Лабораторная работа №13

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

Шуплецов А. А.

6 мая 2023

Российский университет дружбы народов, Москва, Россия

Информация

Докладчик

- Шуплецов Александр Андреевич
- студент ФФМиЕН
- Российский университет дружбы народов
- https://github.com/winnralex

Цели и задачи

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 работы

В домашнем каталоге создадим подкаталог ~/work/os/lab_prog.

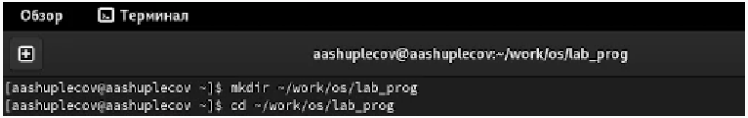


Рис. 1: создание подкаталога

Создадим в нём файлы: calculate.h, calculate.c, main.c.

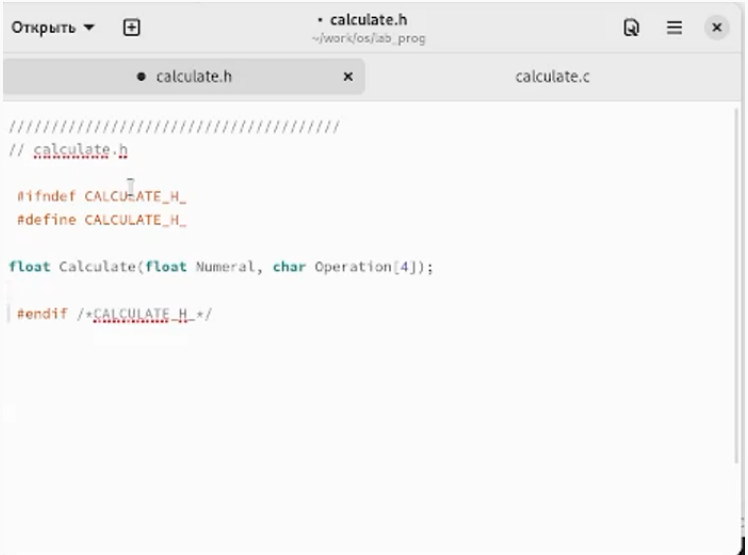
```
[aashuplecov@aashuplecov ~]$ cd ~/work/os/lab_prog
[aashuplecov@aashuplecov lab_prog]$ touch calculate.h calculate.c main.c
[aashuplecov@aashuplecov lab_prog]$ ls
calculate.c  calculate.h  main.c
[aashuplecov@aashuplecov lab_prog]$
```

Рис. 2: создание файлов

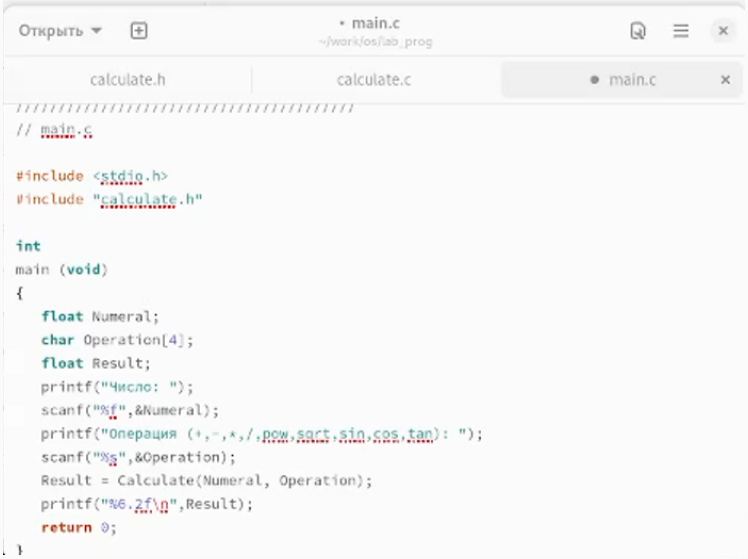
Напишем реализацию функций калькулятора в файле calculate.c.

Рис. 3: реализация функций калькулятора в файле calculate.h;

Напишем интерфейсный файл calculate.h, описывающий формат вызова функции калькулятора.



Напишем основной файл main.c, реализующий интерфейс пользователя к калькулятору.



Выполним компиляцию программы посредством gcc.

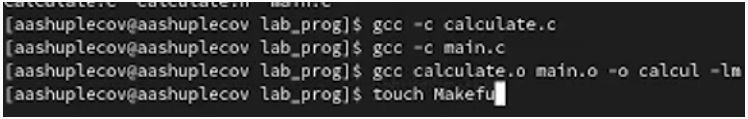
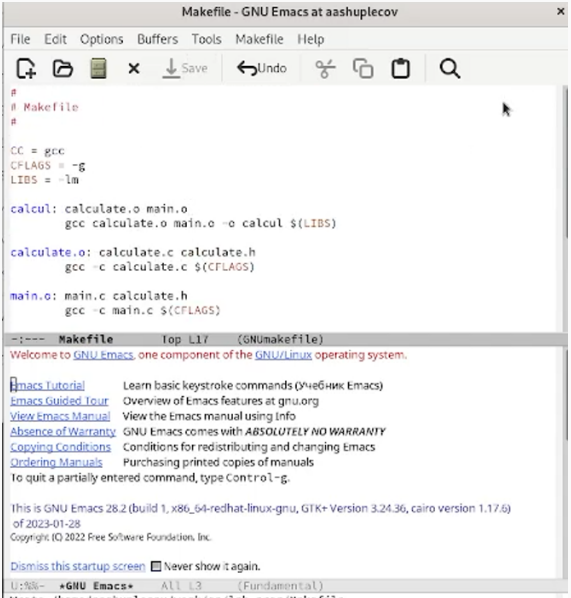


Рис. 6: компиляция программы

Создадим Makefile для нашей программы.



Запустим отладчик программы.

```
(gdb) run
Starting program: /home/aashuplecov/work/os/lab_prog/calcul
Downloading separate debug info for system-supplied DSO at 0x7ffff7fc6000
Downloading separate debug info for /lib64/libm.so.6
Downloading 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: 12
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): *
Второе слагаемое: 12
 24.00
[Inferior 1 (process 4022) exited normally]
```

Рис. 8: отладчик программы

Для постраничного просмотра исходного кода используем команду list.

```
(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;
```

Рис. 9: команда list

Для просмотра определённых строк не основного файла используем list с параметрами.

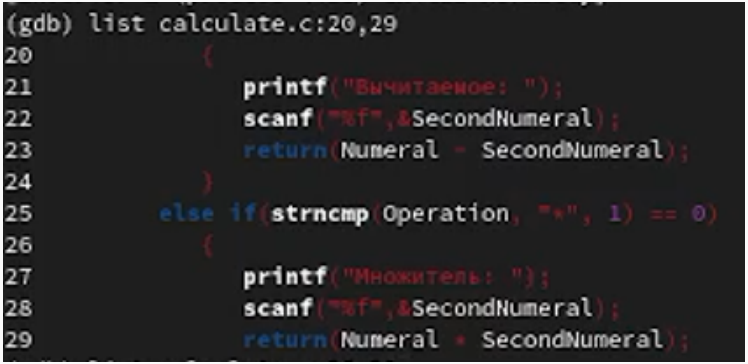


Рис. 10: команда list с параметрами

Установим точку останова в файле calculate.c на строке номер 21.

```
(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.
```

Рис. 11: точка останова

Выведем информацию об имеющихся в проекте точка останова.

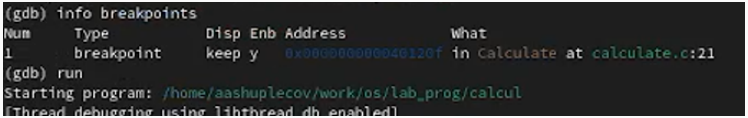


Рис. 12: вывод информации о точках останова

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

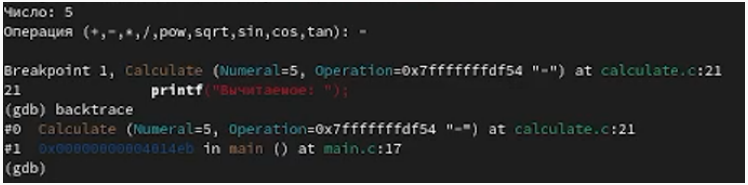


Рис. 13: момент прохождения точки останова

Посмотрим, чему равно на этом этапе значение переменной Numeral.

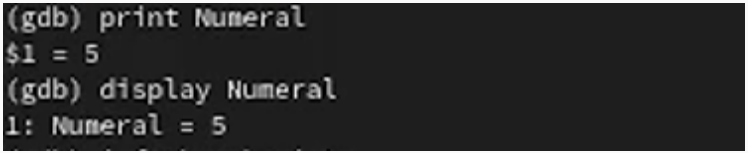


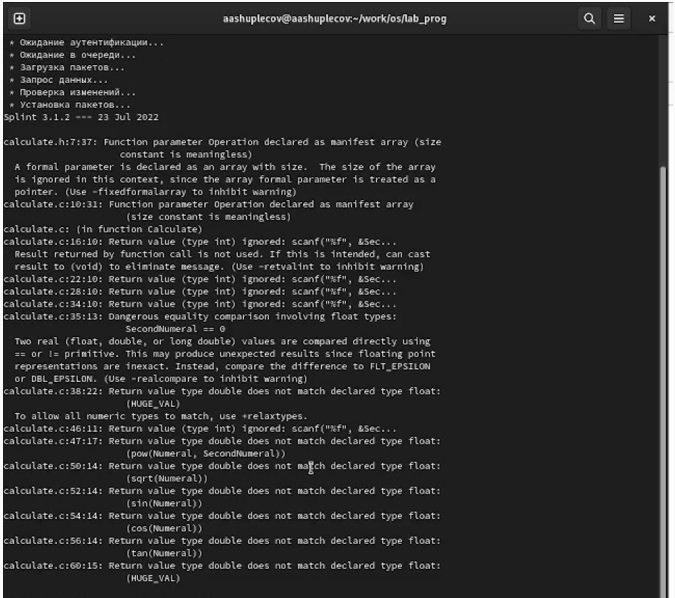
Рис. 14: значение Numeral

Уберем точки останова.

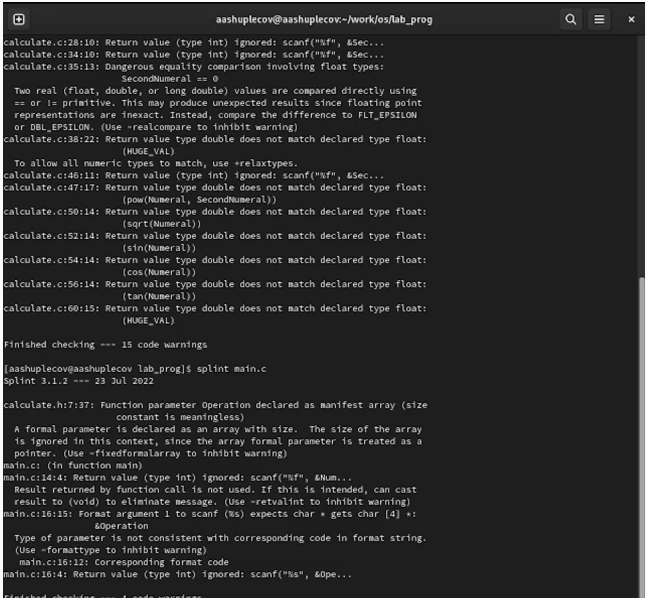
```
(gdb) info breakpoints
Num     Type           Disp Enb Address            What
1       breakpoint     keep y   0x000000000040120f in Calculate at calculate.c:21
        breakpoint already hit 1 time
(gdb) delete 1
(gdb)
```

Рис. 15: удаление точек останова

С помощью утилиты splint проанализируем код файла calculate.c.



С помощью утилиты splint проанализируем код файла main.c.



Вывод

Я приобрел простейшие навыки разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования C калькулятора с простейшими функциями.

Список литературы

Кулябов Д.С. “Материалы к лабораторным работам”