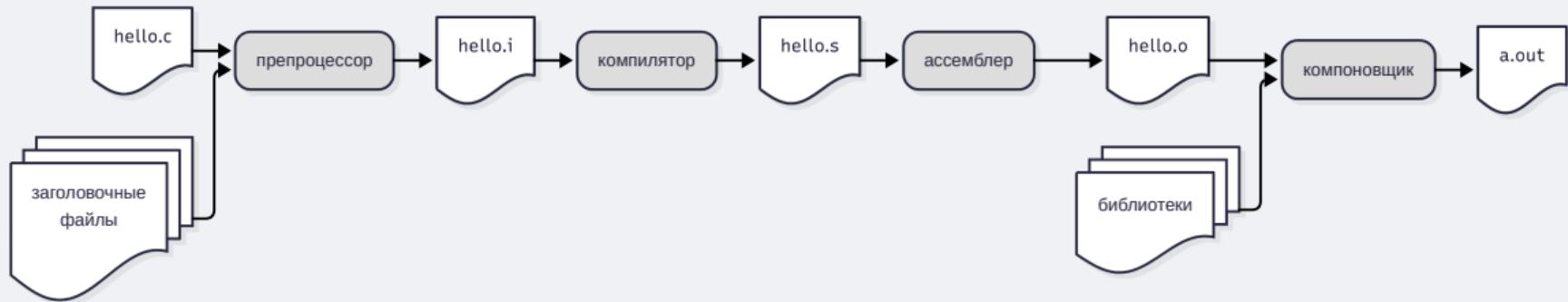


# Процесс компиляции



Санкт-Петербургский  
государственный университет

# Процесс компиляции



# Компилятор

hello.c

```
#include <stdio.h>

#define HELLO_WORLD "Hello, world!\n"

int main(int argc, char *argv[]) {
    printf("%s", HELLO_WORLD);
    return 0;
}
```

```
$ gcc -v hello.c -o hello
```

# Компилятор

```
$ gcc -v hello.c -o hello
...
gcc version 15.2.1 20250813 (GCC)
(1) /usr/lib/gcc/x86_64-pc-linux-gnu/15.2.1/cc1 -quiet -v hello.c -quiet
    ↳ -dumpbase hello.c -dumpbase-ext .c -mtune=generic -march=x86-64 -version -o
    ↳ /tmp/ccLNELT3.s
GNU C23 (GCC) version 15.2.1 20250813 (x86_64-pc-linux-gnu)
(2) as -v --64 -o /tmp/cc8JjKnv.o /tmp/ccLNELT3.s
(3) /usr/lib/gcc/x86_64-pc-linux-gnu/15.2.1/collect2 ...
```

# Первые выводы

GNU C Compiler → GNU Compiler Collection

Современный компилятор — набор инструментов (toolchain)

# Компилятор

```
$ gcc -fprofile-generate -o hello hello.c
```

# Формы существования кода (на Си)

- Исходный код
- Ассемблер
- Машинный код
- Исполняемый код

# Компилятор

```
$ gcc hello.c -o hello
$ gcc -E hello.c -o hello.i
$ gcc -S hello.c -o hello.s
$ gcc -c hello.c -o hello.o
```

# Утилиты для просмотра

```
$ objdump -d hello  
$ hexdump -C hello
```

## Функция main

```
int main() {}
```

```
int main(int argc, char *argv[]) {}
```

Запускаем ./hello

Что будет в argc и argv?

# Функция main

`int main() {}`

`int main(int argc, char *argv[]) {}`

Запускаем `./hello`

Что будет в argc и argv?

`argv[0]` — имя программы

`echo $?` — получить последний код возврата

Что будет, если `main` нет?

# Линкер

## ld — The GNU Linker

- Он же линковщик
- Он же компоновщик
- Он же редактор связей

Передача параметров линкеру при вызове gcc

```
$ gcc -Wl,-v hello.c -o hello
```

# Объявления и определения

Объявление (declaration) — указание имени функции, возвращаемого значения и принимаемых аргументов

```
int distance(int a, int b);
```

Определение (definition) — выполняемая последовательность команд (тело функции)

```
int distance(int a, int b)
{
    return abs(b - a);
}
```

# Домашнее задание

- 1 Написать программу проверки баланса скобок в исходной строке (т.е. число открывающих скобок равно числу закрывающих и выполняется правило вложенности скобок). Считайте, что баланс проверяется для одного типа скобок, а строка может содержать произвольные символы
- 2 Заданы две строки: S и S1. Найти количество вхождений S1 в S как подстроки
- 3 Написать программу, считающую количество нулевых элементов в массиве

# Полезные ссылки

- Driving Compilers by Fabien Sanglard
  - <https://fabiensanglard.net/dc/index.php>
  - Серия заметок про различные части компилятора как целого
  - Многие слайды мотивированы именно ей
- How to Compile and Run C Program in Linux Using gcc?
  - <https://cs-fundamentals.com/c-programming/how-to-compile-c-program-using-gcc>

# Про домашние задания

#include <stdio.h> и #include "stdio.h" означают разные вещи

```
$ gcc -v hello.c -o hello
```

...

#include "..." search starts here:

#include <...> search starts here:

/usr/lib/gcc/x86\_64-pc-linux-gnu/15.2.1/include

/usr/local/include

/usr/lib/gcc/x86\_64-pc-linux-gnu/15.2.1/include-fixed

/usr/include

End of search list.

...