

Front matter

lang: ru-RU title: Средства, применяемые при разработке программного обеспечения в ОС типа UNIX/Linux
author: Данилова Анастасия Сергеевна institute: RUDN University, Moscow, Russian Federation
date: 04.06.2021

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

Средства, применяемые при разработке программного обеспечения в ОС типа 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.

Компиляция исходного текста и построение исполняемого файла

Стандартным средством для компиляции программ в ОС типа UNIX является GCC (GNU Compiler Collection). Это набор компиляторов для разных языков программирования (С, С++, Java, Фортран и др.).

Работа с GCC производится при помощи программы gcc, определяет и осуществляет запуск нужного компилятора для входного файла. Файлы с расширением (суффиксом) .c воспринимаются gcc как программы на языке С, файлы с расширением .cc или .C — как файлы на языке С++, а файлы с расширением .o считаются объектными.

Таким образом, gcc по расширению (суффиксу) .c распознает тип файла для компиляции и формирует объектный модуль — файл с расширением .o. Если требуется получить исполняемый файл с определённым именем (например, hello), то требуется воспользоваться опцией -o и в качестве параметра задать имя создаваемого файла:
gcc -o hello main.c

Компиляция исходного текста и построение исполняемого файла

- с компиляция без компоновки — создаются объектные файлы file.o
- o file-name задать имя file-name создаваемому файлу
- g поместить в файл (объектный или исполняемый) отладочную информацию для отладчика gdb
- MM вывести зависимости от заголовочных файлов С и/или С++ программ в формате, подходящем для утилиты make; при этом объектные или исполняемые файлы не будут созданы
- Wall вывод на экран сообщений об ошибках, возникших во время компиляции

Make

Для сборки разрабатываемого приложения и собственно компиляции полезно воспользоваться командой make. Она позволяет автоматизировать процесс преобразования файлов программы из одной формы в другую, отслеживает взаимосвязи между файлами. Для работы с утилитой make необходимо в корне рабочего каталога с Вашим проектом создать файл с названием makefile или Makefile, в котором будут описаны правила обработки файлов Вашего программного комплекса.

Make

В самом простом случае Makefile имеет следующий синтаксис:

<цель_1> <цель_2> ... : <зависимость_1> <зависимость_2> ...

<команда 1> ... <команда n>

Сначала задаётся список целей, разделённых пробелами, за которым идёт двоеточие и список зависимостей. Затем в следующих строках указываются команды. Строки с командами обязательно должны начинаться с табуляции.

Тестирование и отладка

Во время работы над кодом программы программист неизбежно сталкивается с появлением ошибок в ней. Использование отладчика для поиска и устранения ошибок в программе существенно облегчает жизнь программиста. В комплект программ GNU для ОС типа UNIX входит отладчик GDB (GNU Debugger).

Для использования GDB необходимо скомпилировать анализируемый код программы таким образом, чтобы отладочная информация содержалась в результирующем бинарном файле. Для этого следует воспользоваться опцией -g компилятора gcc: gcc -c file.c -g

После этого для начала работы с gdb необходимо в командной строке ввести одноимённую команду, указав в качестве аргумента анализируемый бинарный файл: gdb file.o

Тестирование и отладка

backtrace - вывод на экран пути к текущей точке останова (по сути вывод названий всех функций) break - установить точку останова (в качестве параметра может быть указан номер строки или название функции) clear - удалить все точки останова в функции continue продолжить выполнение программы delete - удалить точку останова display - добавить выражение в список выражений, значения которых отображаются при достижении точки останова программы finish - выполнить программу до момента выхода из функции info breakpoints - вывести на экран список используемых точек останова list - вывести на экран исходный код (в качестве параметра может быть указано название файла и через двоеточие номера начальной и конечной строк) print - вывести значение указываемого в качестве параметра выражения run - запуск программы на выполнение

Анализ исходного текста программы

Ещё одним средством проверки исходных кодов программ, написанных на языке C, является утилита splint. Эта утилита анализирует программный код, проверяет корректность задания аргументов использованных в программе функций и типов возвращаемых значений, обнаруживает синтаксические и семантические ошибки.

Задания:

1. Создать файлы, в которых будут коды данные в теоретическом материале. 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.
2. С помощью gdb выполнить отладку программы calcul.
3. С помощью утилиты splint проанализируем коды файлов calculate.c и main.c.

Выводы

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.