



Санкт-Петербургский
государственный
университет

Вопросы

- 1 Зачем в C++ (C) код модулей разделяют на *.hpp (*.h) и *.cpp (*.c)?
- 2 Как проходит процесс компиляции CPP файлов в бинарный файл?
- 3 Что такое препроцессор и как он работает?
- 4 Что такое директивы (команды) препроцессора и какие знаете?
- 5 Как работает директива include?
- 6 Что такое макрос, какие макросы бывают и как работают?
- 7 Как работает директива define?
- 8 Как защитить хедер от повторного включения?
- 9 Что именно линкует линкер?



Санкт-Петербургский
государственный
университет

ОТВЕТЫ

1 ЗАЧЕМ В C++ (C) КОД МОДУЛЕЙ РАЗДЕЛЯЮТ НА *.hpp (*.h) И *.cpp (*.c)?

Файлы с расширением *.hpp (*.h) – файлы заголовков – используются для объявления (declaration) функций, классов и других сущностей. А файлы *.cpp (*.c) используются для определения (definition). Нельзя использовать функцию (любую сущность) без и до её объявления в файле. Это вызовет ошибку на этапе компиляции (при анализе кода перед переводом его на машинный язык). Если функция будет объявлена, но не будет определена (.cpp файл), то это вызовет ошибку на этапе линковки.

Функцию можно объявить 3 способами:

1. В КАЖДОМ ФАЙЛЕ, ГДЕ ИСПОЛЬЗУЕТСЯ

```
1 int func();
2
3 int main() {
4     std::cout << func();
5 }
```

2. В ФАЙЛЕ ОПРЕДЕЛЕНИЯ (*.cpp / *.c)

```
1 #include "func.cpp"
2
3 int main() {
4     std::cout << func();
5 }
```

3. В ФАЙЛЕ ОБЪЯВЛЕНИЯ (*.hpp / *.h)

```
1 #include "func.hpp"
2
3 int main() {
4     std::cout << func();
5 }
```

В 1-ом случае, если поменяется что-то в сигнатуре функции, то это изменение придётся вносить во все файлы, где функция используется.

Во 2-ом случае при изменении сигнатуры функции изменение в объявлении надо сделать только один раз в cpp, но директива #include вставит много ненужного текста.

В 3-ем случае оба момента из 1-го и 2-го вариантов будут оптимальными для разработчика и компилятора. Поэтому и были созданы заголовочные файлы *.hpp (*.h) – только для объявления сущностей.



Санкт-Петербургский
государственный
университет

ОТВЕТЫ

2 Как проходит процесс компиляции сpp файлов в бинарный файл?

Процесс компиляции сpp файлов проходит в 3 этапа:

1. Препроцессинг
2. Трансляция
3. Линковка

Во время препроцессинга в исходном файле происходит удаление комментариев, замена макросов на соответствующие тексты, включение необходимого текста библиотек и своих хедеров. При этом, нового файла не создается, а происходит преобразование существующего сpp.

Во время трансляции происходит лексический анализ кода и затем перевод программы с языка С++ на машинный язык для каждого из сpp файлов. В результате для каждого сpp файла создается свой obj файл. В obj файлах отсутствует связь с другими модулями, реализация библиотечных функций, код запуска всей программы.

Во время линковки связываются все obj-ые файлы в единый исполняемый файл, добавляет код запуска для конкретной ОС, и добавляет реализации необходимых библиотечных функций.

3 Что такое препроцессор и как он работает?

Препроцессор — это программа, принимающая данные на входе и выдающая данные для другой программы. В С и С++ препроцессор — это часть компилятора, которая преобразует исходный код перед его трансляцией в машинный код. Препроцессор в исходном файле производит удаление комментариев, замену макросов на соответствующие тексты, включение необходимого текста библиотек и своих хедеров. При этом, нового файла не создается, а происходит преобразование существующего сpp.



Санкт-Петербургский
государственный
университет

ОТВЕТЫ

4 ЧТО ТАКОЕ ДИРЕКТИВЫ (КОМАНДЫ) ПРЕПРОЦЕССОРА И КАКИЕ ЗНАЕТЕ?

ДИРЕКТИВЫ ПРЕПРОЦЕССОРА — ЭТО КОМАНДЫ, КОТОРЫЕ ИСПОЛЬЗУЮТСЯ ДЛЯ УПРОЩЕНИЯ ИЗМЕНЕНИЯ ТЕКСТА ИСХОДНЫХ ПРОГРАММ.

`#define`, `#include`, `#ifndef`, `#undef`, `#pragma` и т. д.

5 КАК РАБОТАЕТ ДИРЕКТИВА `include`?

УКАЗЫВАЕТ ПРЕПРОЦЕССОРУ ВКЛЮЧИТЬ СОДЕРЖИМОЕ УКАЗАННОГО ФАЙЛА В ТОЧКУ, ГДЕ ПОЯВЛЯЕТСЯ ДИРЕКТИВА. ЕСТЬ ДВА ВИДА: `#include "my_file.hpp"` — для включения своих файлов, `#include <out_lib>` — для включения внешних библиотек.

6 ЧТО ТАКОЕ МАКРОС, КАКИЕ МАКРОСЫ БЫВАЮТ И КАК РАБОТАЮТ?

МАКРОС — ЭТО ИДЕНТИФИКАТОР, КОТОРЫЙ ПРЕДСТАВЛЯЕТ КОНСТАНТУ ИЛИ ВЫРАЖЕНИЕ. ОН СОЗДАЁТСЯ ДИРЕКТИВОЙ ПРЕПРОЦЕССОРА `#define`. МАКРОСЫ ОБРАБАТЫВАЕТ ПРЕПРОЦЕССОР, КОТОРЫЙ ВМЕСТО ИМЕНИ МАКРОСА В КОДЕ ПОДСТАВЛЯЕТ ТЕЛО МАКРОСА.

7 КАК РАБОТАЕТ ДИРЕКТИВА `define`?

ДИРЕКТИВА `define` СОЗДАЕТ МАКРОС, КОТОРЫЙ МОЖЕТ ИСПОЛЬЗОВАТЬСЯ ДЛЯ ПРЕДСТАВЛЕНИЯ КОНСТАНТЫ ИЛИ ВЫРАЖЕНИЯ. ПРЕПРОЦЕССОР ПОДСТАВЛЯЕТ ТЕЛО МАКРОСА В МЕСТО КАЖДОГО ЕГО ОБНАРУЖЕНИЯ В КОДЕ.

С ПОМОЩЬЮ МАКРОСОВ МОЖНО СОЗДАВАТЬ НЕ ТОЛЬКО ИЗЯЩНЫЙ КОД, НО И ПЛОДИТЬ НЕ МЕНЕЕ ИЗЯЩНЫЕ БАГИ.





Санкт-Петербургский
государственный
университет

ОТВЕТЫ

8 Как защитить хедер от повторного включения?

Защитить .hpp файл от повторного включения можно либо директивой препроцессора `#pragma once`, либо конструкцией `#ifndef`, `#define`, `#endif`. Директива `#pragma once` может быть не реализована в некоторых компиляторах. Но если она реализована, то лучше использовать её, т. к. в таком случае глобальное пространство имён будет меньше засоряться.

9 Что именно линкует линкер?

Линкер связывает все `obj`-ые файлы в единый исполняемый файл, добавляет код запуска для конкретной ОС, и реализации необходимых библиотечных функций.