

Лабораторная работы №13 “Средства, применяемые при разработке программного обеспечения в ОС типа UNIX/Linux”

Мажитов М.А.

RUDN University, Moscow, Russian Federation

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.

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. Выполнить компиляцию программы посредством gcc.
4. Создать Makefile.
5. С помощью gdb выполнить отладку программы calcul (перед использованием gdb исправьте Makefile):
 - Запустить отладчик GDB, загрузив в него программу для отладки
 - Для запуска программы внутри отладчика ввести команду run
 - Для постраничного (по 9 строк) просмотра исходного кода использовать команду list

- Для просмотра строк с 12 по 15 основного файла использовать `list` с параметрами
- Для просмотра определённых строк не основного файла использовать `list` с параметрами
- Установить точку останова
- Вывести информацию об имеющихся в проекте точка останова
- Запустить программу внутри отладчика и убедитесь, что программа остановится в момент прохождения точки останова
- Посмотреть, чему равно на этом этапе значение переменной `Numeral`
- Сравнить с результатом вывода на экран после использования команды `display Numeral`
- Убрать точки останова

1. Создал в каталоге

/home/mamazhitov/work/study/2021-2022/Операционные системы
подкаталог *lab_prog*. Создал в нем файлы *calculate.h*, *calculate.c*, *main.c*.
Скопировал весь код этих файлов из лабораторки. Выполнил
компиляцию этих файлов. (рис. 1)

```
(gdb) q  
[mamazhitov@fedora lab_prog]$ gcc -c calculate.c  
[mamazhitov@fedora lab_prog]$ gcc -c -g main.c  
[mamazhitov@fedora lab_prog]$ gcc calculate.o main.o -o calcul -lm  
[mamazhitov@fedora lab_prog]$ gdb ./calcul
```

Figure 1: Компиляция файлов

2. С помощью gdb выполнил отладку, загрузив в него программу.
Проверил работу калькулятора.(рис. 2)

```
(gdb) run
Starting program: /home/namazhitov/work/study/2021-2022/Операционные системы/study_2021-2022_os-intro/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: 6
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Вычитаемое: 2
4.00
```

Figure 2: Работа калькулятора

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

```
(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("Onepaueя (+,-,*,/,pow,sqrt,sin,cos,tan): ");
14     scanf("%s",Operation);
15     Result = Calculate(Numeral, Operation);
(gdb) list calculate:20,29
No source file named calculate.
```

Figure 3: вывод команды list

4. Поставил точку останова на 16 строке. Вывел информации о точках останова. (рис. 4)

```
(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) run
```

Figure 4: Точки останова


5. Снова запустил программу. (рис. 5)

```
(gdb) run
Starting program: /home/pamazhitov/work/study/2021-2022/Операционные системы/study_2021-2022_os-intro/lab_prog/calcul
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 6
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Вычитаемое: 4

Breakpoint 1, main () at main.c:16
16      printf("%0.2f\n",Result);
(gdb) backtrace
#0  main () at main.c:16
```

Figure 5: Запуск программы

6. Вывел значения переменной Numeral различными командами. (рис. 6)



```
(gdb) print Numeral
$1 = 6
(gdb) display Numeral
1: Numeral = 6
```

Figure 6: Вывод Numeral

7. Удалил точку останова. (рис. 7)

```
1: NumerAc = 0
(gdb) info breakpoints
Num      Type             Disp Enb Address            What
1        breakpoint      keep y   0x00000000004014f2 in main at main.c:16
          breakpoint already hit 1 time
(gdb) delete 1
(gdb)
```

Figure 7: Удаление точки останова

8. С помощью утилиты splint проанализировал коды файлов calculate.c и main.c. (рис. 8)

[illegible]

Figure 8: Вывод команды splint

Мы создали простейший калькулятор на языке C.