

Задание №2 в рамках вычислительного практикума

Этапы получения исполняемого файла

Кострицкий А. С., Ломовской И. В.

Москва — 2022 — TS2203132017

Содержание

1	Цель работы	1
2	Задание	1
3	Формат защиты	2

1 Цель работы

Целью данной работы является изучение процесса получения исполняемого файла и организации объектных и исполняемых файлов.

2 Задание

Если для ответа на вопрос требуется выполнить какую-либо команду, предполагается, что в ответе будут приведены команда, часть её вывода, на которую Вы опираетесь при ответе, и Ваш ответ.

1. Напишите простую программу¹ на языке Си. В программе должны использоваться директива `include`, директива `define`. В отчёте приведите текст программы.
2. Изучите этапы получения исполняемого файла на примере этой программы. В отчёте приведите команду для выполнения каждого этапа, «выжимку» результатов выполнения этапа, краткое описание того, что происходит на этапе.
3. Почему `gcc` и `clang` мы называем «программами-драйверами»?
4. Найдите описание ключей `v` и `save-temps`. С помощью этих параметров изучите, какие этапы и с какими параметрами выполняются для получения исполняемого файла компилятором `gcc`.
 - (a) В чем отличие этих этапов от Ваших?
 - (b) Каково содержимое временных файлов, которые используются на этих этапах?
 - (c) Чем эти файлы отличаются от тех, которые получили Вы при выполнении второго пункта?

¹Разрешается взять любую уже написанную в рамках лабораторной по Си программу.

- (d) С какими объектными файлами и библиотеками происходит компоновка программы?
 - (e) Найдите, для чего нужны эти объектные файлы.
5. Какие этапы получения исполняемого файла выполняет компилятор clang?
 6. Найдите, с помощью какого ключа компилятора gcc можно передавать параметры компилятору с языка ассемблера. Используйте этот параметр для получения ассемблерного листинга Вашей программы в файл `main_asm.s`.
 7. Найдите, с помощью какого ключа компилятора gcc можно передавать параметры компоновщику. Используйте этот параметр для получения так называемого тар-файла для Вашей программы. Изучите его содержимое.
 8. Выполните дизассемблирование полученного объектного файла. Чем отличается результат дизассемблирования от полученной программы на языке ассемблера?
 9. Добавьте в Вашу программу глобальную проинициализированную переменную и глобальную непроинициализированную переменную. Получите объектный файл. Покажите таблицу символов и секции, из которых состоит объектный файл. В какие секции попали функции, глобальные переменные, локальные переменные?
 10. Добавьте отладочную информацию к объектному файлу. Что изменилось в объектном файле по сравнению с предыдущим пунктом?
 11. Получите исполняемый файл (без отладочной информации).
 12. Ответьте на следующие вопросы:
 - (a) Как отличаются объектный и исполняемый файлы с отладочной информацией и без по размеру?
 - (b) Как отличаются объектный и исполняемый файлы с отладочной информацией и без по количеству секций?
 - (c) Изменилось ли расположение функций, глобальных и локальных переменных?
 13. Какие динамические библиотеки использует Ваш исполняемый файл?

3 Формат защиты

Основная защита задания проходит на шестой неделе.

Студентом к дате основной защиты готовится страница отчёта в формате .docx (.odt), два файла .docx (.odt) и .pdf в виде zip-архива прикрепляются к кафедральному moodle.