Реферат на тему: Этапы обработки команды от исходного кода к исполняемому в операционных системах

Архитектура компьютеров и операционные системы

Ефремова Полина Александровна

Содержание

Список иллюстраций

Список таблиц

# 1 Цель работы

Ознакомиться с последовтельностью обработки программы от исходного кода к исполняемому файлу.

# 2 Введение

Процесс создания и исполнения программы включает несколько этапов, которые трансформируют исходный код в исполняемое приложение. Каждый из этих этапов требует выполнения специфических задач, в результате которых программа становится готовой к использованию в операционной системе. В данном реферате рассмотрены основные этапы обработки программы от исходного кода до исполнимого файла, а также роль операционной системы в этом процессе. Читателю будет описан процесс обработки команды на примере языка С/С++, так как их я изучаю на курсе программирования.

# 3 Основные понятия

**Исходный код** — текст программы, написанный на языке высокого уровня (например, C, Java), который не может быть выполнен напрямую процессором.

**Препроцессинг** — этап, на котором выполняются предварительные обработки исходного кода, такие как удаление комментариев и подключение библиотек.

**Компиляция** — преобразование исходного кода в объектный код (промежуточную форму), который понятен компьютеру, но еще не исполним.

**Ассемблирование** — преобразование объектного кода в машинный код, который процессор может выполнить.

**Линковка** — процесс объединения объектных файлов в один исполнимый файл, с добавлением необходимых библиотек.

**Исполняемый файл** — файл, готовый к выполнению в операционной системе, содержащий машинный код.

**Машинный код** — низкоуровневый код, который процессор может непосредственно выполнить.

**Промежуточный код** — объектный код, который требует дальнейшей обработки для преобразования в исполнимый файл.

# 4 До обработки программы

В первую очередь программист создает исходный код программы. Этот код написан на языке программирования высокого уровня, например, на C, C++, Python, Java и других. Исходный код состоит из текста, который описывает логику работы программы и выполняет конкретные задачи. Исходный код является удобным для человека и легко читаемым, однако операционные системы и процессоры не могут напрямую работать с таким кодом.

# 5 Основные этапы обработки программы (рис. 1).

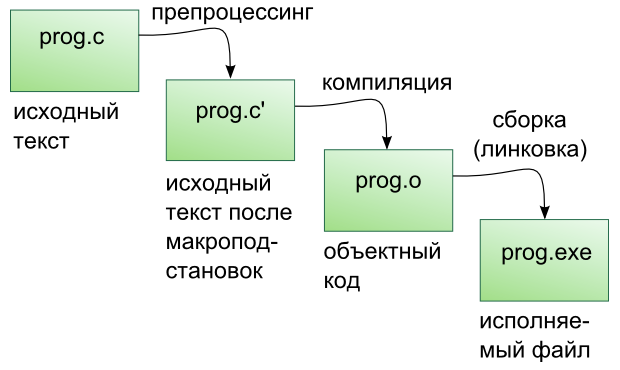


Рис. 1: Этапы обработки программы

## 5.1 Предварительная обработка (Preprocessing)

На этом этапе компилятор обрабатывает директивы препроцессора, такие как #include и #define. Препроцессор заменяет макросы, включает заголовочные файлы и выполняет другие текстовые замены. Например, если в нашем коде есть строка #include <stdio.h>, препроцессор заменит её содержимым файла stdio.h.

## 5.2 Компиляция (Compilation)

На этапе компиляции исходный код переводится в промежуточный код, называемый ассемблерным кодом. Этот код еще не является машинным, но уже ближе к нему. Компилятор анализирует синтаксис и семантику вашего кода, проверяя его на наличие ошибок и преобразуя его в ассемблерный код.

На этом этапе компилятор также может выполнять некоторые оптимизации, чтобы улучшить производительность конечного кода:

* **Лексический анализ.** Последовательность символов исходного файла преобразуется в последовательность лексем.
* **Синтаксический анализ.** Последовательность лексем преобразуется в дерево разбора.
* **Семантический анализ.** Дерево разбора обрабатывается с целью установления его семантики (смысла) — например, привязка идентификаторов к их декларациям, типам, проверка совместимости, определение типов выражений и т. д.
* **Оптимизация.** Выполняется удаление излишних конструкций и упрощение кода с сохранением его смысла.
* **Генерация кода.** Из промежуточного представления порождается объектный код.

## 5.3 Ассемблирование (Assembly)

Ассемблерный код преобразуется в объектный код (машинный код), который может быть выполнен процессором. Объектный код обычно хранится в файлах с расширением .o или .obj.

Объектный код — это программа на языке машинных кодов с частичным сохранением символьной информации, необходимой в процессе сборки.

При отладочной сборке возможно сохранение большого количества символьной информации (идентификаторов переменных, функций, а также типов).

## 5.4 Компоновка (Linking)

На этапе компоновки объектные файлы и библиотеки объединяются в один исполняемый файл. Компоновщик разрешает все внешние ссылки и создает окончательный исполняемый файл, готовый к запуску. Этот процесс включает в себя объединение кода из различных модулей и библиотек, а также разрешение всех символов и адресов.

## 5.5 Создание исполняемого файла

После завершения линковки программа преобразуется в исполняемый файл (например, .exe в Windows или без расширения в Unix-подобных системах). Этот файл содержит все необходимые данные, чтобы программа могла быть загружена и выполнена в операционной системе.

## 5.6 Загрузка программы в память

Перед запуском операционная система загружает исполняемый файл в память. Это включает в себя выделение памяти для программы, загрузку её инструкций и данных в оперативную память, а также настройку стека, кучи и других структур данных, необходимых для её работы.

## 5.7 Исполнение программы

Когда программа загружена в память, операционная система передает управление процессору, и начинается выполнение программы. В процессе выполнения могут происходить системные вызовы, взаимодействие с пользовательскими данными, доступ к файлам и ресурсам операционной системы.

## 5.8 Завершение работы программы

Когда программа завершает выполнение, операционная система освобождает ресурсы, которые она использовала, и завершает процесс. Это может включать закрытие открытых файлов, освобождение памяти и завершение всех потоков программы.

# 6 Заключение

Процесс обработки программы от исходного кода до исполняемого файла представляет собой сложный и многослойный процесс. Он включает в себя несколько ключевых этапов, таких как препроцессинг, компиляция, ассемблирование, линковка и загрузка в память. Каждый из этих этапов играет важную роль в создании работающего программного продукта, который может быть использован пользователями и взаимодействовать с операционной системой.

# Список литературы

1.[Основы компиляции программ на С](https://sky.pro/wiki/html/osnovy-kompilyacii-programm-na-c/)

2.[Этапы обработки программы](https://ozlib.com/1024750/tehnika/etapy_obrabotki_programm#13109)

3.[Этапы компиляции](https://foxford.ru/wiki/informatika/etapy-kompilyatsii?srsltid=AfmBOoqih9jwo1LiCzJ25BKmi-IW6HYaenRXotpGDQ2t4uMHBfgwtKU5&utm_referrer=https%3A%2F%2Fwww.google.com%2F)

4.[Материалы к лекции 3 на ТУИС](https://esystem.rudn.ru/mod/folder/view.php?id=1224409)