

Паневин Н. Б.

студент магистратуры

Тулского государственного педагогического университета им. Л.Н.

Толстого

ПОНЯТИЕ СИСТЕМЫ ПРЕПРОЦЕССИНГА И ЭТАПЫ ЕЕ ПРОЕКТИРОВАНИЯ

Анотация: В этой статье рассматривается концепция системы препроцессинга исходного кода и её роль в разработке программного обеспечения. Препроцессинг представляет собой процесс преобразования исходного кода программы, написанной на одном из языков программирования, в эквивалентную программу на другом языке, называемом целевым. Это достигается с помощью специального программного обеспечения, известного как транспайлер. Программные обеспечения, использующие систему препроцессинга для обработки целевого языка программирования, называются препроцессорами.

Ключевые слова: язык программирования, исходная программа, исходный код, лексический анализ, целевой язык, синтаксический анализ, оптимизация кода, препроцессор.

Panevin N. B.

Master's Degree

Tula State Pedagogical University named after L.N. Tolstoy

THE CONCEPT OF A PREPROCESSING SYSTEM AND THE STAGES OF ITS DESIGN

Abstract: This article discusses the concept of a source code preprocessing system and its role in software development. Preprocessing is the process of converting the source code of a program written in one of the programming languages into an equivalent program in another language, called the target

language. This is achieved with the help of special software known as a transpiler. Software that uses a preprocessing system to handle the target programming language are called preprocessors.

Keywords: *programming language, source program, source code, lexical analysis, target language, parsing, code optimization, preprocessor.*

Система препроцессинга исходного кода представляет собой программное обеспечение, с помощью которого можно выполнить транспилиацию, то есть преобразовать программу, написанную на одном языке программирования, называемым исходным, в эквивалентную программу на другом языке — целевом. Программа, преобразующая исходный код программы, написанной на одном языке программирования в качестве исходных данных, и производящий эквивалентный ему целевой код на другом языке программирования называется транспайлером.

Программы, использующие такую технологию, также называют препроцессором, которые возвращают программный код с дополнительными изменениями, направленными на улучшение читаемости или ускорения работы по его написанию.

Транслитируемым или исходным языком программирования считается язык программирования, в котором содержатся дополнительные операторы и который требует дополнительной обработки для их изменения. Исходный код на данном языке обрабатывается системой препроцессинга.

Целевым языком программирования считается такой язык программирования, который не будет содержать дополнительных операторов. Для преобразования в него используется система препроцессинга.

Систему препроцессинга используют для решения следующих задач:

1) Упрощение инструкций посредством создания пользовательских операций или полноценных выражений.

2) Добавление нового функционала, обыгранного либо в рамках компилируемого языка, либо с наличием готового преобразования в конечный язык программирования.

Конечной целью, для которой разрабатывается система препроцессинга, можно считать оптимизацию времени создания исходного кода посредством выполнения задач, перечисленных выше.

Результирующими данными в результате обработки исходного языка системой препроцессинга могут различаться между собой в зависимости от выбранного метода обработки. Так, например некоторые препроцессоры имеют возможность только выполнять элементарную текстовую подстановку, в то время как другие могут изменить восприятие и текст настолько сильно что смогут стать новым языком программирования.

В процессе обработки исходного языка системой препроцессинга используются следующие понятия:

1) Лексема — это символ или последовательность из нескольких символов исходной программы, которая соответствует шаблону токена и идентифицируется лексическим анализатором как экземпляр токена.

2) Шаблон — это описание вида, который может принимать лексема токена. В случае ключевого слова шаблон представляет собой последовательность символов, образующих это ключевое слово.

3) Токен — это пара, состоящая из идентификатора токена и его значения. Идентификатора токена — это абстрактный символ или последовательность из нескольких символов, представляющий тип лексической единицы.

Во время работы системы препроцессинга в ней могут присутствовать следующие этапы:

1) Лексический анализ – процесс, при котором исходный код разделяется на составные части и на каждую из них накладывается грамматическая структура. В последствии препроцессор использует эту

структуру для создания промежуточного представления исходной программы.

2) Синтаксический анализ – процесс, при котором анализатор использует первые компоненты токенов, полученных при лексическом анализе, для создания промежуточного представления, которое описывает грамматическую структуру потока токенов.

3) Транслитерация – процесс, при котором происходит непосредственное преобразование промежуточного представления исходного кода в эквивалентное промежуточное состояние целевого языка с обработкой операторов исходного языка. Процесс преобразования происходит с использованием таблицы правил, в которой описано, какие элементы или группы элементов промежуточного состояния должны быть преобразованы в эквивалентные элементы или группы элементов промежуточного состояния целевого языка.

4) Оптимизация кода – процесс, занимающийся обработкой промежуточного представления исходной программы с целью получения более качественного кода. На данном этапе могут быть убраны излишние преобразования, упрощены нагруженные конструкции, а также изменен порядок вычислений и замена аналогичными, но менее затратную инструкцию.

5) Генерация кода – процесс, при котором преобразует оптимизированное промежуточное представление исходного кода, полученного на этапе оптимизации кода, в целевой язык. Если целевой язык представляет собой машинный код, для каждой переменной, используемой программой, выбираются соответствующие регистры или ячейки памяти после чего промежуточные команды транслируются в последовательности машинных команд, выполняющих те же действия.

Список используемых источников:

1. Препроцессор [Электронный ресурс]. – Режим доступа: <https://learn.microsoft.com/ru-ru/cpp/preprocessor/preprocessor?view=msvc-170>. – Дата доступа: 26.06.2024
2. Препроцессор [Электронный ресурс]. – Режим доступа: <https://math.spbu.ru/user/dlebedin/preproc.pdf>. – Дата доступа: 26.06.2024
3. CSS-препроцессоры в 2024. Большой обзор [Электронный ресурс]. Режим доступа: <https://htmlacademy.ru/blog/css/css-preprocessor>. – Дата доступа: 26.06.2024
4. Препроцессоры [Электронный ресурс]. – Режим доступа: https://mrmlnc.gitbooks.io/less-guidebook-for-beginners/content/chapter_1/preprocessors.html. – Дата доступа: 26.06.2024
5. Компилятор [Электронный ресурс]. – Режим доступа: <https://bigenc.ru/c/kompiliator-6d89ee>. – Дата доступа: 26.06.2024
6. ЧТО ТАКОЕ КОМПИЛЯТОР? КАК РАБОТАЕТ: ВИДЫ И ПРИМЕРЫ [Электронный ресурс]. – Режим доступа: <https://digitalocean.ru/n/chto-takoe-kompilyator>. – Дата доступа: 26.06.2024
7. Препроцессор Pug(Jade) [Электронный ресурс]. – Режим доступа: <https://gist.github.com/neretin-trike/53aff5afb76153f050c958b82abd9228/>. – Дата доступа: 26.06.2024
8. Что такое компилятор [Электронный ресурс]. – Режим доступа: <https://elbrusboot.camp/blog/chto-takoie-kompiliator/>. – Дата доступа: 26.06.2024