

Лабораторная работа №13

Демидова Е.А.

Операционные системы – 2022

Российский Университет дружбы народов

Введение

Цель работы

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.

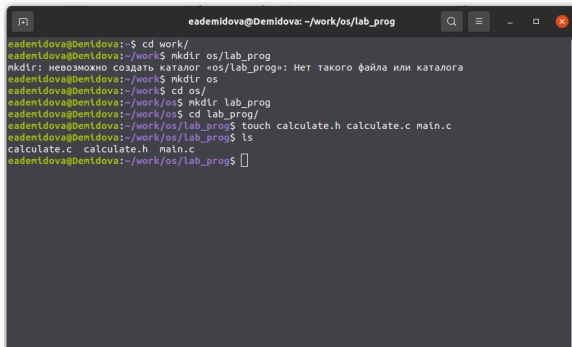
Задачи

1. В домашнем каталоге создайте подкаталог `~/work/os/lab_prog`.
2. Создайте в нём файлы: `calculate.h`, `calculate.c`, `main.c`. Это будет примитивнейший калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, брать квадратный корень, вычислять `sin`, `cos`, `tan`. При запуске он будет запрашивать первое число, операцию, второе число. После этого программа выведет результат и остановится.
3. Выполните компиляцию программы посредством `gcc`.
4. При необходимости исправьте синтаксические ошибки.
5. Создайте `Makefile`.
6. С помощью `gdb` выполните отладку программы `calcul`.
7. С помощью утилиты `splint` попробуйте проанализировать коды файлов `calculate.c` и `main.c`.

Результаты работы

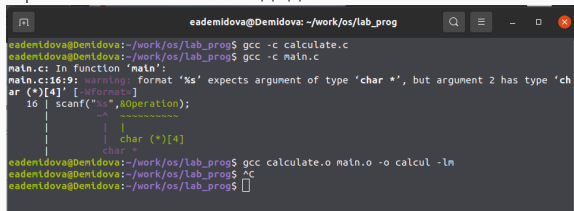
Создание каталога и файлов

В домашнем каталоге создадим подкаталог `~/work/os/lab_prog`. Создадим в нём файлы: `calculate.h`, `calculate.c`, `main.c`. Это будет примитивнейший калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, брать квадратный корень, вычислять `sin`, `cos`, `tan`. При запуске он будет запрашивать первое число, операцию, второе число. После этого программа выведет результат и остановится.

A terminal window titled 'eademidova@Demidova: ~/work/os/lab_prog' with standard window controls. The terminal shows a sequence of commands to create a directory structure and files. The first command 'cd work/' changes the directory. The second 'mkdir os/lab_prog' fails with an error message. The third 'mkdir os' succeeds. The fourth 'cd os/' changes the directory. The fifth 'mkdir lab_prog' succeeds. The sixth 'cd lab_prog/' changes the directory. The seventh 'touch calculate.h calculate.c main.c' creates the three files. The eighth 'ls' lists the files. The prompt returns after each command.

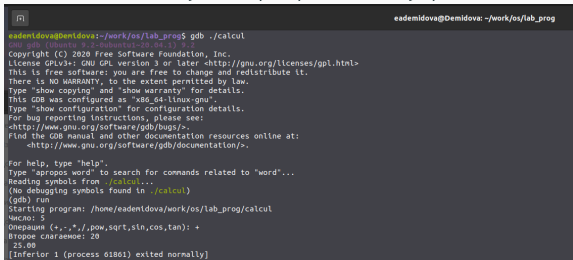
```
eademidova@Demidova:~$ cd work/
eademidova@Demidova:~/work$ mkdir os/lab_prog
mkdir: невозможно создать каталог «os/lab_prog»: Нет такого файла или каталога
eademidova@Demidova:~/work$ mkdir os
eademidova@Demidova:~/work$ cd os/
eademidova@Demidova:~/work/os$ mkdir lab_prog
eademidova@Demidova:~/work/os$ cd lab_prog/
eademidova@Demidova:~/work/os/lab_prog$ touch calculate.h calculate.c main.c
eademidova@Demidova:~/work/os/lab_prog$ ls
calculate.c calculate.h main.c
eademidova@Demidova:~/work/os/lab_prog$
```

Выполним компиляцию программы посредством gcc. Затем исправим ошибку в файле main.c, уберем амперсант перед считыванием Operation. А также создадим Makefile.



```
eademidova@Demidova: ~/work/os/lab_prog
eademidova@Demidova:~/work/os/lab_prog$ gcc -c calculate.c
eademidova@Demidova:~/work/os/lab_prog$ gcc -c main.c
main.c: In function 'main':
main.c:16:9: warning: format '%s' expects argument of type 'char *', but argument 2 has type 'char (*)[4]' [-Wformat=]
   16 | scanf("%s", &operation);
      |          ^~~~~~
      |          |
      |          char (*)[4]
      |          char *
eademidova@Demidova:~/work/os/lab_prog$ gcc calculate.o main.o -o calcul -lm
eademidova@Demidova:~/work/os/lab_prog$ ^C
eademidova@Demidova:~/work/os/lab_prog$
```

С помощью gdb выполним отладку программы calcul. Запустим отладчик GDB, загрузив в него программу для отладки с помощью команды `gdb ./calcul`. Для запуска программы внутри отладчика введём команду `run`.



```
eademidova@Demidova: ~/work/os/lab_prog
eademidova@Demidova:~/work/os/lab_prog$ gdb ./calcul
GNU gdb (Ubuntu 9.2-2ubuntu1-20.04.1) 9.2
Copyright (C) 2020 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-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://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/eademidova/work/os/lab_prog/calcul
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 20
25.00
[Inferior 1 (process 61861) exited normally]
```


Для постраничного (по 9 строк) просмотра исходного код используем команду `list`. Для просмотра строк с 12 по 15 основного файла используем `list` с параметрами. Для просмотра определённых строк не основного файла используем `list` с параметрами.

```
Reading symbols from ./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("Введи: ");
14     scanf("%f",&Numeral);
15     printf("Осепане (%f,%f,%f,pow,sqrt,sin,cos,tan): ");
(gdb) list calculate.c:22-29
```

Установим точку останова в файле `calculate.c` на строке номер 12.
Выведем информацию об имеющихся в проекте точка останова.

```
(gdb) list main.c:4,7
4      #include <stdio.h>
5      #include "calculate.h"
6
7      int
(gdb) list main.c:4,7
4      #include <stdio.h>
5      #include "calculate.h"
6
7      int
(gdb) break 12
Breakpoint 1 at 0x1558: file main.c, line 13.
(gdb) info breakpoints
Num   Type             Disp Enb Address            What
1     breakpoint      keep y   0x0000000000000158  in main at main.c:13
(gdb) □
```

Запуск программы с точкой останова

Запустим программу внутри отладчика и убедимся, что программа остановится в момент прохождения точки останова.

```
1 breakpoint keep y 0x0000000000001550 in main at main.c:13
(gdb) run
Starting program: /home/eademidova/work/os/lab_prog/calcul

Breakpoint 1, main () at main.c:13
13 printf("Число: ");
(gdb) backtrace
#0 main () at main.c:13
(gdb) █
```

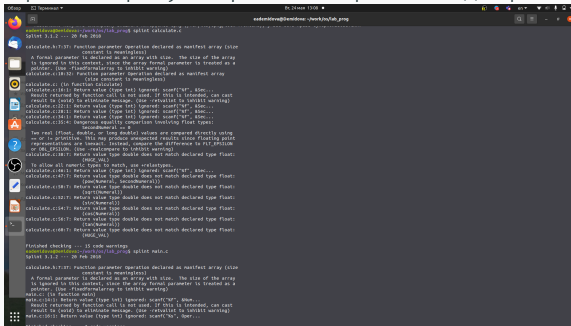
Значение переменной Numeral

Посмотрим, чему равно на этом этапе значение переменной Numeral.
Сравним с результатом вывода на экран, они равны. Уберём точки

останова.

```
#0 main () at main.c:13
(gdb) print Numeral
$2 = 3.0611365e-41
(gdb) display Numeral
1: Numeral = 3.0611365e-41
(gdb) info breakpoints
Num  Type             Disp Enb Address            What
1     breakpoint         keep y   0x0000555555555558  in main at main.c:13
      breakpoint already hit 1 time
(gdb) delete 1
(gdb)
```

С помощью утилиты splint попробуем проанализировать коды файлов



```
cs@csdmldev@csdmldev:~/work/jn/fab_prog$ splint calculate.c
Splint 3.1.2 --- 20 Feb 2008

calculate.c:17:37: 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 #define FORMAL to inhibit warning)
calculate.c:18:32: Function parameter operation declared as manifest array
(size constant is meaningless)
calculate.c: (In function calculate)
calculate.c:118:1: Return value (type int) ignored: scanf("%d", &len)...
Result returned by function call is not used. If this is intended, can cast
result to void to eliminate message. (Use #returnval to inhibit warning)
calculate.c:121:1: Return value (type int) ignored: scanf("%d", &len)...
calculate.c:124:1: Return value (type int) ignored: scanf("%d", &len)...
calculate.c:126:1: Return value (type int) ignored: scanf("%d", &len)...
calculate.c:135:4: Inequality equality comparison involving float types:
float <math>2.0 < 1</math>
Two real (float, double, or long double) values are compared directly using
== or != operator. This may produce unexpected results: since floating point
representations are human-, instead, compare the difference to FLT_EPSILON
or DBL_EPSILON. (Use #realcompare to inhibit warning)
calculate.c:138:7: Return value type double does not match declared type float:
(float <math>2.0</math>)
To allow all numeric types to match, use #realtypes.
calculate.c:141:1: Return value (type int) ignored: scanf("%d", &len)...
calculate.c:147:7: Return value type double does not match declared type float:
(float <math>2.0</math> * <math>1000000000.0</math>)
calculate.c:158:7: Return value type double does not match declared type float:
(float <math>2.0</math> * <math>1000000000.0</math>)
calculate.c:162:7: Return value type double does not match declared type float:
(float <math>2.0</math> * <math>1000000000.0</math>)
calculate.c:164:7: Return value type double does not match declared type float:
(float <math>2.0</math> * <math>1000000000.0</math>)
calculate.c:166:7: Return value type double does not match declared type float:
(float <math>2.0</math> * <math>1000000000.0</math>)
calculate.c:168:7: Return value type double does not match declared type float:
(float <math>2.0</math> * <math>1000000000.0</math>)
Finished checking --- 15 code warnings
cs@csdmldev@csdmldev:~/work/jn/fab_prog$ splint main.c
Splint 3.1.2 --- 20 Feb 2008

calculate.c:17:37: 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 #define FORMAL to inhibit warning)
main.c:118 Function main
main.c:121:1: Return value (type int) ignored: scanf("%d", &len)...
Result returned by function call is not used. If this is intended, can cast
result to void to eliminate message. (Use #returnval to inhibit warning)
main.c:124:1: Return value (type int) ignored: scanf("%d", &len)...
main.c:126:1: Return value (type int) ignored: scanf("%d", &len)...
main.c:135:4: Inequality equality comparison involving float types:
float <math>2.0 < 1</math>
Two real (float, double, or long double) values are compared directly using
== or != operator. This may produce unexpected results: since floating point
representations are human-, instead, compare the difference to FLT_EPSILON
or DBL_EPSILON. (Use #realcompare to inhibit warning)
main.c:138:7: Return value type double does not match declared type float:
(float <math>2.0</math>)
```

calculate.c и main.c.

В результате выполнения лабораторной работы 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.

1. Отладчик GDB [Электронный ресурс]. Maxim Chirkov, 2002. URL:
https://www.opennet.ru/docs/RUS/linux_base/node199.html.

Спасибо за внимание