

Отчёт по тринадцатой лабораторной работе

По дисциплине Операционные Системы

Плугатар Илья Михайлович

0.1 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.

0.2 Задание:

Провести компиляцию и отладку программы с помощью стандартных средств UNIX/Linux-систем.

0.3 Выполнение задания:

1. Создаём подкаталог и файлы для компиляции. Компилируем файлы (с ключом -g для правильности работы некоторых функций отладчика gdb).

```
implugatar@dk8n77 ~ $ cd work/os/lab_prog
implugatar@dk8n77 ~/work/os/lab_prog $ touch calculate.h
implugatar@dk8n77 ~/work/os/lab_prog $ touch calculate.c
implugatar@dk8n77 ~/work/os/lab_prog $ touch main.c
implugatar@dk8n77 ~/work/os/lab_prog $ ls
calculate.c  calculate.h  main.c
implugatar@dk8n77 ~/work/os/lab_prog $
```

```
implugatar@dk8n77 ~/work/os/lab_prog $ gcc -c -g calculate.c
implugatar@dk8n77 ~/work/os/lab_prog $ gcc -c -g main.c
implugatar@dk8n77 ~/work/os/lab_prog $ gcc calculate.o main.o -o calcul -lm
implugatar@dk8n77 ~/work/os/lab_prog $
```

2. Создаём Makefile (пояснение содержания: компиляция файлов программы и очистка).
3. Запускаем дебаггер gdb.

```
implugatar@dk8n77 ~/work/os/lab_prog $ gdb ./calcul
GNU gdb (Gentoo 10.2 vanilla) 10.2
Copyright (C) 2021 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-pc-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://bugs.gentoo.org/>.
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)
```

Рис. 1: Текст, выводимый дебаггером при запуске

4. Запускаем программу внутри отладчика.

```
(gdb) run
Starting program: /afs/.dk.sci.pfu.edu.ru/home/i/m/implugatar/work/os/lab_prog/c
calcul
Число: 3
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): pow
Степень: 4
81.00
[Inferior 1 (process 16583) exited normally]
(gdb)
```

Рис. 2: Тестовый запуск программы и вывод кода завершения без ошибок

5. Используем команду `list` для просмотра кода. Для отображения конкретного диапазона строк вводим команду с параметрами верхней и нижней границы отрывка файла (см. изобр. ниже). Для отображения строк не основного файла указываем в параметрах вторичный файл.

```
(gdb) list
1      #include <stdio.h>
2      #include "calculate.h"
3      int main(void)
4      {
5          float Numeral;
6          char Operation[4];
7          float Result;
8          printf("Число: ");
9          scanf("%f",&Numeral);
10         printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
(gdb)
10         printf("Операция (+,-,*,/,pow,sqrt,sin,cos,tan): ");
(gdb)
11         scanf("%s",&Operation);
12         Result = Calculate(Numeral,Operation);
13         printf("%6.2f\n",Result);
14         return 0;
15     }
(gdb)
```

```
(gdb) list 12,15
12      Result = Calculate(Numeral,Operation);
13      printf("%6.2f\n",Result);
14      return 0;
15  }
```

```
(gdb) list calculate.c:20,29
20      return(Numeral - SecondNumer1);
21  }
22      else if(strncmp(Operation, "*", 1) == 0)
23  {
24      printf("Множитель: ");
25      scanf("%f",&SecondNumer1);
26      return(Numeral * SecondNumer1);
27  }
28      else if(strncmp(Operation, "/", 1) == 0)
29  {
(gdb)
```

6. Размещаем точку останова (брейкпойнт) на строке 21, до момента ввода второго аргумента.

```

(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 0x1241: file calculate.c, line 21.
(gdb)

```

Рис. 3: Установка брейкпойнта

7. Запускаем программу для проверки правильности остановки программы. С помощью команды backtrace выводим информацию об остановке. Отображаем значение Numeral двумя способами. Убираем точки останова.

```

(gdb) run
Starting program: /afs/.dk.sci.pfu.edu.ru/home/i/m/implugatar/work/os/lab_prog/calcul
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Breakpoint 1, Calculate (Numeral=5, Operation=0x7fffffffcf54 "-")
  at calculate.c:21
21     printf("Вычитаемое: ");
(gdb) backtrace
#0 Calculate (Numeral=5, Operation=0x7fffffffcf54 "-") at calculate.c:21
#1 0x0000555555555566 in main () at main.c:12
(gdb)

```

```
(gdb) print Numeral
$1 = 5
(gdb) █
```

```
(gdb) print Numeral
$1 = 5
(gdb) display Numeral
1: Numeral = 5
(gdb) █
```

```
(gdb) info breakpoints
Num      Type           Disp Enb Address                  What
1        breakpoint     keep y   0x0000555555555241 in Calculate
                                                at calculate.c:21
        breakpoint already hit 1 time
(gdb) delete 1
(gdb) █
```

0.4 Заключение

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