

Лабораторная работа №13

Дисциплина: Операционные системы

Кондратьев Арсений Вячеславович

25.09.2022

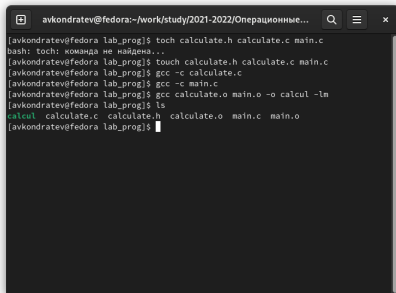
Российский университет дружбы народов, Москва, Россия

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

Выполнение лабораторной работы

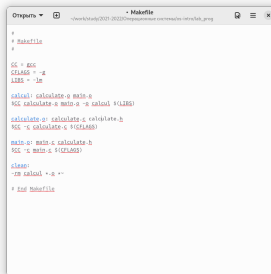
1. Выполнил компиляцию программы посредством gcc(рис.1)

A terminal window titled 'avkondratev@fedora: ~/work/study/2021-2022/Операционные...' showing a series of commands and their outputs. The commands include 'touch', 'gcc -c', and 'gcc' to compile a C program. The output shows the creation of object files and the final executable 'calcul'.

```
avkondratev@fedora lab_prog]$ touch calculate.h calculate.c main.c
bash: touch: команда не найдена...
avkondratev@fedora lab_prog]$ touch calculate.h calculate.c main.c
avkondratev@fedora lab_prog]$ gcc -c calculate.c
avkondratev@fedora lab_prog]$ gcc -c main.c
avkondratev@fedora lab_prog]$ gcc calculate.o main.o -o calcul -lm
avkondratev@fedora lab_prog]$ ls
calcul calculate.c calculate.h calculate.o main.c main.o
avkondratev@fedora lab_prog]$
```

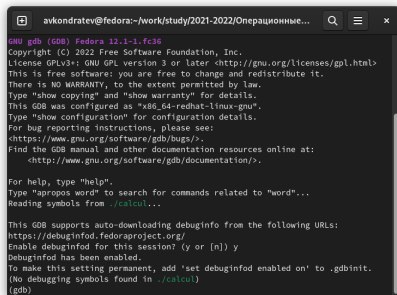
Figure 1: Рис. 1

2. Создал Makefile в котором дописал опцию -g в CFLAGS для корректной компиляции объектных файлов и использовал переменную CC в которую помещен компилятор. Он необходим для автоматической компиляции calculate.c, main.c и объединения в один исполняемый файл calcul. Clean нужна для автоматического удаления файлов. Переменная CFLAGS отвечает за опции. Переменная LIBS отвечает за опции для объединения.(рис.2)



```
#  
# Makefile  
#  
CC = gcc  
CFLAGS = -g  
LIBS = -lm  
  
calcul: calculate.o main.o  
%CC calculate.o main.o -o calcul $(CFLAGS)  
  
calculate.o: calculate.c calculate.h  
%CC -c calculate.c $(CFLAGS)  
  
main.o: main.c calculate.h  
%CC -c main.c $(CFLAGS)  
  
clean:  
-rm calcul *.o *~  
  
# End Makefile
```

3. Запустил отладчик GDB, загрузив в него программу для отладки(рис.3)



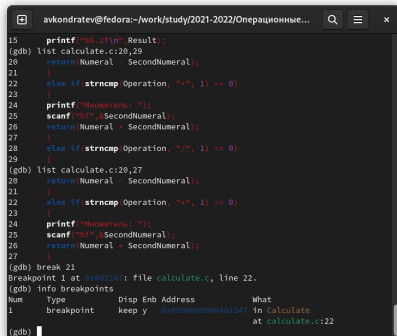
```
GNU gdb (GDB) Fedora 12.1-1.fc36
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" 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:
<https://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...

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.
(No debugging symbols found in ./calcul)
(gdb)
```

Figure 3: Рис. 3

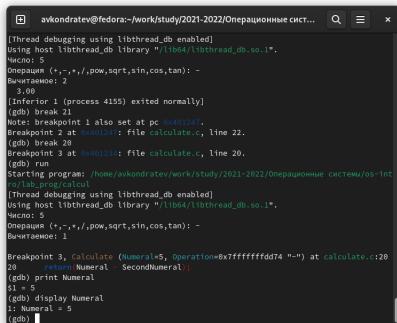
4. Установил точку останова в файле calculate.c на строке номер 21 и вывел информацию о точках останова с помощью info breakpoints(рис.4)



```
avkondratev@fedora:~/work/study/2021-2022/Операционные...
15 printf("%0.2f\n", Result);
(gdb) list calculate.c:20,29
20 return Numeral - SecondNumeral;
21 }
22 else if strcmp(Operation, "+") == 0)
23 {
24 printf("Нужны еще 2");
25 scanf("%f", &SecondNumeral);
26 return Numeral + SecondNumeral;
27 }
28 else if strcmp(Operation, "/") == 0)
29 {
(gdb) list calculate.c:20,27
20 return Numeral - SecondNumeral;
21 }
22 else if strcmp(Operation, "+") == 0)
23 {
24 printf("Нужны еще 2");
25 scanf("%f", &SecondNumeral);
26 return Numeral + SecondNumeral;
27 }
(gdb) break 21
Breakpoint 1 at 0x401247: file calculate.c, line 22.
(gdb) info breakpoints
Num Type      Disp Enb Address             What
1   breakpoint keep y  0x0000000000401247 in calculate
                          at calculate.c:22
(gdb)
```

Figure 4: Рис. 4

- Убедился, что программа останавливается в момент точки останова, с помощью print Numeral выяснил что эта переменная равна 5 на тот момент и сравнил результат с командой display Numeral(рис.5)



```
avkondratev@fedora:~/work/study/2021-2022/Операционные сист...
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Вычитаемое: 2
3.00
[Inferior 1 (process 4155) exited normally]
(gdb) break 21
Note: breakpoint 1 also set at pc 0x401247.
Breakpoint 2 at 0x401247: file calculate.c, line 22.
(gdb) break 20
Breakpoint 3 at 0x401234: file calculate.c, line 20.
(gdb) run
Starting program: /home/avkondratev/work/study/2021-2022/Операционные системы/os-int
no lib_gprof/statout
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Число: 5
Операция (+,-,*,/,pow,sqrt,sin,cos,tan): -
Вычитаемое: 1

Breakpoint 3, calculate (Numeral=5, Operation=0x7fffffffdd74 "-") at calculate.c:20
20     return Numeral - SecondNumeral;
(gdb) print Numeral
$1 = 5
(gdb) display Numeral
1: Numeral = 5
(gdb)
```

Figure 5: Рис. 5

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