
Front matter

lang: ru-RU title: Средства, применяемые при разработке программного обеспечения в ОС типа UNIX/Linux subtitle: ДР по ОС №14 author: Аникин Константин Сергеевич group: НПИбд-01-20

Formatting

toc-title: "Содержание" toc: true # Table of contents toc_depth: 2 lof: true # List of figures lot: true # List of tables fontsize: 12pt linestretch: 1.5 papersize: a4paper documentclass: screprpt polyglossia-lang: russian polyglossia-otherlangs: english mainfont: PT Serif romanfont: PT Serif sansfont: PT Sans monofont: PT Mono mainfontoptions: Ligatures=TeX romanfontoptions: Ligatures=TeX sansfontoptions: Ligatures=TeX,Scale=MatchLowercase monofontoptions: Scale=MatchLowercase indent: true pdf-engine: lualatex header-includes: - \linepenalty=10 # the penalty added to the badness of each line within a paragraph (no associated penalty node) Increasing the value makes tex try to have fewer lines in the paragraph. - \interlinepenalty=0 # value of the penalty (node) added after each line of a paragraph. - \hyphenpenalty=50 # the penalty for line breaking at an automatically inserted hyphen - \exhyphenpenalty=50 # the penalty for line breaking at an explicit hyphen - \binoppenalty=700 # the penalty for breaking a line at a binary operator - \relpenalty=500 # the penalty for breaking a line at a relation - \clubpenalty=150 # extra penalty for breaking after first line of a paragraph - \widowpenalty=150 # extra penalty for breaking before last line of a paragraph - \displaywidowpenalty=50 # extra penalty for breaking before last line before a display math - \brokenpenalty=100 # extra penalty for page breaking after a hyphenated line - \predisplayspace=10000 # penalty for breaking before a display - \postdisplayspace=0 # penalty for breaking after a display - \floatingpenalty = 20000 # penalty for splitting an insertion (can only be split footnote in standard LaTeX) - \raggedbottom # or \flushbottom - \usepackage{float} # keep figures where there are in the text

- \floatplacement{figure}{H} # keep figures where there are in the text

Цели 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.

Задание

- Ознакомиться с теоретическим материалом.
- Выполнить задание

Теоретическое введение

Стадии цикла разработки ПО [\[1\]](#):

Анализ требований, Проектирование, Разработка, Документация, Тестирование, Внедрение

Стандартным средством для компиляции программ в ОС типа UNIX является GCC (GNU Compiler Collection). Это набор компиляторов для разного рода языков программирования (C, C++, Java, Фортран и др.). Работа с GCC производится при помощи одноимённой управляющей программы gcc, которая интерпретирует аргументы командной строки, определяет и осуществляет запуск нужного компилятора для входного файла.

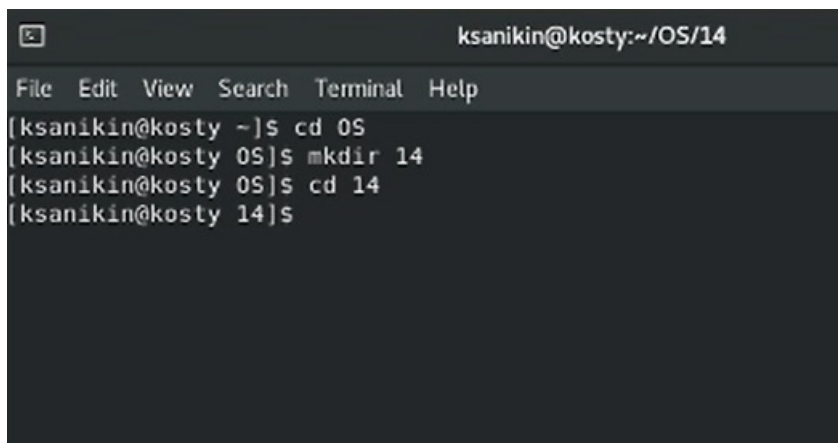
Во время работы над кодом программы программист неизбежно сталкивается с появлением ошибок в ней. Использование отладчика для поиска и устранения ошибок в программе существенно облегчает жизнь программиста. В комплект программ GNU для ОС типа UNIX входит отладчик GDB (GNU Debugger).

Ещё одним средством проверки исходных кодов программ, написанных на языке C, является утилита splint. Эта утилита анализирует программный код, проверяет корректность задания аргументов использованных в программе функций и типов возвращаемых значений, обнаруживает синтаксические и семантические ошибки.

Краткий справочник команд Linux [\[2\]](#)

Выполнение лабораторной работы

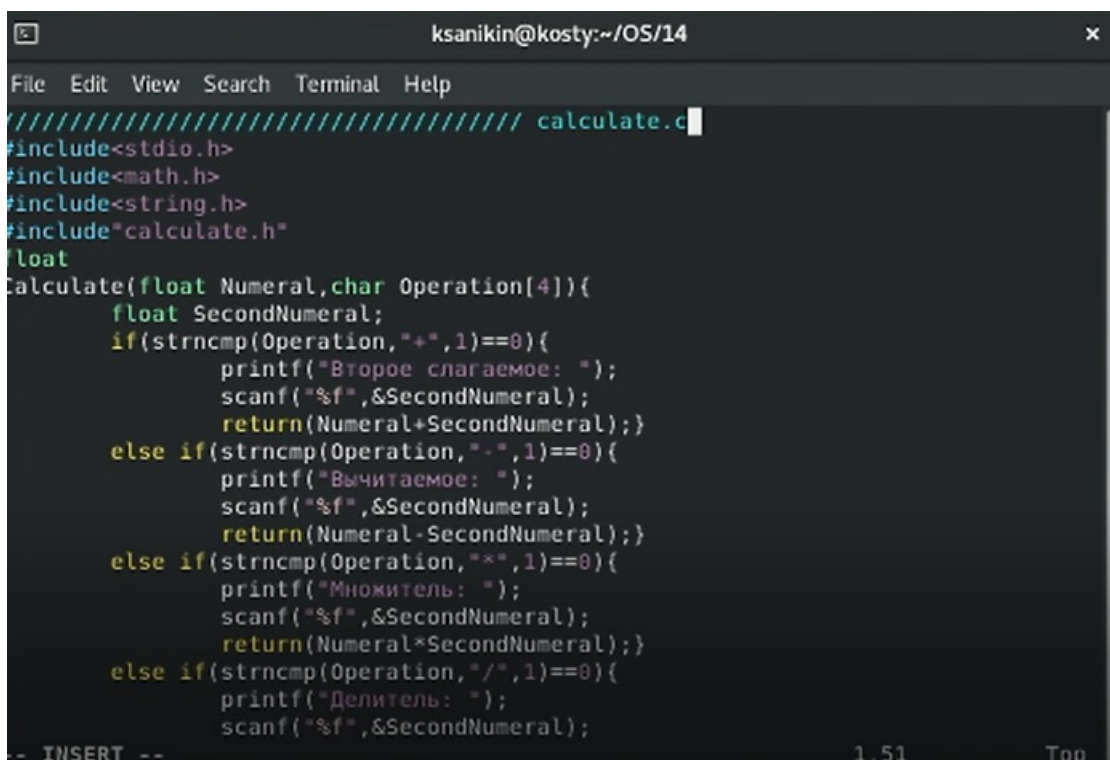
1. Создание папки. Я придерживаюсь другой структуры, отличной от указанной в работе (рис. 1)

A terminal window titled 'ksanikin@kosty:~/OS/14' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and their outputs:

```
[ksanikin@kosty ~]$ cd OS
[ksanikin@kosty OS]$ mkdir 14
[ksanikin@kosty OS]$ cd 14
[ksanikin@kosty 14]$
```

Рис. 1: Создание папки

2.1 Текст файла calculate.c (рис. 2)

A text editor window titled 'ksanikin@kosty:~/OS/14' with a menu bar (File, Edit, View, Search, Terminal, Help). The file 'calculate.c' is open, showing the following code:

```
//////////////////////////////////// calculate.c
#include<stdio.h>
#include<math.h>
#include<string.h>
#include"calculate.h"
float
calculate(float Numeral,char Operation[4]){
    float SecondNumeral;
    if(strncmp(Operation,"+",1)==0){
        printf("Второе слагаемое: ");
        scanf("%f",&SecondNumeral);
        return(Numeral+SecondNumeral);}
    else if(strncmp(Operation,"-",1)==0){
        printf("Вычитаемое: ");
        scanf("%f",&SecondNumeral);
        return(Numeral-SecondNumeral);}
    else if(strncmp(Operation,"*",1)==0){
        printf("Множитель: ");
        scanf("%f",&SecondNumeral);
        return(Numeral*SecondNumeral);}
    else if(strncmp(Operation,"/",1)==0){
        printf("Делитель: ");
        scanf("%f",&SecondNumeral);
```

Рис. 2: Текст файла calculate.c

2.2 Текст файла calculate.h (рис. 3)


```
ksanikin@kosty:~/OS/14
File Edit View Search Terminal Help
[ksanikin@kosty 14]$ gcc -c calculate.c
In file included from calculate.c:6:
calculate.h:5:42: error: expected ')' before '[' token
floatCalculate(floatNumeral,charOperation[4]);
                                         ^
                                         )

[ksanikin@kosty 14]$ vi calculate.h
[ksanikin@kosty 14]$ gcc -c calculate.c
In file included from calculate.c:6:
calculate.h:5:1: warning: data definition has no type or storage class
floatCalculate(float Numeral,char Operation[4]);
^~~~~~
calculate.h:5:1: warning: type defaults to 'int' in declaration of 'floatCalculate' [-Wimplicit-int]
[ksanikin@kosty 14]$ vi calculate.h
[ksanikin@kosty 14]$ gcc -c calculate.c
[ksanikin@kosty 14]$ gcc -c main.c
[ksanikin@kosty 14]$ gcc calculate.o main.o calcul -lm
gcc: error: calcul.o: No such file or directory
gcc: error: calcul: No such file or directory
[ksanikin@kosty 14]$ gcc calculate.o main.o calcul -lm
gcc: error: calcul: No such file or directory
[ksanikin@kosty 14]$ gcc calculate.o main.o -o calcul -lm
[ksanikin@kosty 14]$
```

Рис. 5: Создание исполняемых файлов с помощью gcc

5. Готовый Makefile. Анализ его вынесен отдельным пунктом в конце работы (рис. 6)

```
ksanikin@kosty:~/OS/14
File Edit View Search Terminal Help
# Makefile
#

CC=gcc
CFLAGS=
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 *~

# End Makefile

-- INSERT -- 20,14 Bot
```

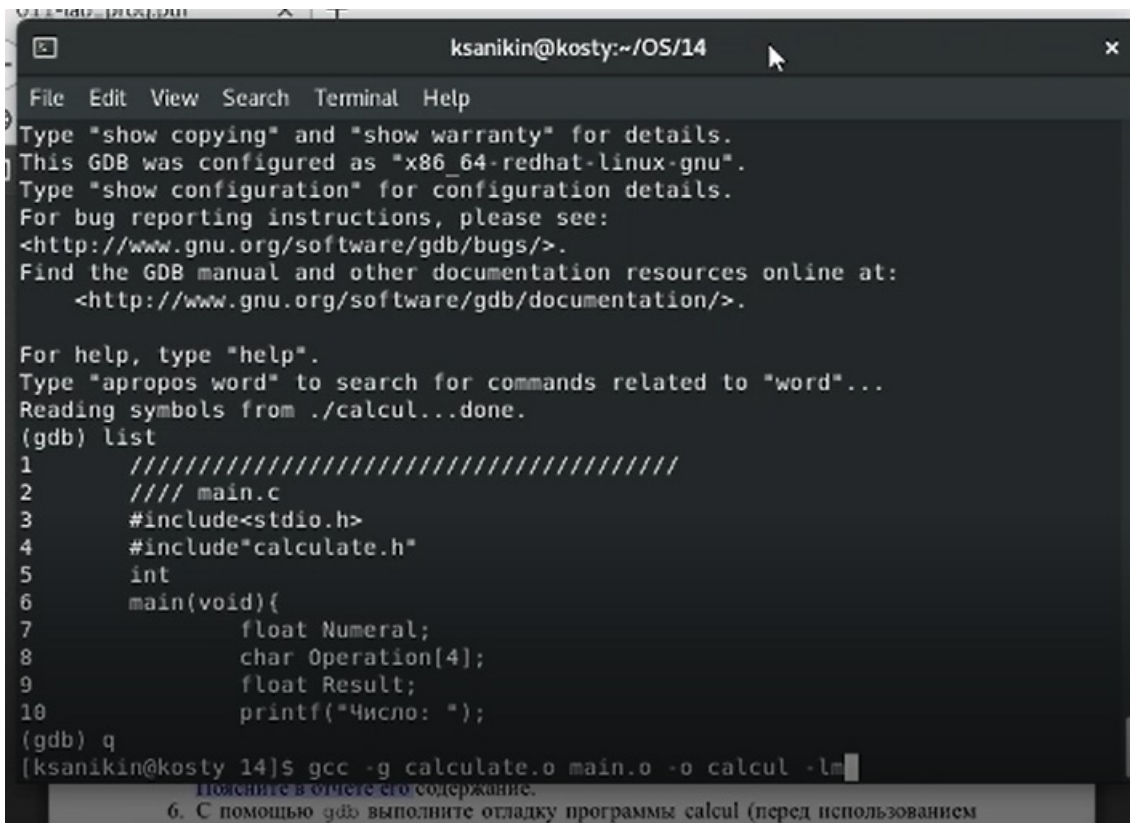
Рис. 6: Готовый Makefile

5.5 Далее в работе упоминается исправление makefile для работы с calculom

Однако для нормального дебаггинга в gdb понадобилось вовсе не исправить makefile

(который в ходе работы вообще не используется)

а добавить атрибут -g ко всем командам пункта 3 (рис. 7)



```
ksanikin@kosty:~/OS/14
File Edit View Search Terminal Help
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:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

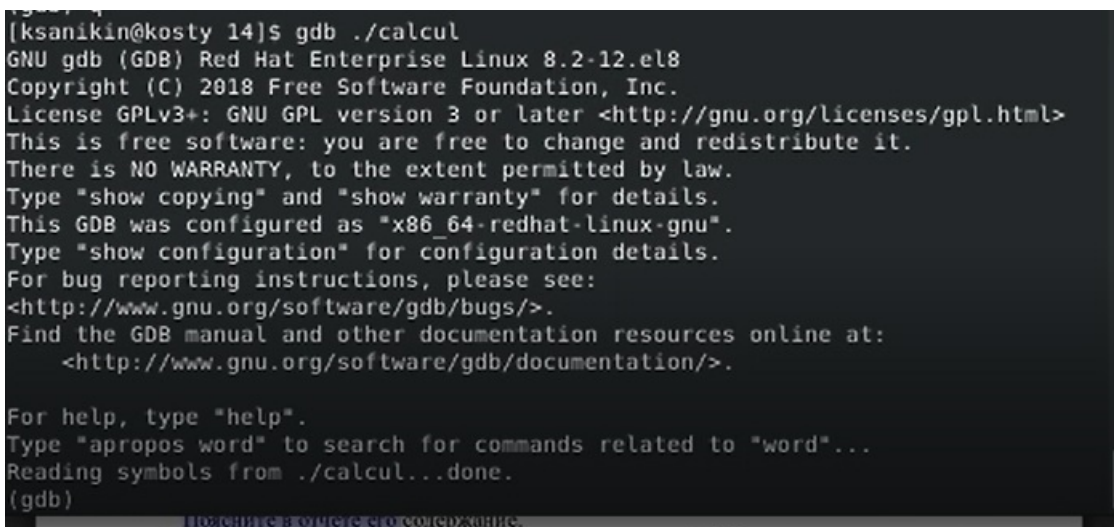
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./calcul...done.
(gdb) list
1  //////////////////////////////////////
2  /// main.c
3  #include<stdio.h>
4  #include"calculate.h"
5  int
6  main(void){
7      float Numeral;
8      char Operation[4];
9      float Result;
10     printf("Число: ");
(gdb) q
[ksanikin@kosty 14]$ gcc -g calculate.o main.o -o calcul -lm
```

Поясните в ответе его содержание.

6. С помощью gdb выполните отладку программы calcul (перед использованием

Рис. 7: Добавление атрибута -g

6.1 Запуск gdb (рис. 8)



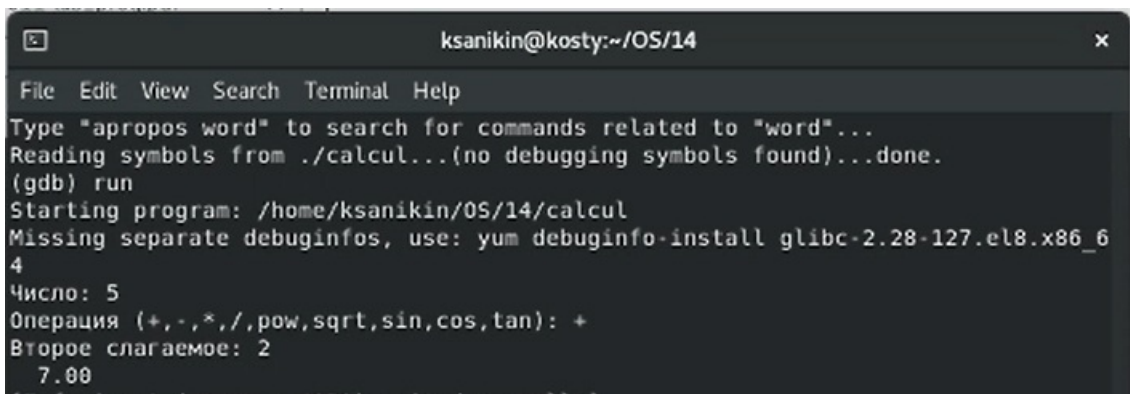
```
[ksanikin@kosty 14]$ gdb ./calcul
GNU gdb (GDB) Red Hat Enterprise Linux 8.2-12.el8
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, 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:
<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...done.
(gdb)
```

Поясните в ответе его содержание.

Рис. 8: gdb./calcul

6.2 Запуск программы командой run (рис. 9)

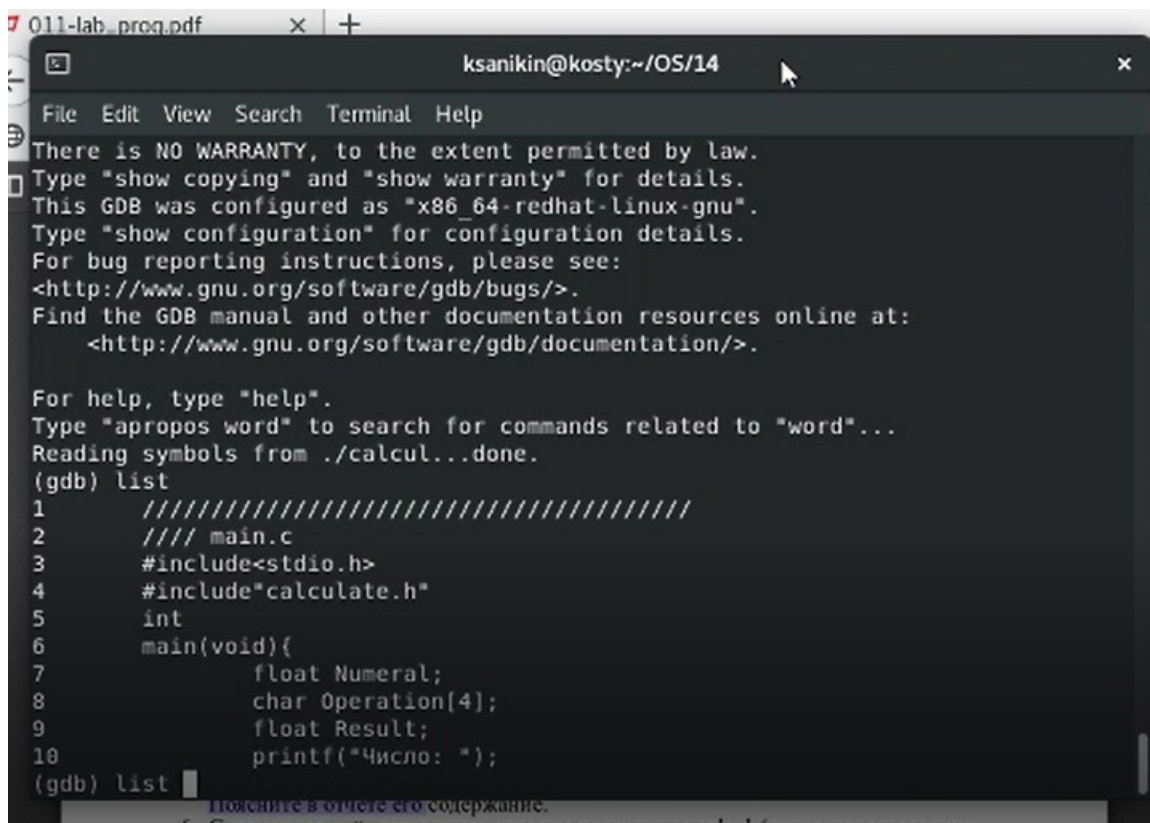


```
[ksanikin@kosty 14]$ gdb ./calcul
GNU gdb (GDB) Red Hat Enterprise Linux 8.2-12.el8
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, 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:
<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...(no debugging symbols found)...done.
(gdb) run
Starting program: /home/ksanikin/OS/14/calcul
Missing separate debuginfos, use: yum debuginfo-install glibc-2.28-127.el8.x86_64
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): +
Второе слагаемое: 2
7.00
(gdb)
```

Рис. 9: Команда run

6.3 Команда list (рис. 10)



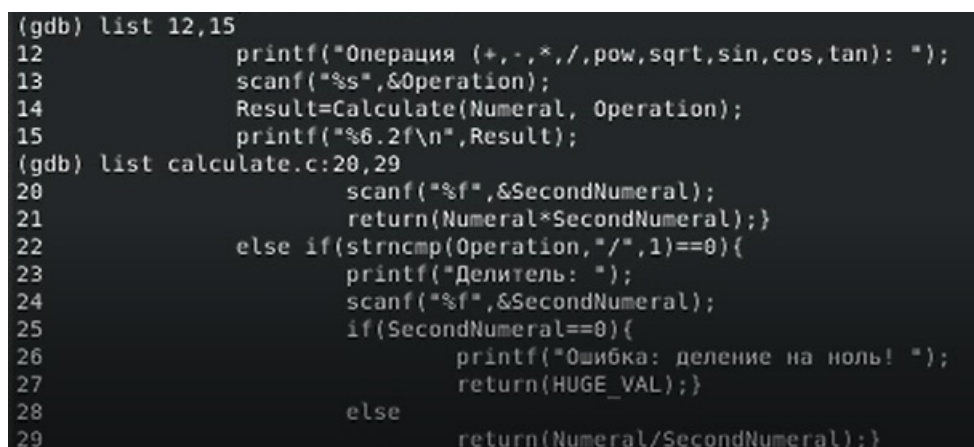
A terminal window titled 'ksanikin@kosty:~/OS/14' showing the GDB startup sequence. The window has a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The startup text includes a disclaimer, version information ('x86_64-redhat-linux-gnu'), and links to the GDB manual and bug reporting page. After reading symbols from './calcul...', the user enters the 'list' command, which displays the first 10 lines of the source file 'main.c'. The code includes standard headers and defines a 'main' function with variables for a numeral, operation, and result, followed by a printf statement in Russian.

```
ksanikin@kosty:~/OS/14
File Edit View Search Terminal Help
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:
<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...done.
(gdb) list
1  //////////////////////////////////////////
2  /// main.c
3  #include<stdio.h>
4  #include"calculate.h"
5  int
6  main(void){
7      float Numeral;
8      char Operation[4];
9      float Result;
10     printf("Число: ");
(gdb) list
```

Рис. 10: Команда list

6.4-6.5 Еще две команды list в других расцветках (рис. 11)

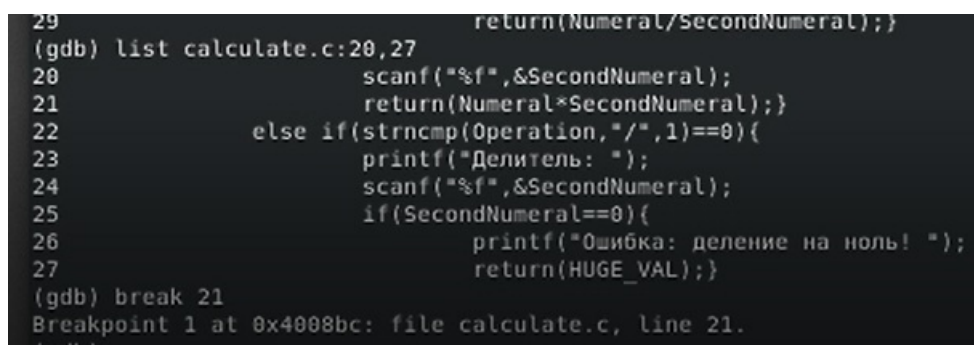


A terminal window showing the 'list' command used with line ranges. The first command 'list 12,15' shows lines 12-15 of the source file, which contain a printf statement for operation prompts, a scanf for the operation, a call to the 'Calculate' function, and another printf for the result. The second command 'list calculate.c:20,29' shows lines 20-29 of the 'calculate.c' file, which include a scanf for the second numeral, a return statement for multiplication, and a conditional block for division that checks for a zero denominator and prints an error message in Russian before returning the result or HUGE_VAL.

```
(gdb) list 12,15
12     printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
13     scanf("%s",&Operation);
14     Result=Calculate(Numeral, Operation);
15     printf("%6.2f\n",Result);
(gdb) list calculate.c:20,29
20     scanf("%f",&SecondNumeral);
21     return(Numeral*SecondNumeral);}
22     else if(strncmp(Operation,"/",1)==0){
23         printf("Делитель: ");
24         scanf("%f",&SecondNumeral);
25         if(SecondNumeral==0){
26             printf("Ошибка: деление на ноль! ");
27             return(HUGE_VAL);}
28         else
29             return(Numeral/SecondNumeral);}
```

Рис. 11: Команды list

6.6-6.7 Еще одна команда list, установка брейк-поинта (рис. 12)

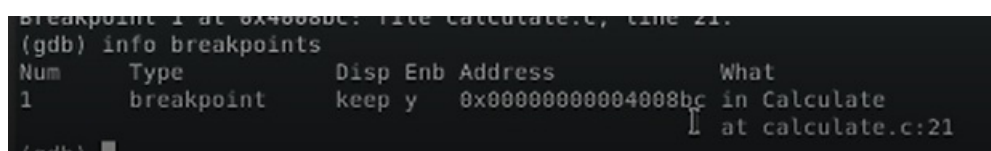


A terminal window showing the 'list' command and setting a breakpoint. The 'list calculate.c:20,27' command shows lines 20-27 of 'calculate.c', which include a scanf for the second numeral, a return for multiplication, and a conditional block for division. The user then enters the 'break 21' command, which sets a breakpoint at line 21 of the file 'calculate.c'.

```
(gdb) list calculate.c:20,27
20     scanf("%f",&SecondNumeral);
21     return(Numeral*SecondNumeral);}
22     else if(strncmp(Operation,"/",1)==0){
23         printf("Делитель: ");
24         scanf("%f",&SecondNumeral);
25         if(SecondNumeral==0){
26             printf("Ошибка: деление на ноль! ");
27             return(HUGE_VAL);}
(gdb) break 21
Breakpoint 1 at 0x4008bc: file calculate.c, line 21.
(gdb)
```

Рис. 12: Установка брейк-поинта

6.8 Информация о брейк-поинтах (рис. 13)



A terminal window showing the 'info breakpoints' command. The output displays a table with columns 'Num', 'Type', 'Disp', 'Enb', 'Address', and 'What'. It shows a single breakpoint at address 0x000000004008bc, which is a 'breakpoint' with 'keep y' disposition, located at 'calculate.c:21'.

```
Breakpoint 1 at 0x4008bc: file calculate.c, line 21.
(gdb) info breakpoints
Num      Type             Disp Enb Address            What
1        breakpoint      keep y  0x000000004008bc  in Calculate
          at calculate.c:21
(gdb)
```

Рис. 13: info breakpoints

6.9 backtrace, сравнение print и display, удаление точки останова (рис. 14)

```
(gdb) run
Starting program: /home/ksanikin/OS/14/calcul
Число: 5
Операция (+, -, *, /, pow, sqrt, sin, cos, tan): *
Множитель: backtrace

Breakpoint 1, Calculate (Numeral=5, Operation=0x7fffffffdf94 "**")
  at calculate.c:21
21      return(Numeral*SecondNumeral);}
(gdb) print Numeral
$1 = 5
(gdb) display Numeral
1: Numeral = 5
(gdb) info breakpoints
Num   Type             Disp Enb Address                  What
1      breakpoint       keep y   0x00000000004008bc  in Calculate
                                at calculate.c:21

    breakpoint already hit 1 time
(gdb) delete 1
(gdb) 2
```

Рис. 14: Оставшиеся пункты работы

- 7. Для этого пункта необходим splint, который я не понял, как установить

Анализ Makefile

Первые три не-комментария - объявление переменных.

Последняя - аргументы для создания calcul

Вторая - пустая, используется при создании исполняемых файлов. Видимо, её можно как-то заполнить извне

Первая нигде не используется. Её смысл мне неясен.

Далее идут четыре метода(?) создания файла.

Первые три идентичны командам создания файлов, вводимых в работе.

Причём первый в ходе работы вызывает второй и третий, как я понял.

Последний же убирает все файлы, по сути - метод удаления.

Выводы

Работа выполнена полностью, если не считать проблем со splintом

Библиографический список

[1]: [Стадии цикла разработки ПО](#)

[2]: [Краткий справочник команд Linux](#)