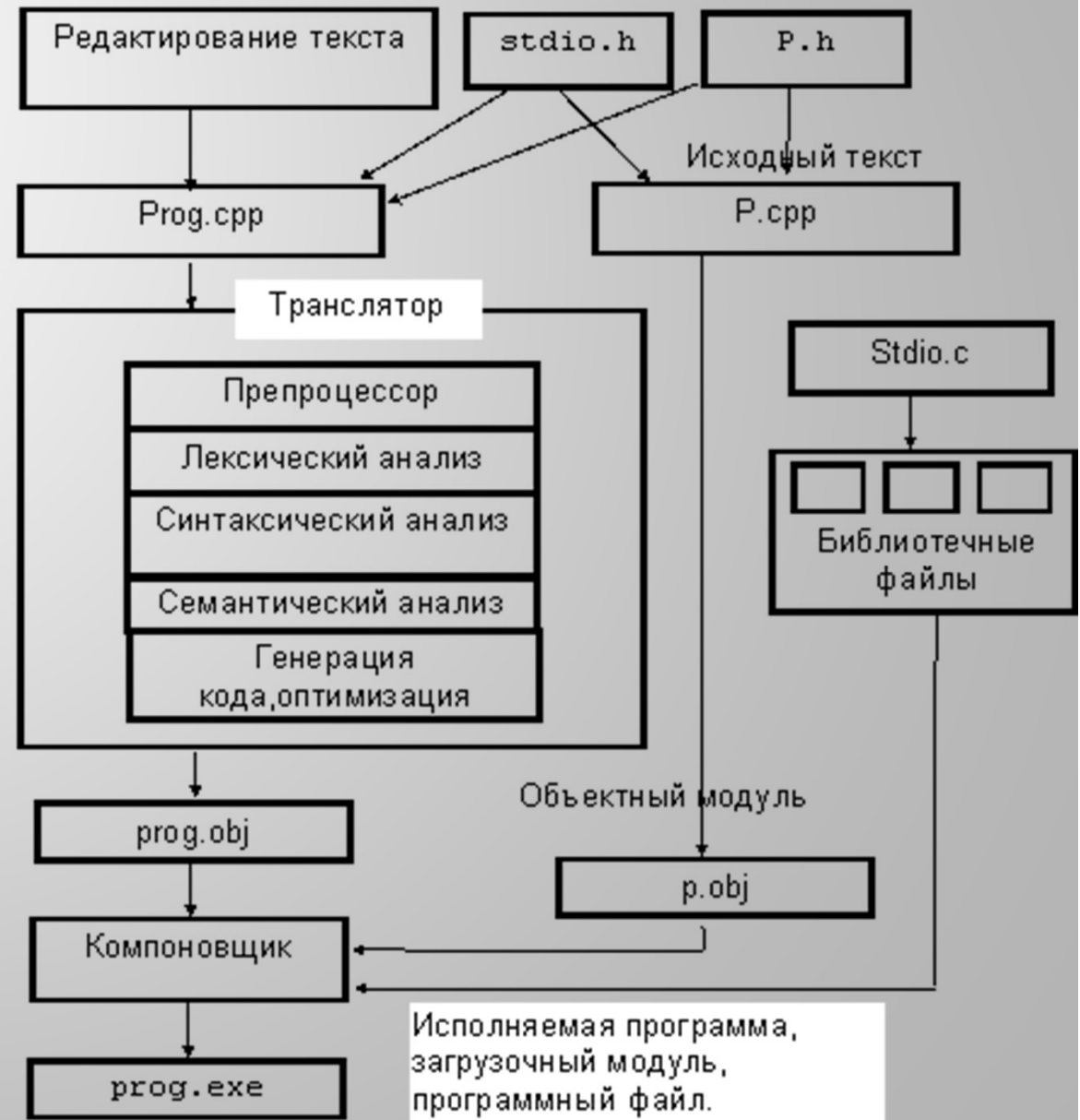


**БИБЛИОТЕКИ
ДИНАМИЧЕСКОЙ
КОМПОНОВКИ.
DLL**

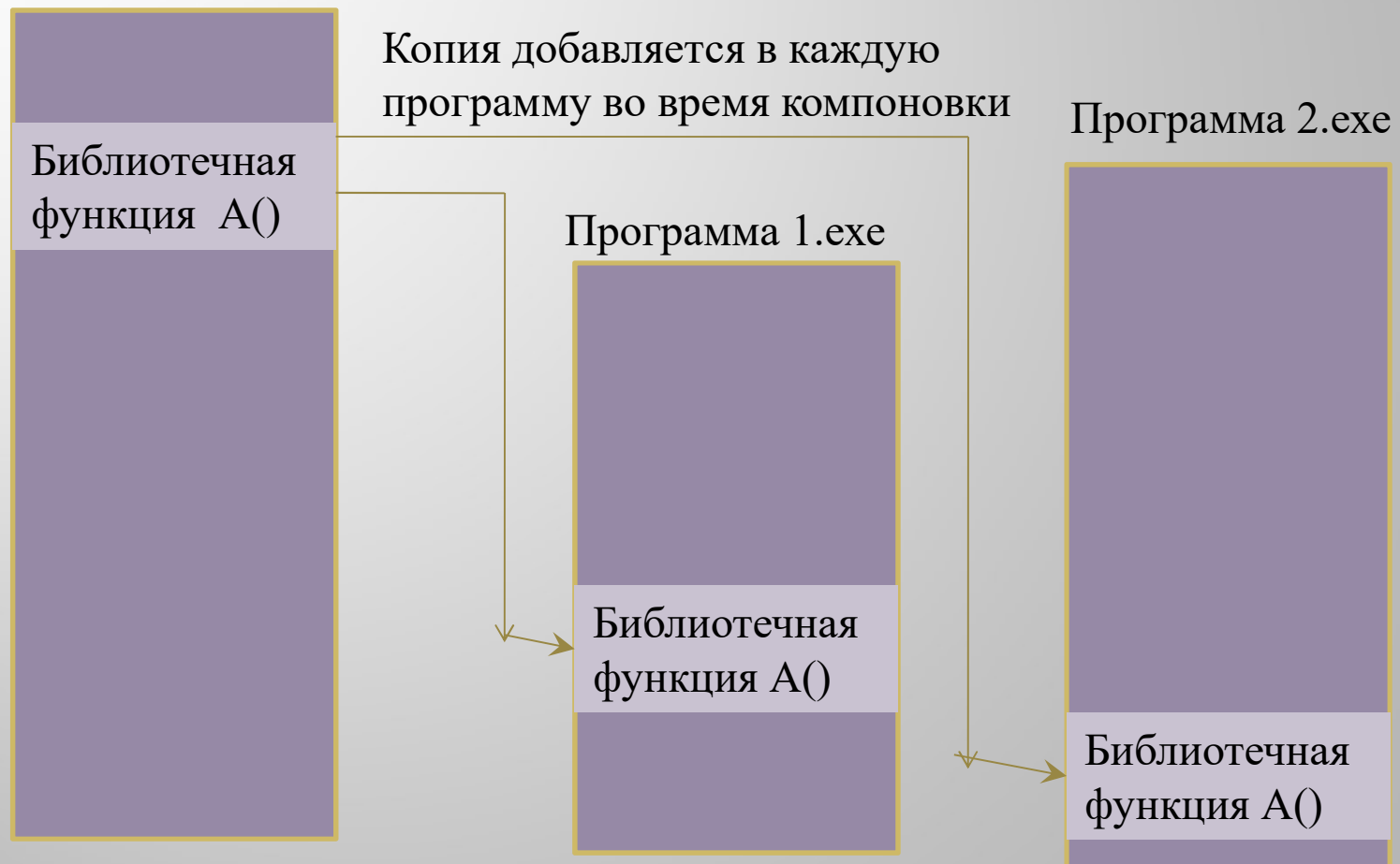
Создание исполняемого файла

- Подготовка текста приложения на языке программирования.
- Трансляция в .obj файл
- Компоновка .exe файла

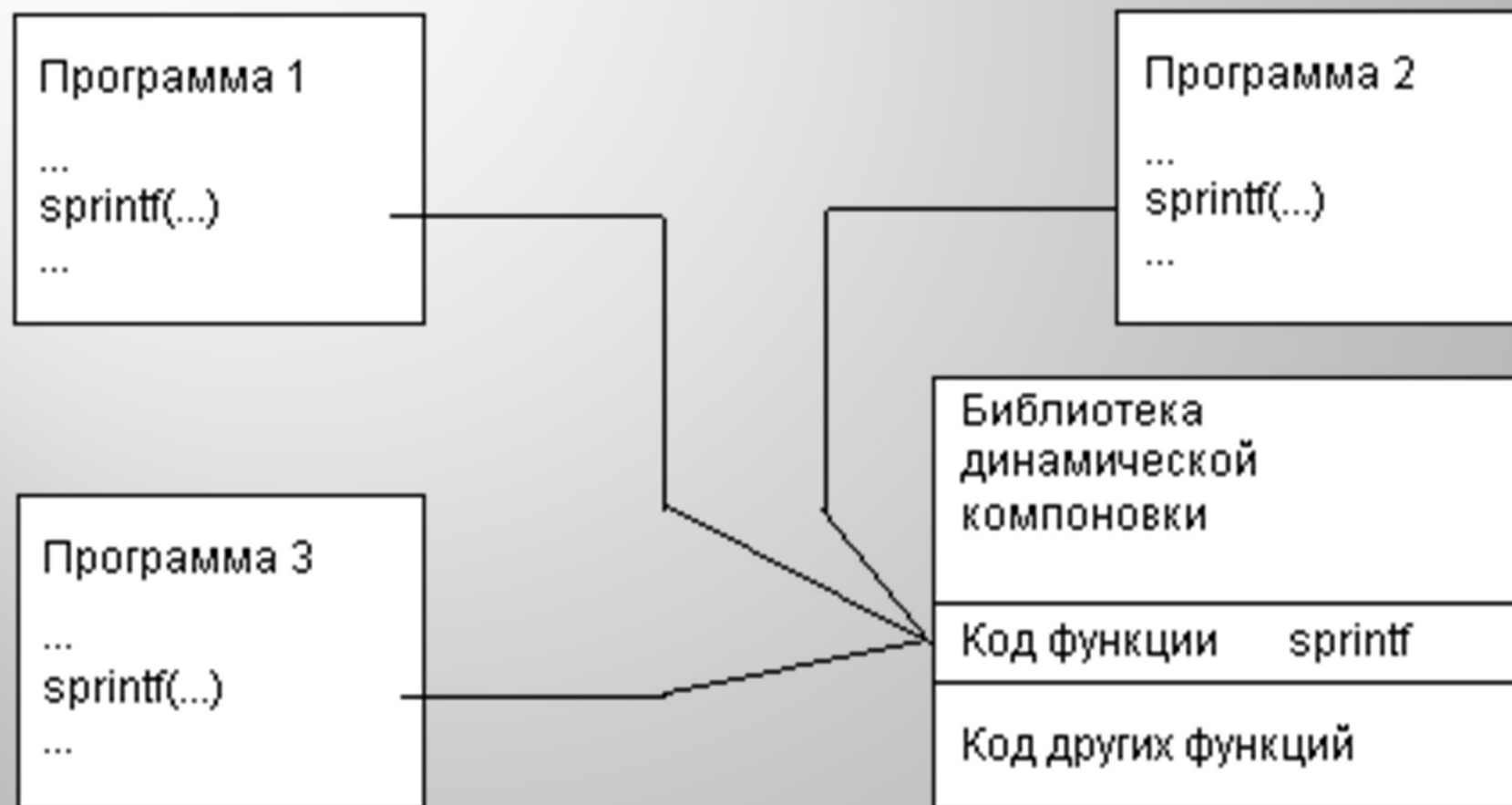


Статическая компоновка

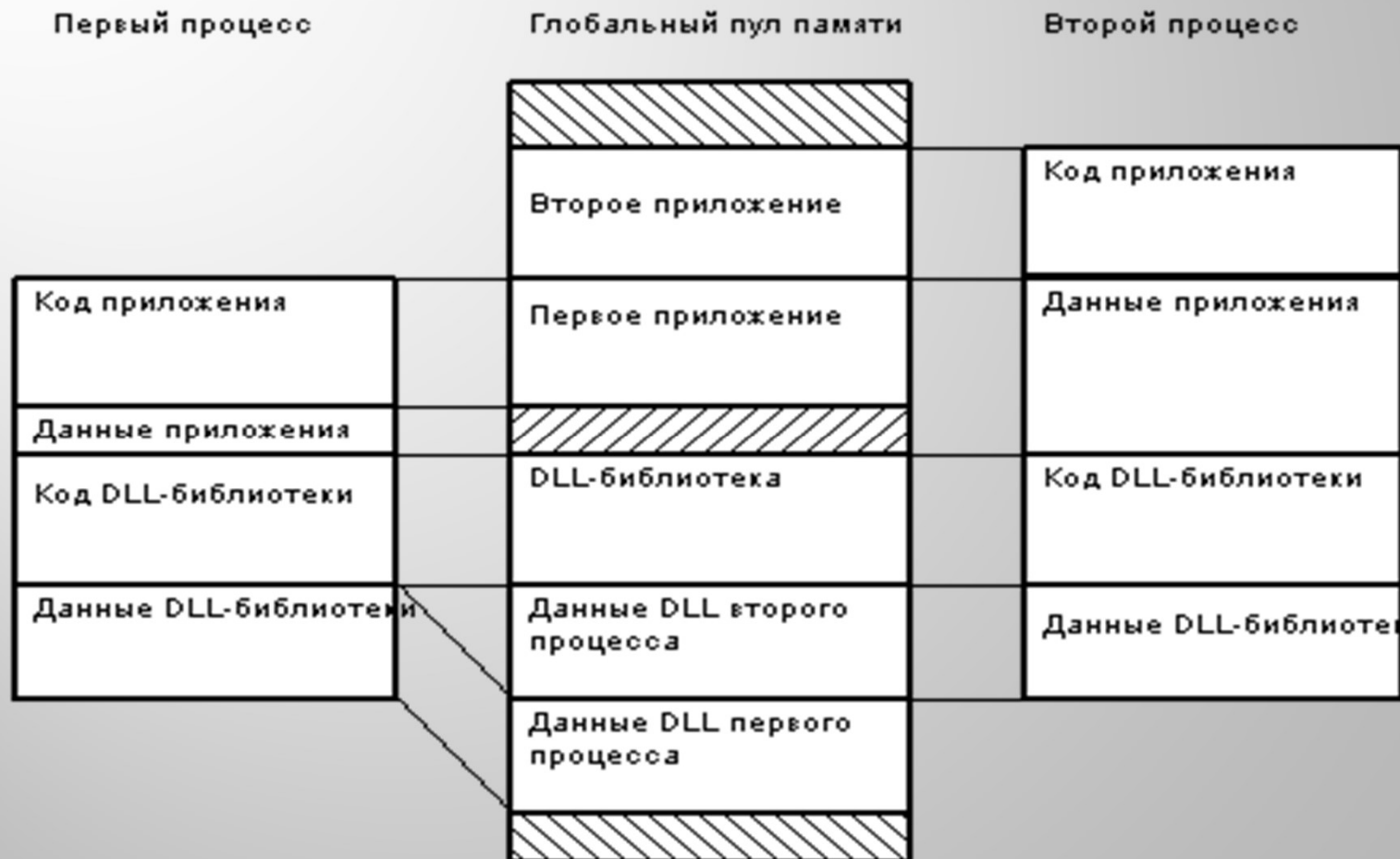
Статическая библиотека



Динамическая компоновка



Отображение DLL в адресном пространстве процессов



Структура приложения

Процесс 1

Первая копия приложения

Данные
Стек
Очередь сообщений

Процесс 2

Вторая копия приложения

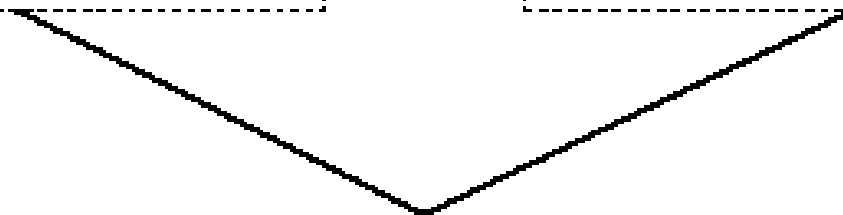
Данные
Стек
Очередь сообщений

DLL-библиотека

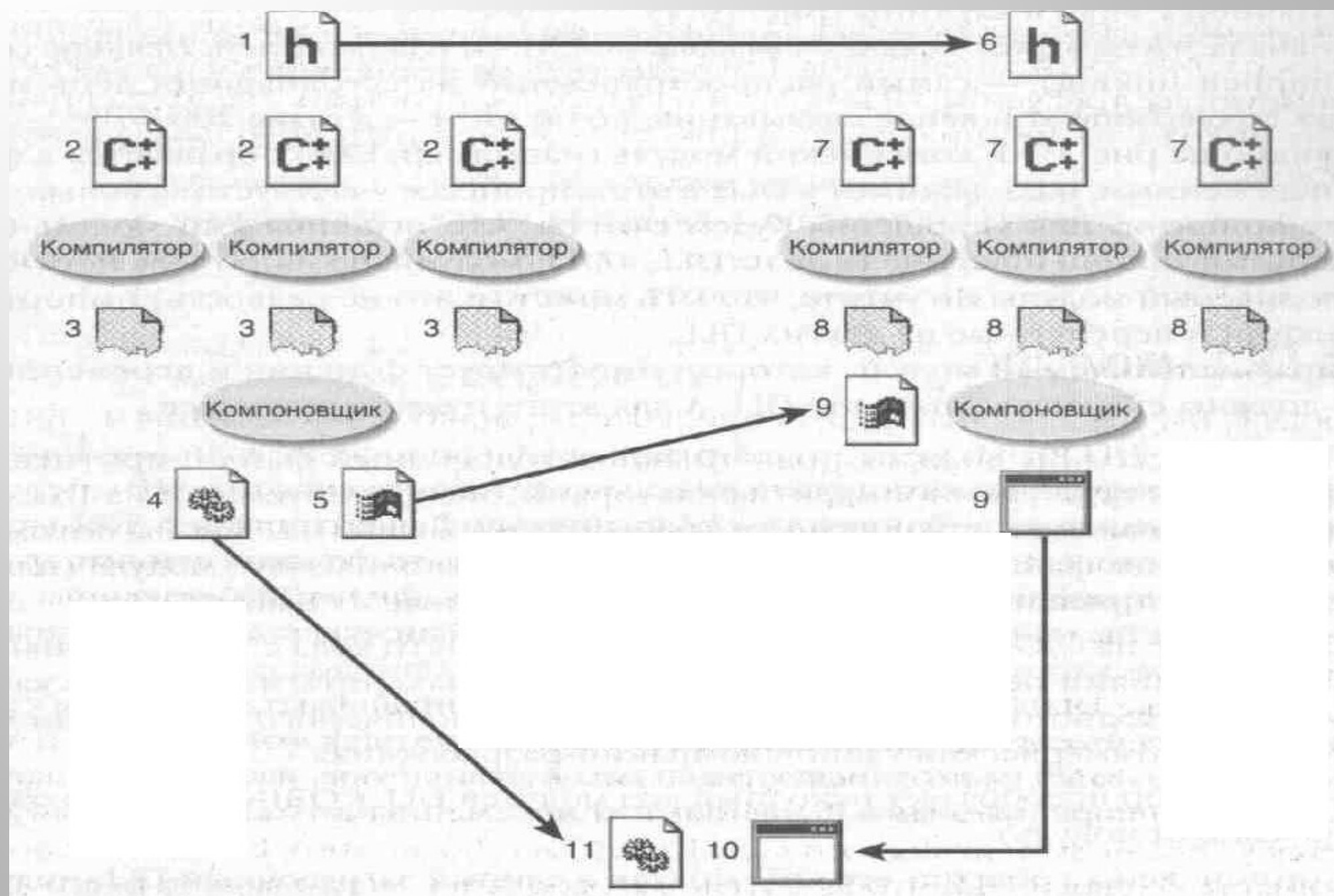
Код
Ресурсы
Данные

Модуль приложения

Код
Ресурсы



Неявное связывание



Порядок неявного связывания

1. В заголовочном файле определяются экспортируемые функции
MyDLL.h

```
EXPORT extern "C" __declspec (dllexport) int CALLBACK  
MyFunc (char *str);
```

2. Раздел экспорта

ordinal	hint	RVA	name
1	0	0001A3C6	Add
2	1	0001A367	MyFunc

3. При создании исходного EXE – модуля необходимо определить внешнюю функцию

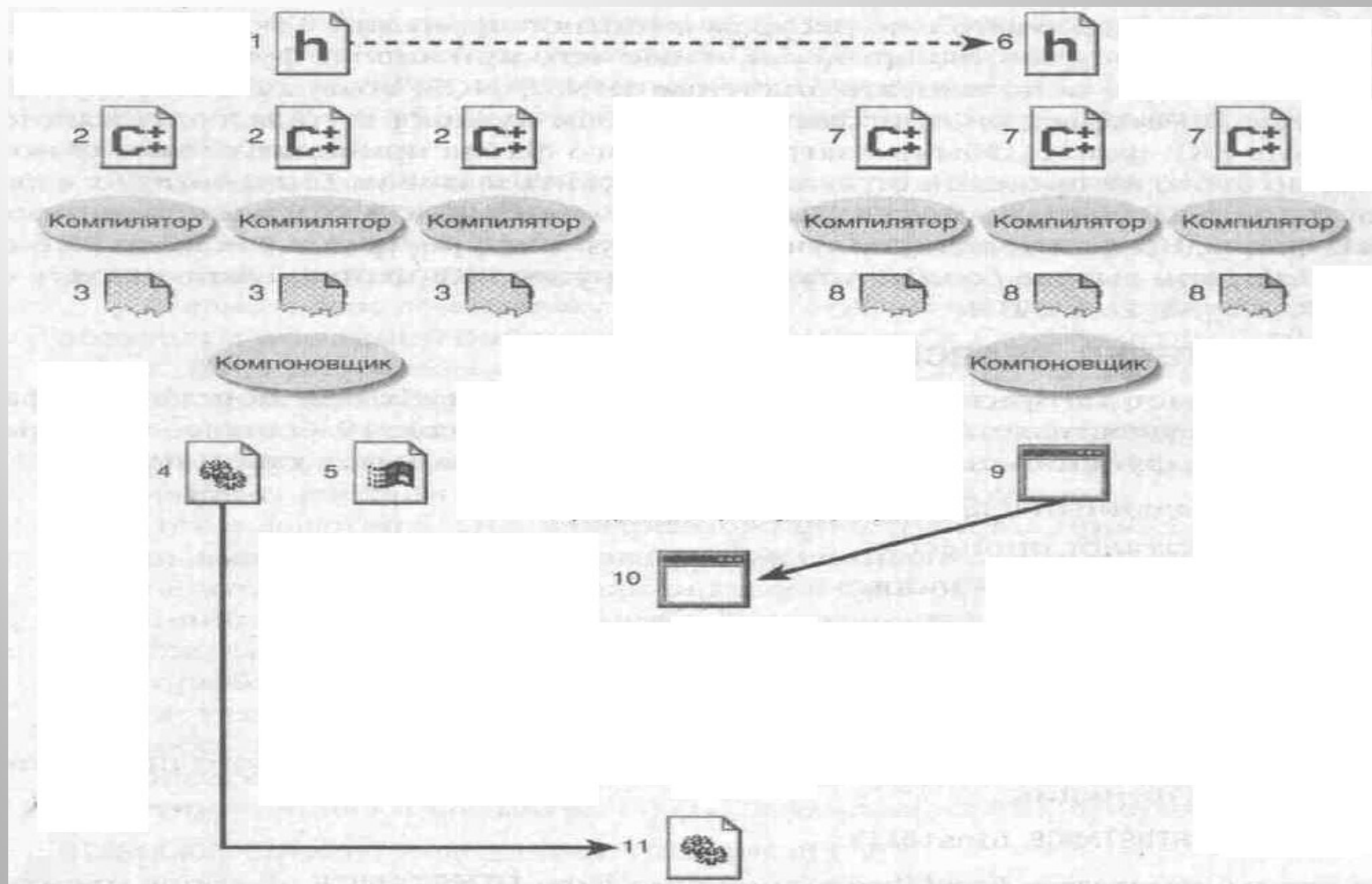
```
#include "MyDLL.h"  
extern "C" __declspec(dllimport) MyFunc(char * str);
```

4. Компоновщик создает в конечном EXE-модуле *раздел импорта*

5. Прямой вызов функции

```
int iCode=MyFunc("Hello");
```


Явное связывание



Пример динамической загрузки

```
typedef HWND (WINAPI *MYDLLPROC) (LPSTR);
MYDLLPROC  GetWindow;
FARPROC  MyPool;
HANDLE    hDLL;
hDLL = LoadLibrary("DLLDEMO.DLL");
    if(hDLL != NULL)

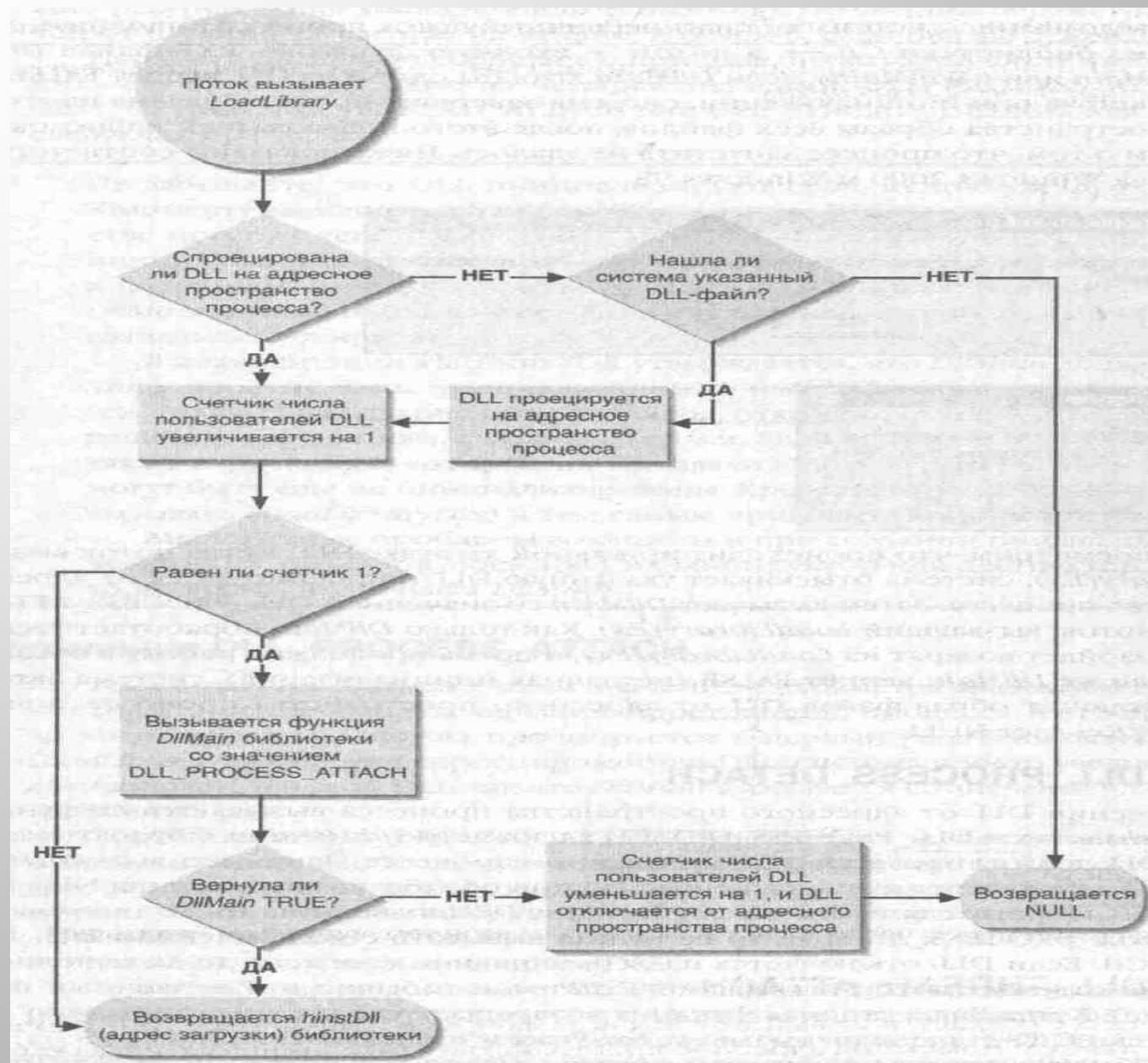
{
    MyPool = GetProcAddress(hLib, MAKEINTRESOURCE(8));
    GetWindow = (MYDLLPROC) GetProcAddress(hDLL,
                                           "MyFunc");
    if(GetWindow!= NULL)
    {
        int Code=(* GetWindow)("Hello");// вызываем функцию
    }
    FreeLibrary(hDLL);
}
```

Основная функция DLL

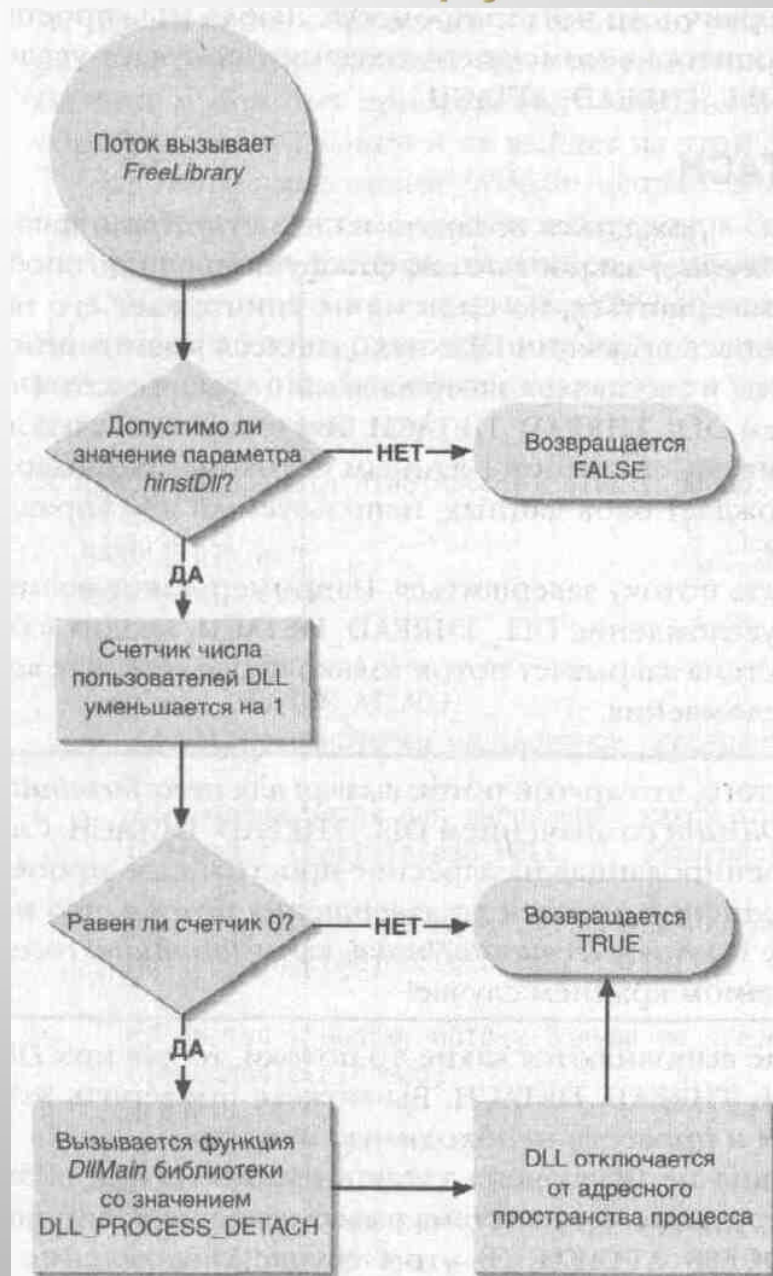
```
BOOL WINAPI DllMain( HINSTANCE hinstDLL,  
                    DWORD   fdwReason,  
                    LPVOID   lpvReserved);
```

Значение fdwReason	Описание
DLL_PROCESS_ATTACH	Запуск процесса или вызов функции LoadLibrary
DLL_THREAD_ATTACH	Создана новая задача
DLL_THREAD_DETACH	Задача завершает свою работу не аварийным способом
DLL_PROCESS_DETACH	Нормальное завершение процесса или вызова функции FreeLibrary

Вызов потоком функции LoadLibrary



Вызов потоком функции FreeLibrary



Файл определения .DEF

Формат описания экспортируемой функции:

EXPORTS

ИмяТочкиВхода [=ВнутрИмя] [@Номер] [NONAME]
[CONSTANT]

ПРИМЕР:

LIBRARY DLLNAME

DESCRIPTION 'DLL-библиотека DLLNAME'

EXPORTS

DrawBitmap=MyDraw @4

ShowAll

MyPoolPtr @5 CONSTANT

GetMyPool @8 NONAME

MyFunc @9