Лабораторная работа No 14

Средства, применяемые приразработке программного обеспечения в ОС типа UNIX/Linux

Кеан Путхеаро НПИбд-01-20

Содержание

# Цель работы

Приобрести простейшие навыки разработки, анализа, тестирования и отладкиприложений в ОС типа UNIX/Linux на примере создания на языке программирова-ния С калькулятора с простейшими функциями.

# Задание

1. В домашнем каталоге создайте подкаталог ~/work/os/lab\_prog.
2. Создайте в нём файлы: calculate.h, calculate.c, main.c. Это будет примитивнейший калькулятор, способный складывать, вычитать, умножать и делить, возводить число в степень, брать квадратный корень, вычислять sin, cos, tan. При запуске он будет запрашивать первое число, операцию, второе число. После этого программа выведет результат и остановится. Реализация функций калькулятора в файле calculate.h: //////////////////////////////////// // 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); if(SecondNumeral == 0) { printf(”Ошибка: деление на ноль! “); return(HUGE\_VAL); } else return(Numeral / SecondNumeral); } else if(strncmp(Operation,”pow“, 3) == 0) { printf(”Степень: “); scanf(”%f“,&SecondNumeral); return(pow(Numeral, SecondNumeral)); } else if(strncmp(Operation,”sqrt“, 4) == 0) return(sqrt(Numeral)); else if(strncmp(Operation,”sin“, 3) == 0) return(sin(Numeral)); else if(strncmp(Operation,”cos“, 3) == 0) return(cos(Numeral)); else if(strncmp(Operation,”tan“, 3) == 0) return(tan(Numeral)); else { printf(”Неправильно введено действие "); return(HUGE\_VAL); } } Интерфейсный файл calculate.h, описывающий формат вызова функциикалькулятора: /////////////////////////////////////// // calculate.h #ifndef CALCULATE\_H\_ #define CALCULATE\_H\_ float Calculate(float Numeral, char Operation[4]); #endif /*CALCULATE\_H\_*/ Основной файл main.c, реализующий интерфейс пользователя к калькулятору: //////////////////////////////////////// // main.c
3. Выполните компиляцию программы посредством gcc: gcc -c calculate.c gcc -c main.c gcc calculate.o main.o -o calcul -lm
4. При необходимости исправьте синтаксические ошибки.
5. Создайте Makefile со следующим содержанием:

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 Поясните в отчёте его содержание.

1. С помощью gdb выполните отладку программы calcul (перед использованием gdb исправьте Makefile): – Запустите отладчик GDB, загрузив в него программу для отладки: gdb ./calcul – Для запуска программы внутри отладчика введите команду run: run – Для постраничного (по 9 строк) просмотра исходного код используйте команду list: list – Для просмотра строк с 12 по 15 основного файла используйте list с параметрами: list 12,15 – Для просмотра определённых строк не основного файла используйте list с параметрами: list calculate.c:20,29 – Установите точку останова в файле calculate.c на строке номер 21: list calculate.c:20,27 break 21 – Выведите информацию об имеющихся в проекте точка останова: info breakpoints – Запустите программу внутри отладчика и убедитесь, что программа остановится в момент прохождения точки останова: run 5

* backtrace – Отладчик выдаст следующую информацию: #0 Calculate (Numeral=5, Operation=0x7fffffffd280 “-”) at calculate.c:21 #1 0x0000000000400b2b in main () at main.c:17 а команда backtrace покажет весь стек вызываемых функций от начала программы до текущего места. – Посмотрите, чему равно на этом этапе значение переменной Numeral, введя: print Numeral На экран должно быть выведено число 5. – Сравните с результатом вывода на экран после использования команды: display Numeral – Уберите точки останова: info breakpoints delete 1

1. С помощью утилиты splint попробуйте проанализировать коды файлов calculate.c и main.c.

# Выполнение лабораторной работы

1. В домашнем каталоге создал подкаталог ~/work/os/lab\_prog.

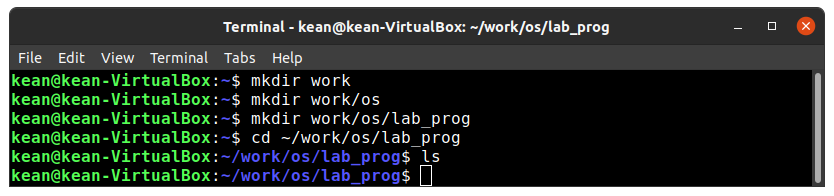


Figure 1: каталог

1. Создал в нём файлы: calculate.h, calculate.c, main.c.

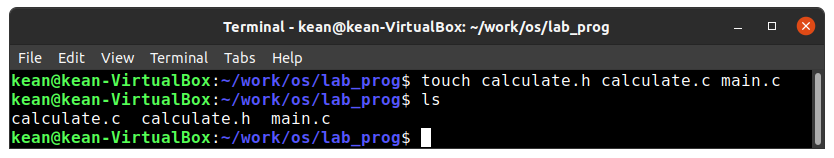


Figure 2: файлы

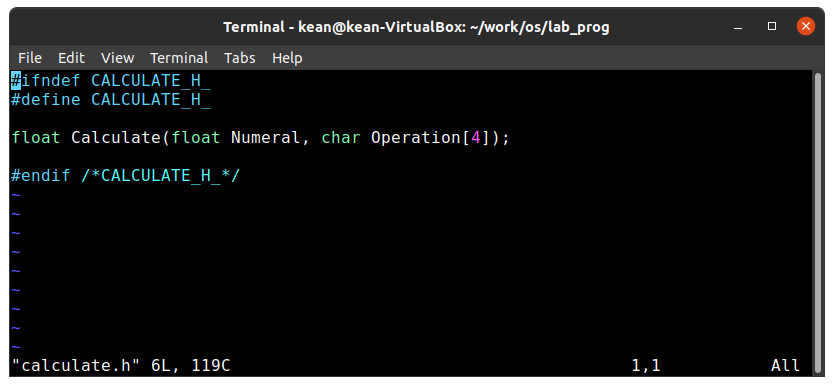


Figure 3: calculate.h файл

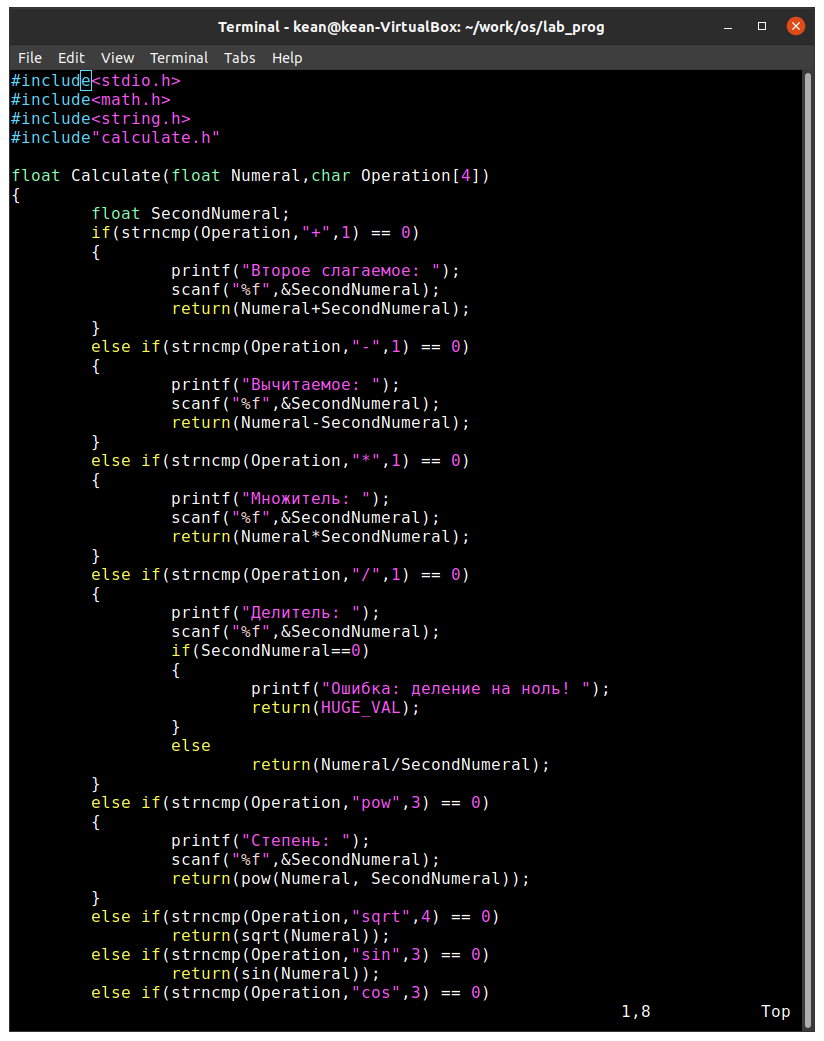


Figure 4: calculate.c файл

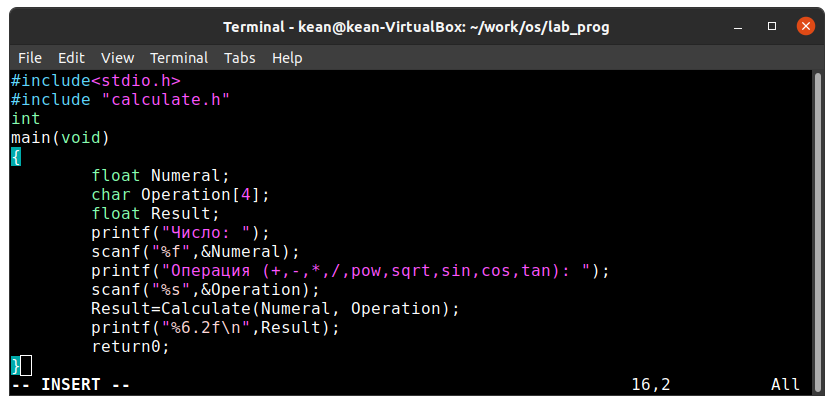


Figure 5: main.c файл

1. Выполнил компиляцию программы посредством gcc

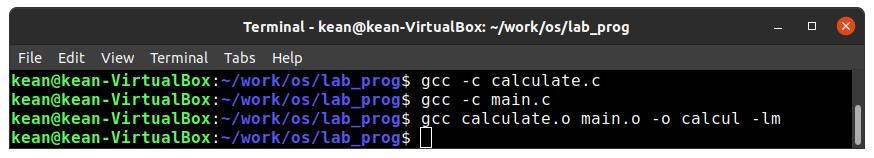


Figure 6: компиляция gcc

1. Исправил синтаксические ошибки в файле main.c (удалил & перед operator в линии scanf(“%s”,&Operation);)
2. Создал Makefile со следующим содержанием

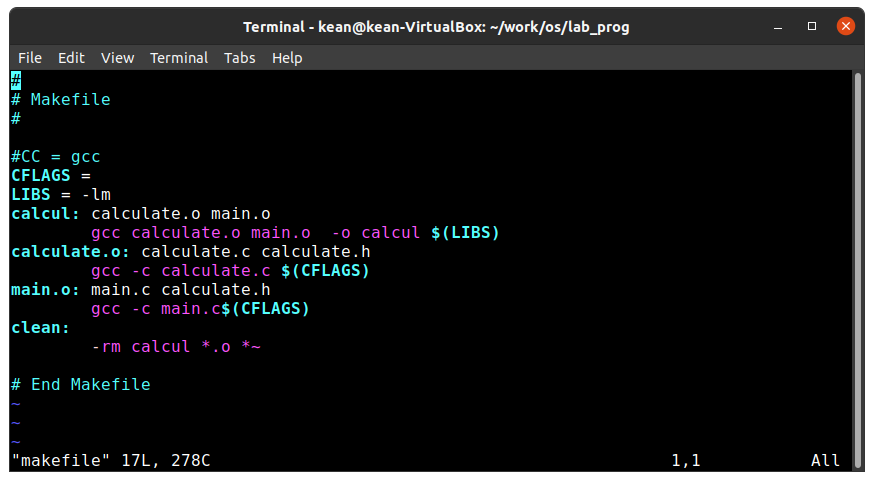


Figure 7: Makefile

1. С помощью gdb выполнил отладку программы calcul (перед использованием gdb исправьте Makefile)

Запустите отладчик GDB, загрузив в него программу для отладки:gdb ./calcul

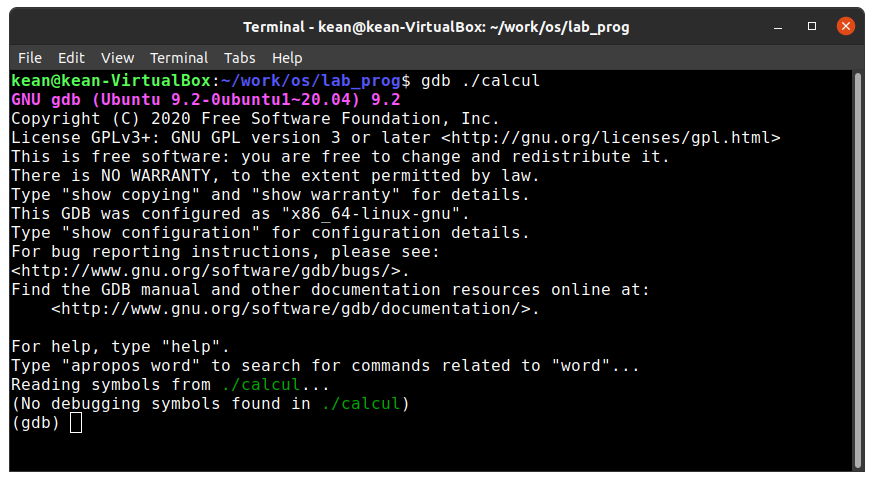


Figure 8: gdb ./calcul

Для запуска программы внутри отладчика ввел команду run: run

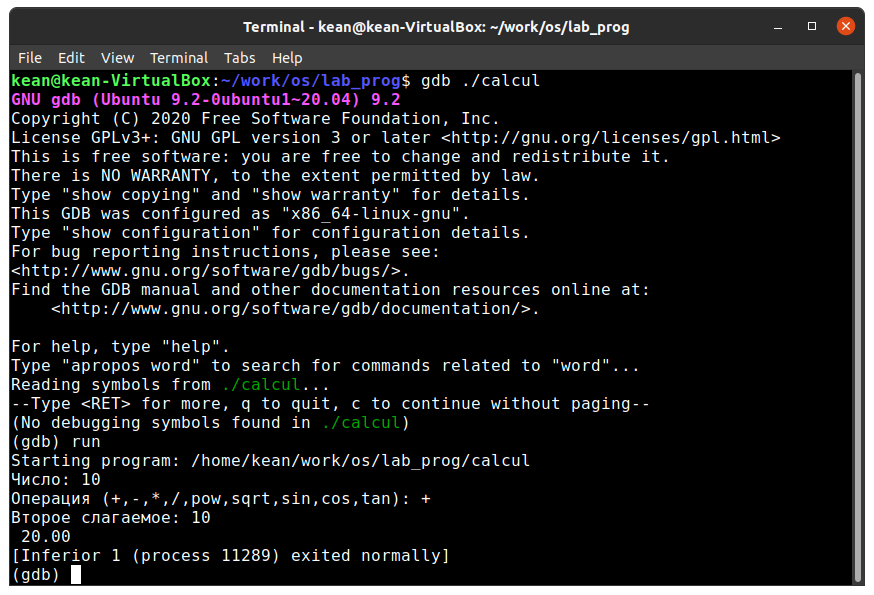


Figure 9: gdb run

1. С помощью утилиты splint попробовал проанализировать коды файлов calculate.c и main.c.

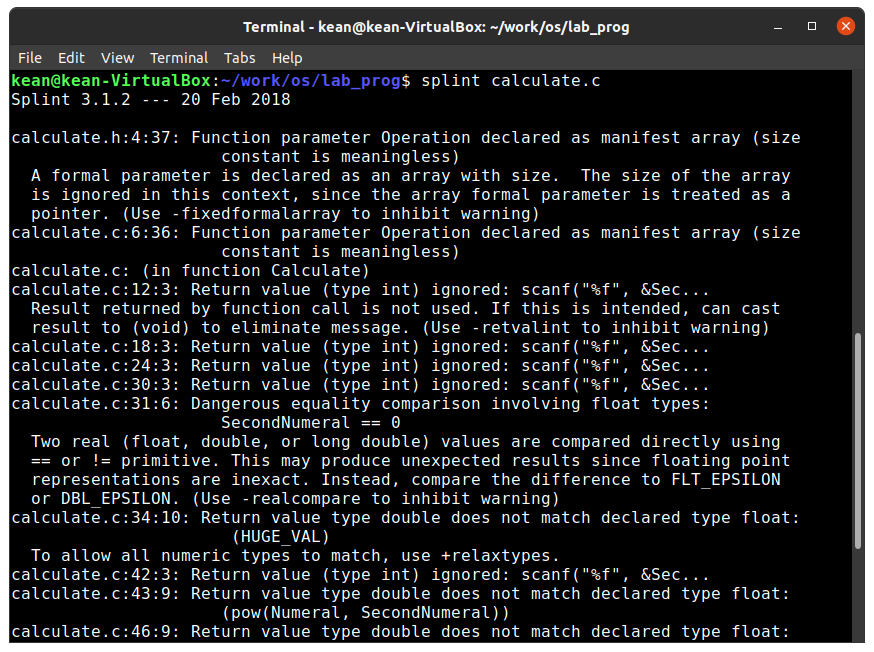


Figure 10: splint calculate.c

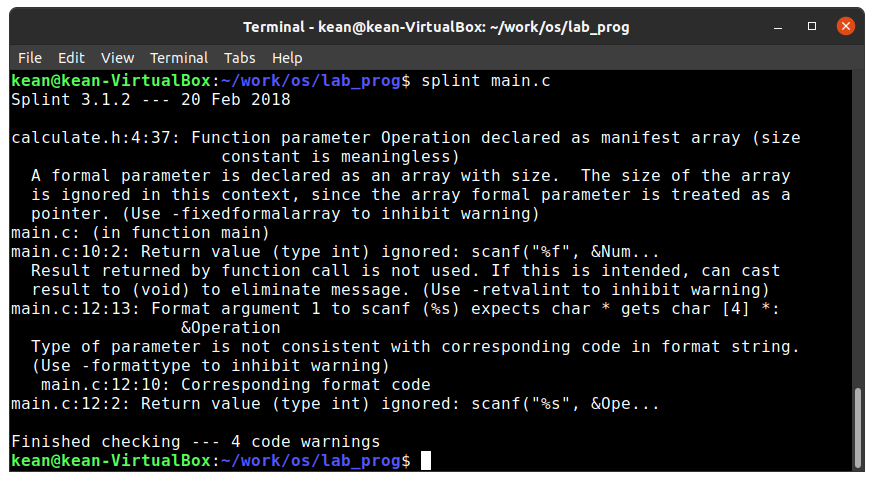


Figure 11: splint main.c

# Вывод

В результате работы , я приобрёл простейшие навыки разработки, анализа, тестирования и отладки приложений в Линукс

# Библиография

1. (Лабораторная работа №14) https://esystem.rudn.ru/pluginfile.php/1142386/mod\_resource/content/2/011-lab\_prog.pdf
2. (stackexchange) https://vi.stackexchange.com/questions/10209/execute-current-buffer-as-bash-script-from-vim
3. (BASH: функция getopts — используем опции в скриптах) https://esystem.rudn.ru/pluginfile.php/1142380/mod\_resource/content/3/009-lab\_shell\_prog\_2.pdf
4. (stackoverflow) https://stackoverflow.com/questions/16483119/an-example-of-how-to-use-getopts-in-bash