

Отчет по лабораторной работе №13

дисциплина Операционные системы

Колобова Елизавета, гр. НММбд-01-22

22 марта 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Колобова Елизавета
- студент
- Российский университет дружбы народов

Вводная часть

- программирование в оболочке ОС UNIX/Linux.

- 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.

- Процессор **pandoc** для входного формата Markdown
- Автоматизация процесса создания: **Makefile**

Создание презентации

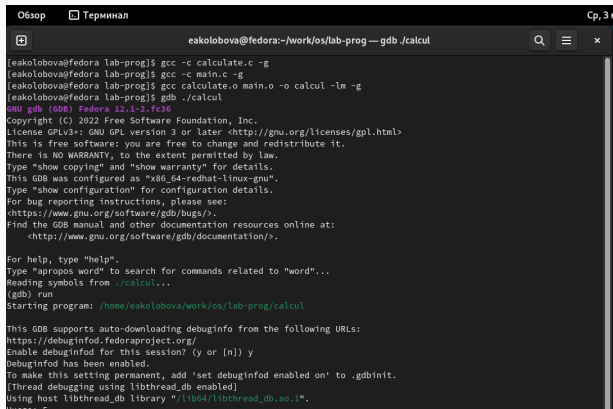
1. В домашнем каталоге создадим подкаталог `~/work/os/lab_prog`
2. Создадим в нём файлы: `calculate.h`, `calculate.c`, `main.c`.
3. Заполним созданные файлы согласно материалам лабораторной работы. Реализация функций калькулятора представлена в файле `calculate.c`, основной файл `main.c` реализует интерфейс пользователя к калькулятору. Синтаксические ошибки утилитой `ыздште` не выявляются, так что их там, скорее всего, нет

Создание Makefile

Создадим Makefile с содержанием, указанным в материалах лабораторной работы. Основное в содержании файла - это указанный в нем порядок компиляции исполняемых файлов. Сразу отредактируем файл, переместив блок с компиляцией программы `calcul` ниже всех остальных, сразу перед блоком `clean`, т.к. для компиляции итоговой программы требуются файлы, которые создаются в блоках выше.

```
1 #
2 # Makefile
3 #
4
5 CC = gcc
6 CFLAGS =
7 LIBS = -lm
8
9 calculate.o: calculate.c calculate.h
10     gcc -c calculate.c $(CFLAGS)
11
12 main.o: main.c calculate.h
13     gcc -c main.c $(CFLAGS)
14
```

Выполним компиляцию программы посредством gcc, как указано в лабораторной работе, с добавлением ключа -g, чтобы далее программу можно было загрузить в отладчик без ошибок. С помощью gdb выполним отладку программы calcul и выполним указанные в материалах к лаб. работе действия



```
Обзор Терминал Cp, 3
eakolobova@fedora:~/work/os/lab-prog — gdb ./calcul

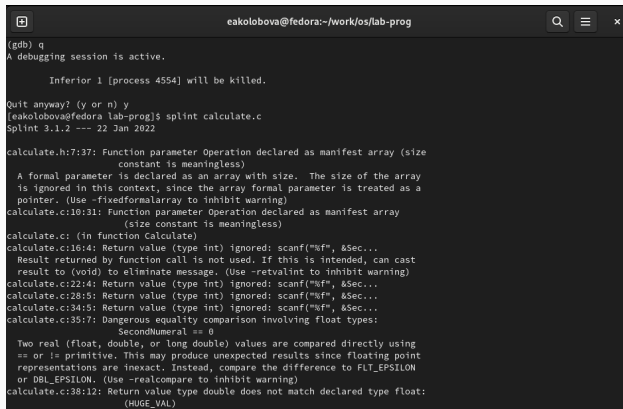
[eakolobova@fedora lab-prog]$ gcc -c calculate.c -g
[eakolobova@fedora lab-prog]$ gcc -c main.c -g
[eakolobova@fedora lab-prog]$ gcc calculate.o main.o -o calcul -lm -g
[eakolobova@fedora lab-prog]$ gdb ./calcul
GNU gdb (GDB) Fedora 12.1-2.fc36
Copyright (C) 2022 Free Software Foundation, Inc.
License 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...
(gdb) run
Starting program: /home/eakolobova/work/os/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".
(gdb) _
```

Исследование с помощью утилиты splint

С помощью утилиты splint попробуем проанализировать коды файлов calculate.c и main.c. В основном выводятся предупреждения о несоответствии типов переменных и возвращаемых значений, игнорировании определенных функций, использовании параметров в кач-ве указателей и прочих сомнительных, но не влияющих на работу программы вещах



```
eakolobova@fedora: ~/work/os/lab-prog
(gdb) q
A debugging session is active.

Inferior 1 [process 4554] will be killed.

Quit anyway? (y or n) y
[eakolobova@fedora lab-prog]$ splint calculate.c
Splint 3.1.2 --- 22 Jan 2022

calculate.h:7:37: Function parameter Operation declared as manifest array (size
constant is meaningless)
A formal parameter is declared as an array with size. The size of the array
is ignored in this context, since the array formal parameter is treated as a
pointer. (Use -fixedformalarray to inhibit warning)
calculate.c:10:31: Function parameter Operation declared as manifest array
(size constant is meaningless)
calculate.c: (in function Calculate)
calculate.c:16:4: Return value (type int) ignored: scanf("%f", &Sec...
Result returned by function call is not used. If this is intended, can cast
result to (void) to eliminate message. (Use -retvalint to inhibit warning)
calculate.c:22:4: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:28:5: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:34:5: Return value (type int) ignored: scanf("%f", &Sec...
calculate.c:35:7: Dangerous equality comparison involving float types:
SecondNumeral == 0
Two real (float, double, or long double) values are compared directly using
== or != primitive. This may produce unexpected results since floating point
representations are inexact. Instead, compare the difference to FLT_EPSILON
or DBL_EPSILON. (Use -realcompare to inhibit warning)
calculate.c:38:12: Return value type double does not match declared type float:
(HUGE_VAL)
```

Результаты

Результатом проделанной работы является Приобретение простейших навыков разработки, анализа, тестирования и отладки приложений в ОС типа UNIX/Linux на примере создания на языке программирования С калькулятора с простейшими функциями.