Front matter

lang: ru-RU title: Лабораторная работа №13 "Средства, применяемые при разработке программного обеспечения в ОС типа UNIX/Linux" author: Камкина Арина Леонидовна institute: RUDN University, Moscow, Russian Federation date: NEC--2022

Formatting

toc: false slide_level: 2 theme: metropolis header-includes:

- \metroset{progressbar=frametitle,sectionpage=progressbar,numbering=fraction}
- '\makeatletter'
- '\beamer@ignorenonframefalse'
- '\makeatother' aspectratio: 43 section-titles: true

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.

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.
3. 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

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

Выполнение лабораторной работы

1. Создала подкаталог lab_prog и перешла в него (рис. [-@fig:001])

ПСоздание и переход в каталог { #fig:001 width=70% }

2. Создала в нём файлы: calculate.h, calculate.c, main.c (рис. [-@fig:002])

Создание файлов { #fig:002 width=70% }

3. Выполнила компиляцию программы посредством gcc (рис. [-@fig:003])

Компиляция программы { #fig:003 width=70% }

4. Создала Makefile (рис. [-@fig:004])

Создание Makefile { #fig:004 width=70% }

5. Исправила все ошибки и добавила отступы, где нужно (рис. [-@fig:005])

Итоговый Makefile { #fig:005 width=70% }

6. Выполнила отладку через gdb (рис. [-@fig:006])

Отладка gdb { #fig:006 width=70% }

7. Запустила программу (рис. [-@fig:007])

Запуск программы { #fig:007 width=70% }

8. Постранично просмотрела исходный код (рис. [-@fig:008])

Постранично код { #fig:008 width=70% }

9. Просмотрела определённый диапазон строчек кода (рис. [-@fig:009])

Диапазон кода { #fig:009 width=70% }

10. Просмотрела определённый диапазон строчек кода неосновного файла (рис. [-@fig:0010])

Диапазон кода неосновного файла { #fig:0010 width=70% }

11. Установила точку останова в файле calculate.c на строке номер 21 (рис. [-@fig:0011])

Установка точки останова { #fig:0011 width=70% }

12. Вывела информацию об имеющихся в проекте точка останова (рис. [-@fig:0012])

Информация о точках останова { #fig:0012 width=70% }

13. Запустила программу внутри отладчика и убедилась, что программа остановится в момент прохождения точки останова: (рис. [-@fig:0013])

Программа с точкой останова { #fig:0013 width=70% }

14. Посмотрела, чему равно Numeral в разных моментах - они равны (рис. [-@fig:0014])

Numeral { #fig:0014 width=70% }

15. Посмотрела информацию о точке останова (рис. [-@fig:0015])

Информация о точке останова { #fig:0015 width=70% }

16. Удалила точку останова (рис. [-@fig:0016])

Удаление точки останова { #fig:0016 width=70% }

Выводы

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