЕ МАНГЕ `mmnve@dk8n`

Вводная часть

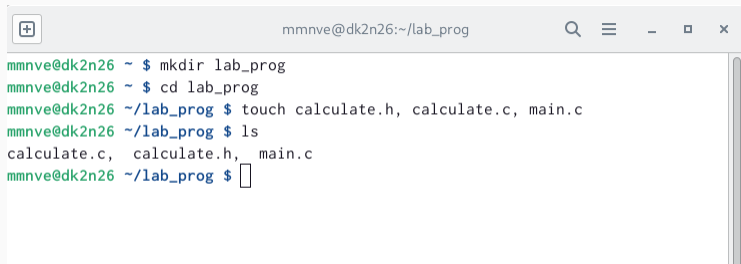
- Работа с программами позволит создать готовые скрипты для различных задач.

- Командный процессор ОС UNIX

- 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.

Выполнение заданий.

В домашнем каталоге создайте подкаталог `~/work/os/lab_prog`. Создайте в нём файлы: `calculate.h`, `calculate.c`, `main.c`

A terminal window with a title bar showing the user 'mmnve' on host 'dk2n26' in the directory '~/lab_prog'. The terminal contains a sequence of commands to create a directory and files. The output shows the files 'calculate.c', 'calculate.h', and 'main.c' being created and then listed.

```
mmnve@dk2n26 ~ $ mkdir lab_prog
mmnve@dk2n26 ~ $ cd lab_prog
mmnve@dk2n26 ~/lab_prog $ touch calculate.h, calculate.c, main.c
mmnve@dk2n26 ~/lab_prog $ ls
calculate.c, calculate.h, main.c
mmnve@dk2n26 ~/lab_prog $
```

Рис. 1: Создание файлов

```
#1  0x0000000000004014eb  III  III  III  
(gdb) print Numeral  
$1 = 5  
(gdb) display Numeral  
1: Numeral = 5  
(gdb) █
```

Рис. 2: calculate.c

```

(gdb) info breakpoints
Num      Type          Disp Enb Address                What
1        breakpoint    keep y  0x000000000040120f in Calculate
                                     at calculate.c:21
        breakpoint already hit 1 time
2        breakpoint    keep y  0x000000000040120f in Calculate
                                     at calculate.c:21
        breakpoint already hit 1 time
(gdb) delete 1
(gdb) info breakpoints
Num      Type          Disp Enb Address                What
2        breakpoint    keep y  0x000000000040120f in Calculate
                                     at calculate.c:21
        breakpoint already hit 1 time
(gdb) delete 2
(gdb) info breakpoints
No breakpoints or watchpoints.
(gdb) █

```

Рис. 3: main.c

```
Splint 3.1.2 --- 23 Jul 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:14:2: 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:16:13: 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:16:10: Corresponding format code
main.c:16:2: Return value (type int) ignored: scanf("%s", &Ope...

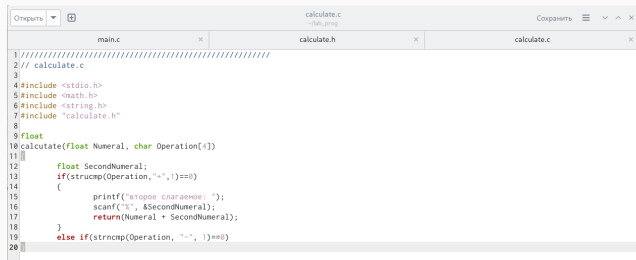
Finished checking --- 4 code warnings
```

Рис. 4: calculate.h

Выполните компиляцию программы посредством gcc. При необходимости исправьте синтаксические ошибки

Компиляция

Рис. 5: Компиляция

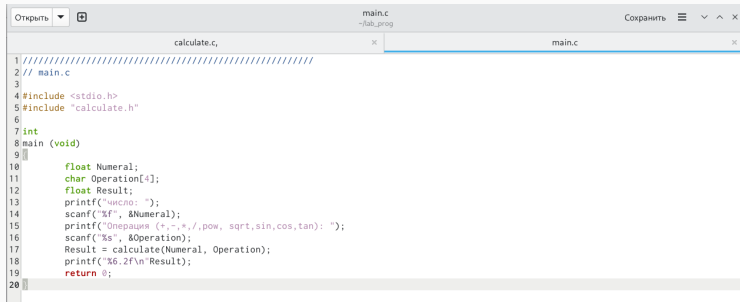


The screenshot shows a code editor with three tabs: main.c, calculate.h, and calculate.c. The calculate.c tab is active, displaying the following C code:

```
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(strcmp(Operation,"+")==0)
14     {
15         printf("второе слагаемое: ");
16         scanf("%f", &SecondNumeral);
17         return(Numeral + SecondNumeral);
18     }
19     else if(strcmp(Operation, "-")==0)
20     {
```

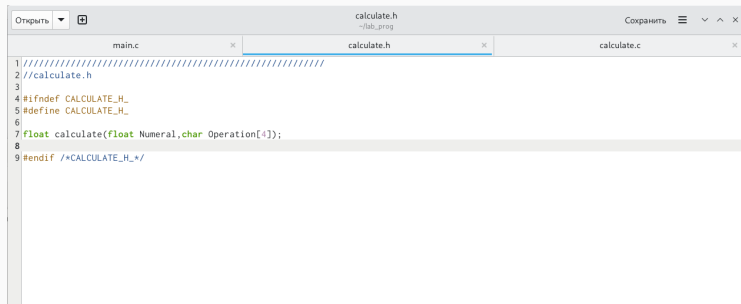
Рис. 6: Компиляция

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



```
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);
17     Result = calculate(Numeral, Operation);
18     printf("%.2f\n", Result);
19     return 0;
20 }
```

Рис. 7: Работа программы



```
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_*/
```

Рис. 8: list - несколько первых строк

```
[Enter for 1 (process 6615), exited normally]
(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)
```

Рис. 9: Команда info breakpoints

```
10  
(gdb) list 12,15  
12      float Result;  
13      printf("Число: ");  
14      scanf("%f",&Numeral);  
15      printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");  
(gdb) █
```

Рис. 10: Работа программы с прерыванием

```
(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) █
```

Рис. 11: backtrace

```
(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.
(gdb) █
```

Рис. 12: Значение переменной Numeral

```
(gdb) break 21
Breakpoint 1 at 0x40120f: file calculate.c, line 21.
(gdb) info breakpoints
Num      Type             Disp Enb Address                  What
1        breakpoint      keep y   0x000000000040120f in calculate
                                                at calculate.c:21
(gdb) █
```

Рис. 13: Та же переменная, но как бы после выполнения программы

```
[#####]5
Download failed: Таймер истек. Continuing without separate debug info for system-supplied DSO at 0x7ffff7fc6000.
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -

Breakpoint 1, Calculate (Numeral=5, Operation=0x7fffffffda54 "-")
    at calculate.c:21
21                                     printf("Вычитаемое: ");
(gdb)
```

Рис. 14: Удаление точек прерывания

Результаты

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