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

Мальсагов А.А.

Цель 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.

Выполнение лабораторной работы

1. Создал каталог work/os/lab_prog. СОздал в нем файлы **calculate.h, calculate.c, main.c, makefile**. Скопировал весь код из лабораторки. Выполнил компиляцию этих файлов.(рис. 1)

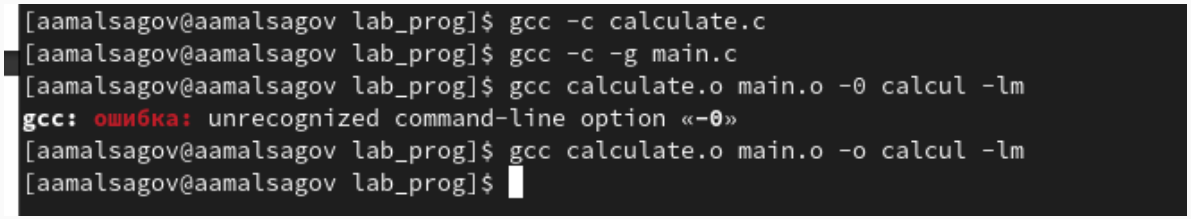

Figure 1: Компиляция

Выполнение лабораторной работы

2. Выполнил отладку.Прверил работу калькулятора (рис. 2)

```
[aamalsagov@aamalsagov lab_prog]$ gdb ./calcul
GNU gdb (GDB) Fedora 12.1-1.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) run
Starting program: /home/aamalsagov/work/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 0.01 MB separate debug info for system-supplied DSO at 0x7ffff7fc4000
Downloading 2.25 MB separate debug info for /lib64/libm.so.6
Downloading 7.39 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: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Вычитаемое: 2
  3.00
[Inferior 1 (process 65375) exited normally]
(gdb)
```

Выполнение лабораторной работы

3. Вывел первые 9 строк файла main Затем вывел с 12 по 15 строки. (рис. 3)

```
[Inferior 1 (process 65375) exited normally]
(gdb) list
1       ////////////////////////////////////////
2       // main.c
3
4       #include <stdio.h>
5       #include "calculate.h"
6       int main (void)
7       {
8           float Numeral;
9           char Operation[4];
10          float Result;
(gdb) list 12,15
12          scanf("%f",&Numeral);
13          printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
14          scanf("%s",Operation);
15          Result = Calculate(Numeral, Operation);
(gdb)
```

Figure 3: Вывод команды list

Выполнение лабораторной работы

4. Поставил точку остановы.(рис. 4)

```
15          Result = Calculate(Numeral, Operation);
(gdb) break 16
Breakpoint 1 at 0x4014f2: file main.c, line 16.
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x00000000004014f2 in main at main.c:16
(gdb)
```

Figure 4: Точка остановы

Выполнение лабораторной работы

5. Снова запустил калькулятор.(рис. 5)

```
(gdb) run
Starting program: /home/aamalsagov/work/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): -
Вычитаемое: 3

Breakpoint 1, main () at main.c:16
16          printf("%6.2f\n",Result);
(gdb)
```

Figure 5: Запуск программы

Выполнение лабораторной работы

6. Вывел значение Numeral. (рис. 6)

```
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Вычитаемое: 3

Breakpoint 1, main () at main.c:16
16          printf("%6.2f\n",Result);
(gdb) print Numeral
$1 = 5
(gdb) display Numeral
1: Numeral = 5
(gdb)
```

Figure 6: Значение Numeral

Выполнение лабораторной работы

7. С помощью утилиты splint проанализировал коды файлов calculate.c и main.c. (рис. 7)

```
[aamalsagov@aamalsagov 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:16:9: 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:22:9: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:28:5: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:34:5: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:35:8: 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:38:15: Return value type double does not match declared type float:
                     (HUGE_VAL)
  To allow all numeric types to match, use +relaxtypes.
calculate.c:46:5: Return value (type int) ignored: scanf("%f", &Sec...
```

Figure 7: splint

Выводы

Мы создали простейший калькулятор.