

Лабораторная работа №13

Белов Максим Сергеевич - НПИбд-01-21

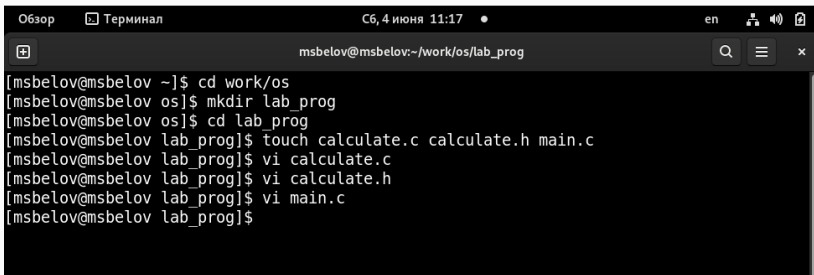
04.06.2022

Средства, применяемые при
разработке программного
обеспечения в ОС типа UNIX/Linux

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.

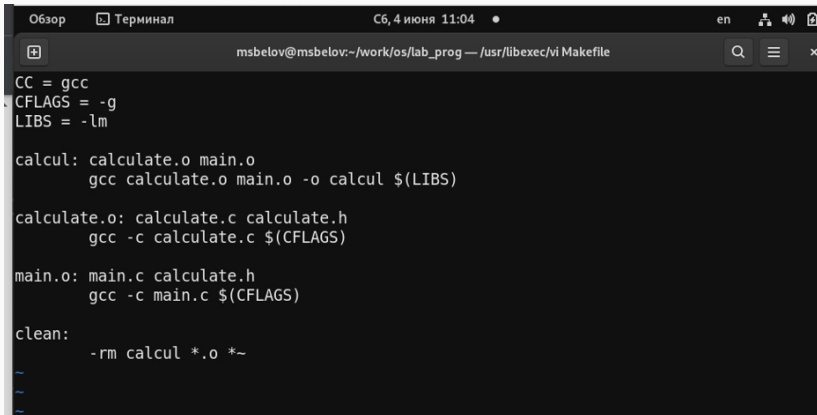
Выполнение заданий

1. Создадим каталог и в нем нужные нам файлы



```
Обзор Терминал C6, 4 июня 11:17 en [иконки]
msbelov@msbelov:~/work/os/lab_prog
[msbelov@msbelov ~]$ cd work/os
[msbelov@msbelov os]$ mkdir lab_prog
[msbelov@msbelov os]$ cd lab_prog
[msbelov@msbelov lab_prog]$ touch calculate.c calculate.h main.c
[msbelov@msbelov lab_prog]$ vi calculate.c
[msbelov@msbelov lab_prog]$ vi calculate.h
[msbelov@msbelov lab_prog]$ vi main.c
[msbelov@msbelov lab_prog]$
```

2. Создадим Makefile для автоматической компиляции программы



The image shows a terminal window with a dark background. The title bar at the top indicates the window is titled 'Обзор' (Overview) and 'Терминал' (Terminal), with the date and time 'Сб, 4 июня 11:04'. The terminal content shows a Makefile being edited in the vi editor. The Makefile defines variables for the compiler (CC), flags (CFLAGS), and libraries (LIBS), and includes rules for building the 'calcul' target, the 'calculate.o' object file, the 'main.o' object file, and a 'clean' target to remove object files and the executable.

```
msbelov@msbelov:~/work/os/lab_prog — /usr/libexec/vi 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 *~

~
~
~
```

3. Выполним отладку программы calcul. Запусти программу с помощью run

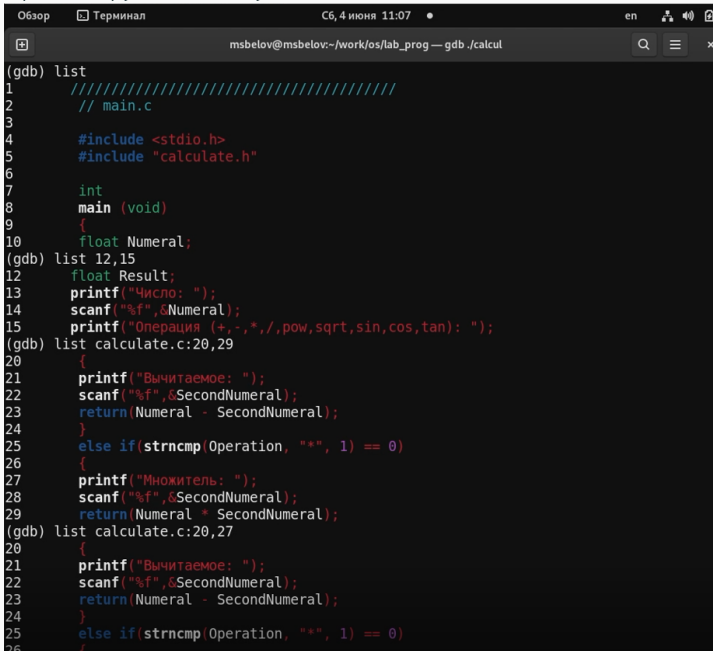
```
Обзор Терминал C6, 4 июня 11:05 en
msbelov@msbelov:~/work/os/lab_prog — gdb ./calcul

[msbelov@msbelov lab_prog]$ gdb ./calcul
GNU gdb (GDB) Fedora 11.2-2.fc35
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) run
Starting program: /home/msbelov/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.
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 2
Операция (+, -, *, /, pow, sqrt, sin, cos, tan): pow
Степень: 3
8.00
```

4. Протестируем команду list



The screenshot shows a terminal window with a dark background. The title bar at the top indicates the window is titled 'Терминал' (Terminal) and shows the date and time 'Сб, 4 июня 11:07'. The terminal content shows a GDB session where the 'list' command is used to display source code. The code is from a file named 'calculate.c' and includes standard headers and a 'main' function. The GDB prompts '(gdb) list' are followed by the code lines. The code includes comments in Russian and uses various mathematical functions like 'pow', 'sqrt', 'sin', 'cos', and 'tan'.

```
Обзор  Терминал  Сб, 4 июня 11:07  en  🔊  🗨  🗑
msbelov@msbelov:~/work/os/lab_prog — gdb ./calcul

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


5. Поставим точку останова на 21 строке в calculate.c

```
(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) run
Starting program: /home/msbelov/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=0x7fffffffddc4 "-") at calculate.c:21
21      printf("Вычитаемое: ");
(gdb) backtrace
#0 Calculate (Numeral=5, Operation=0x7fffffffddc4 "-") at calculate.c:21
#1 0x00000000004014eb in main () at main.c:17
(gdb) print Numeral
$1 = 5
(gdb) display Numeral
1: Numeral = 5
(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
(gdb) delete 1
(gdb) info breakpoints
No breakpoints or watchpoints.
```

6. splint calculate.c

```
Обзор Терминал C6, 4 июня 11:10 en
msbelov@msbelov:~/work/os/lab_prog

[msbelov@msbelov lab_prog]$ splint calculate.c
splint 3.1.2 --- 23 Jul 2021

calculate.h:7:38: 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:10:32: Function parameter Operation declared as manifest array
(size constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:16:1: 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:22:2: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:28:2: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:35:2: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:36:5: 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:39:8: Return value type double does not match declared type float:
(HUGE_VAL)
To allow all numeric types to match, use +relaxtypes.
calculate.c:47:2: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:48:8: Return value type double does not match declared type float:
(pow(Numeral, SecondNumeral))
calculate.c:51:8: Return value type double does not match declared type float:
(sqrt(Numeral))
calculate.c:53:8: Return value type double does not match declared type float:
(sin(Numeral))
calculate.c:55:8: Return value type double does not match declared type float:
(cos(Numeral))
```

Выводы

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