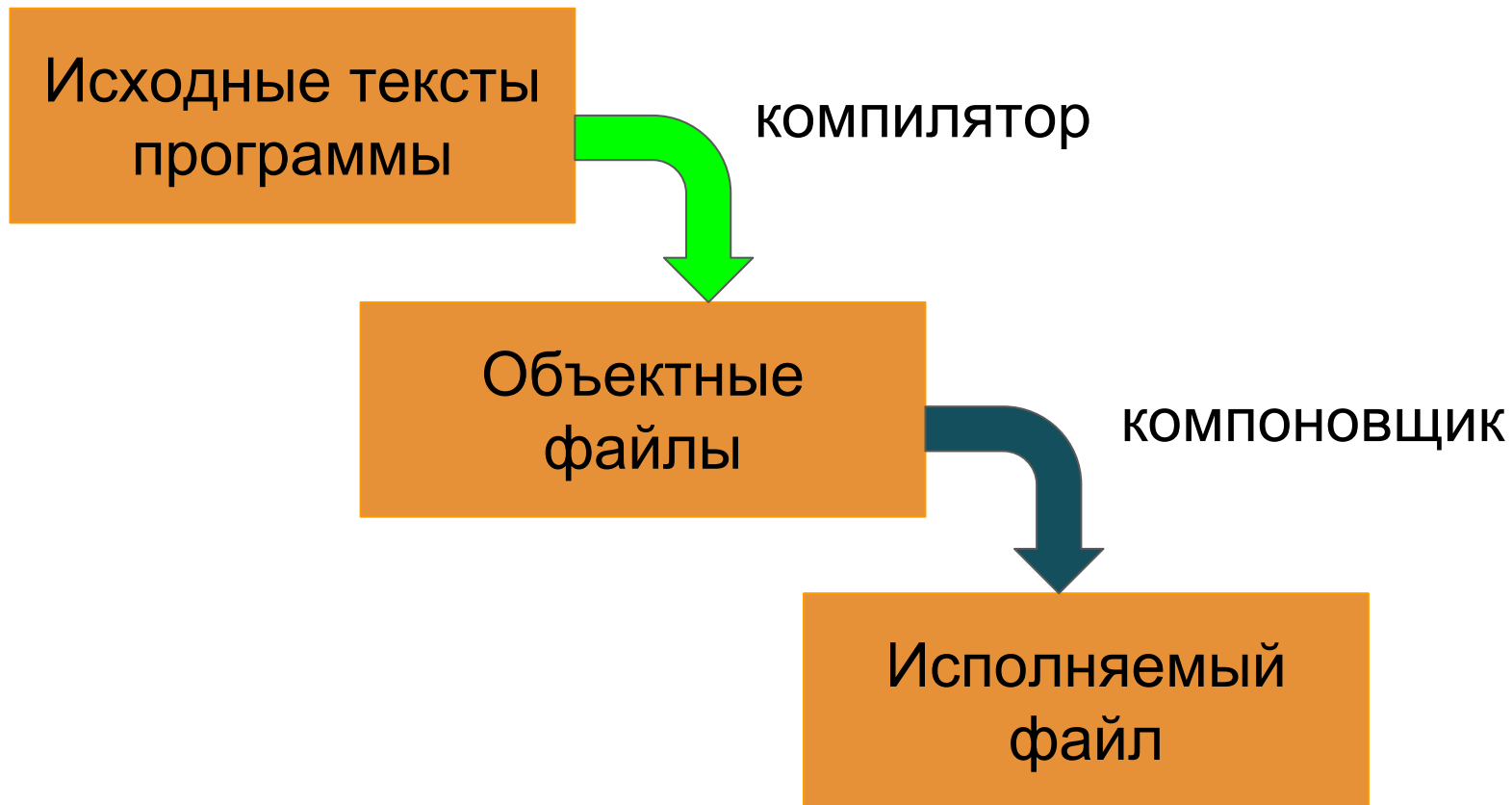


Разработка ПО в ОС Linux

Разработка приложений



Компилятор

- gcc - gnu compiler collection - компилятор языка C
- g++ - компилятор языка C++
- синтаксическая проверка, выдача ошибок компиляции
- `$gcc -o main main.c` => исполняемый файл `main`
- `$gcc -c main.c` => объектный файл `main.o`

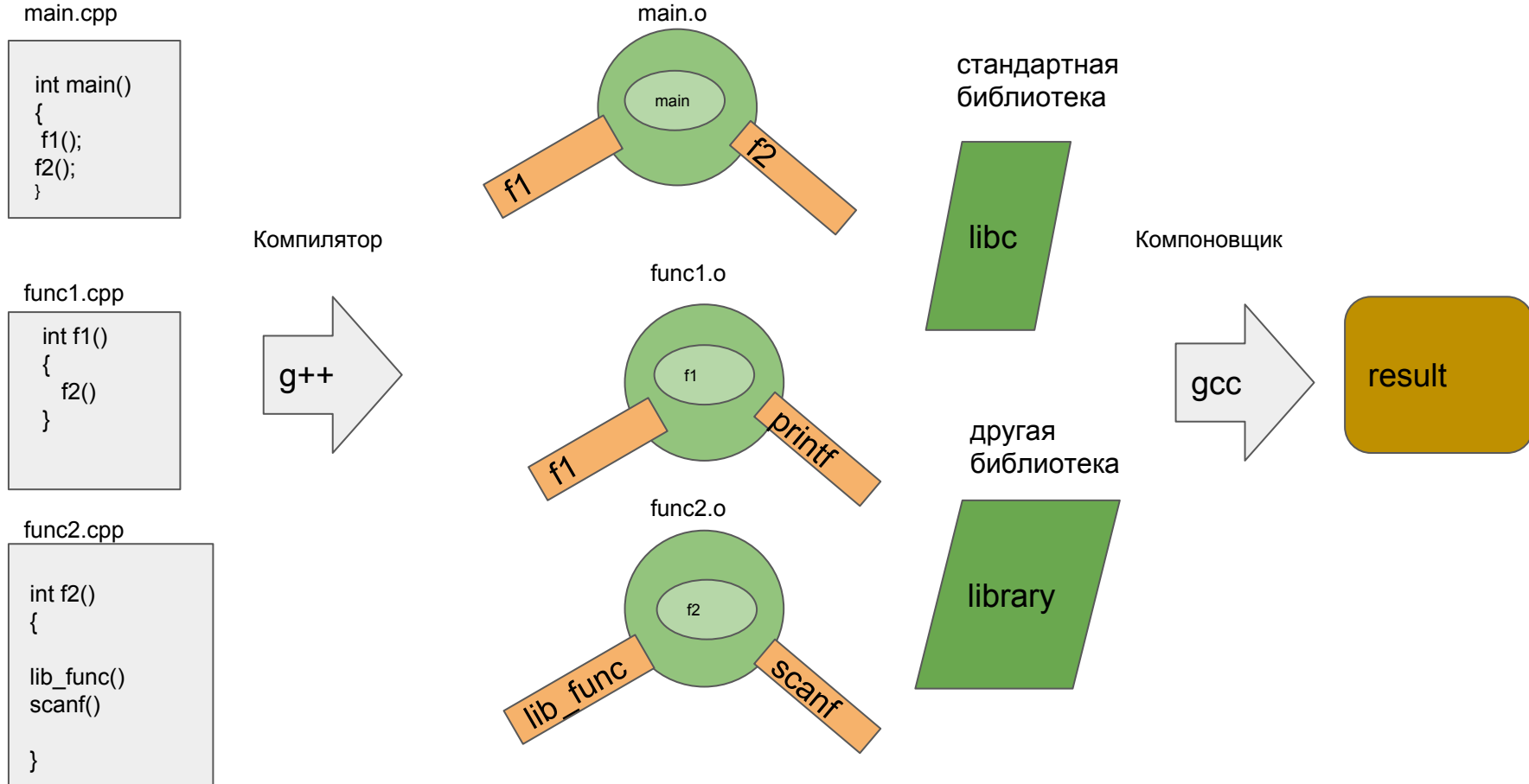
Компоновщик (линковщик)

- линковка определение связей между модулями (библиотеками), построение исполняемого файла
 - `$gcc -c main.c` \Rightarrow `main.o`
 - `$gcc -c func.c` \Rightarrow `func.o`
 - `$gcc -c func2.c` \Rightarrow `func2.o`
 - `$gcc main.o func.o func2.o -o result` \Rightarrow `result`
- ошибки линковки (*unresolved reference*)

Использование библиотек

- динамическая библиотека
 - `gcc -c -fpic func.c`
 - `gcc -c -fpic func2.c`
 - `gcc -shared -o libfoo.so func.o func2.o`
- `$gcc main.o -I/path/to/headers -L/path/to/library -lfoo -o result`
- запуск приложения, использующего библиотеку
 - `export LD_LIBRARY_PATH=/path/to/library:$LD_LIBRARY_PATH`
- стандартная библиотека

Процесс линковки



Автоматизация сборки

- Makefile
- Autotools
- CMake
- Среда разработки

Makefile

- `make main.c`
- Использование: `make`, `make clean`
- текстовый файл
- умная сборка - собираются только обновленные файлы
- директивы
 - `target`: dependencies
system command(s)

Makefile

all: result

result: main.o func1.o func2.o

g++ main.o func1.o func2.o -o result -L/path/To/Lib -llibrary

main.o: main.cpp

g++ -c main.cpp -I/path/toheaders

func1.o: func1.cpp

g++ -c func1.cpp -I/path/toheaders

func2.o: func2.cpp

g++ -c func2.cpp -I/path/toheaders

clean:

rm -rf *.o result

cmake

- Макроязык конфигурирования сборки
- Поиск библиотек, установленных в системе
- Генерация Makefile
- сборка отдельно от исходных файлов
- Генерация проектных файлов для различных сред разработки

CMake (CMakeLists.txt)

```
cmake_minimum_required(VERSION 2.8)
```

```
project(hello_world)
```

```
find_package(library REQUIRED)
```

```
include_directories(${library_INCLUDE_DIR})
```

```
add_executable(result main.cpp func.cpp)
```

```
add_library(func func1.cpp func2.cpp)
```

```
target_link_library(result func $(library_LIBRARY))
```

Управление проектами

- Среда разработки
 - Настройка проекта
 - Управление зависимостями
 - Отображение ошибок
 - Отладка программ
- Eclipse
- QtCreator