Практическое занятие 2
Введение в язык C

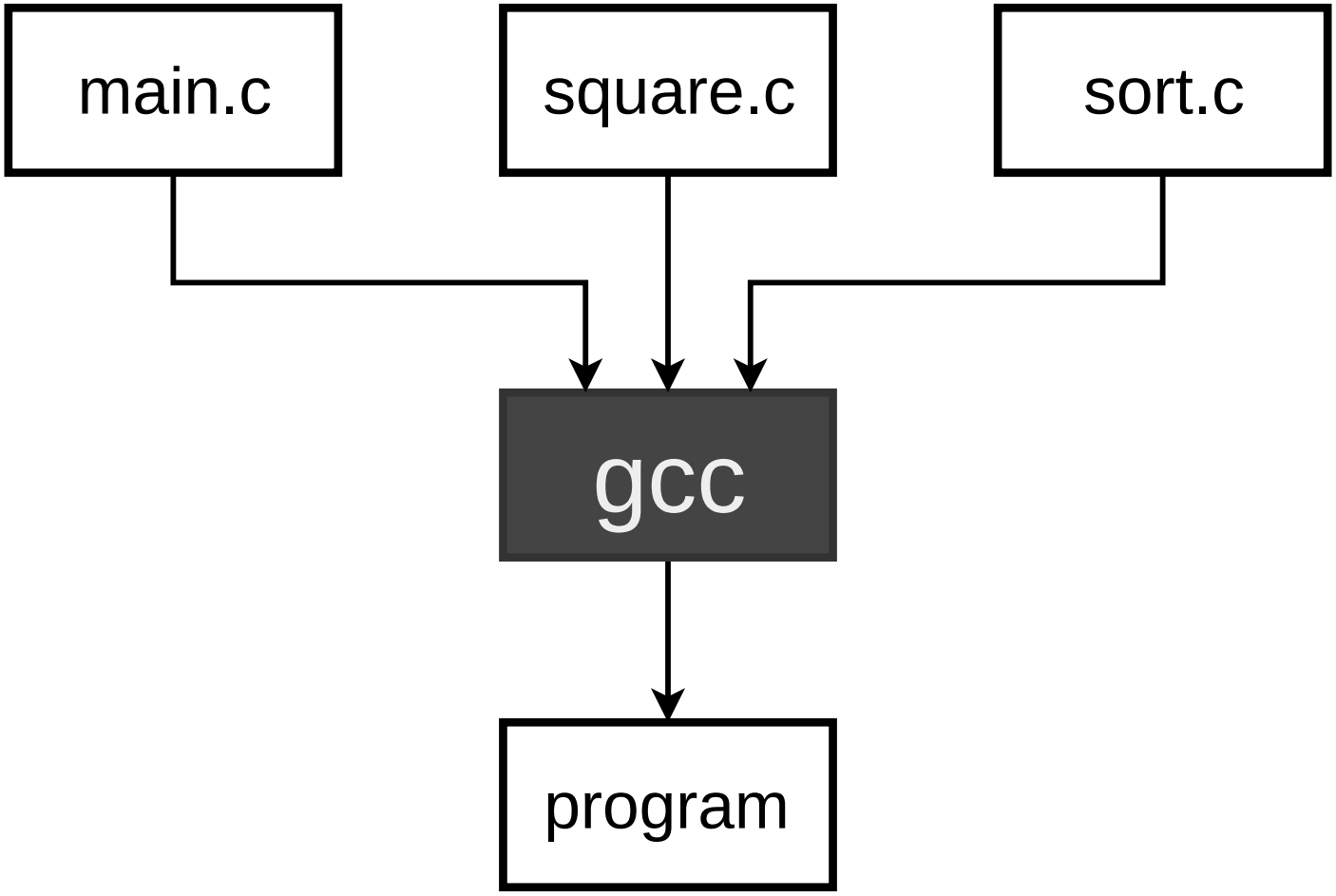
Пименов Евгений Сергеевич

Курс «Программирование»

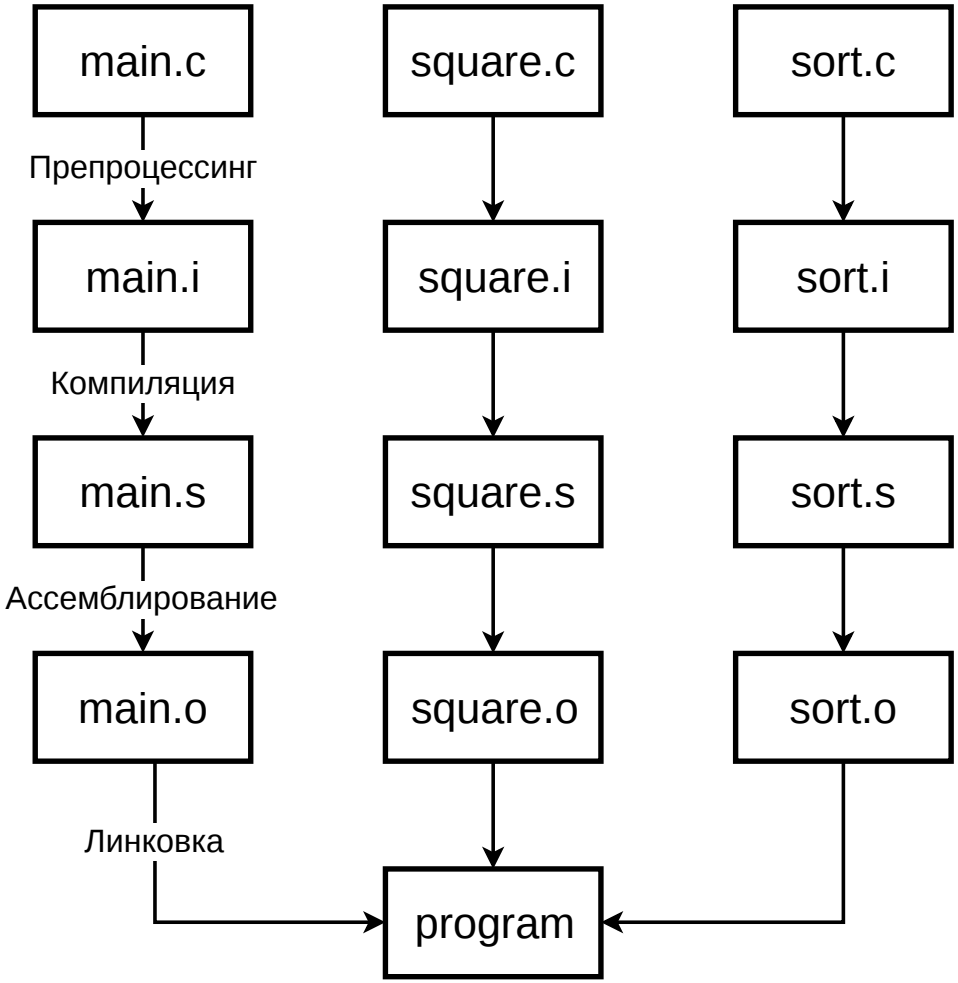
Сибирский государственный университет телекоммуникаций и информатики (Новосибирск)

Осенний семестр, 2016

Процесс компиляции



Процесс компиляции



Зачем это нужно знать?

- На каждом этапе возможны ошибки
- На каждом этапе возможны оптимизации
- На каждой стадии процесс компиляции можно прервать
- Для каждой стадии есть опции gcc

Поэтапная сборка приложения

Препроцессинг	`gcc -E file.c -o file.i`
Компиляция	`gcc -S file.i`
Ассемблирование	`gcc -c file.s`
Линковка	`gcc -o program file.o ...`

Задача

Разработать приложение для перевода температуры из °F в °C. Вывести таблицу для значений от 0°F до 300°F. Формула для перевода:

$$^\circ C = \frac{5}{9} \cdot (^\circ F - 32)$$

Вывод таблицы температур °F, °C


```
01. #include <stdio.h>
02.
03. int main()
04. {
05.     int fahr, celsius;
06.     int lower, upper, step;
07.
08.     lower = 0;
09.     upper = 300;
10.     step = 20;
11.
12.     fahr = lower;
13.
14.     while (fahr <= upper) {
15.         celsius = 5 * (fahr - 32) / 9;
16.         printf("%d\t%d\n", fahr, celsius);
17.         fahr = fahr + step;
18.     }
19.
20.     return 0;
21. }
```

Построчный разбор кода

Объявления переменных знакового целого типа

```
01.     int fahr, celsius;
02.     int lower, upper, step;
```

Построчный разбор кода

Присваивание переменным начальных значений

```
01.     lower = 0;
02.     upper = 300;
03.     step = 20;
```

Построчный разбор кода


```
01.     fahr = lower;
02.
03.     while (fahr <= upper) {
04.         celsius = 5 * (fahr - 32) / 9;
05.         printf("%d\t%d\n", fahr, celsius);
06.         fahr = fahr + step;
07.     }
```

Построчный разбор кода


```
printf("%d\t%d\n", fahr, celsius);
```

Построчный разбор кода


```
printf("%d\t%d\n", fahr, celsius);
```

- `%d` — спецификатор для знакового целого в десятичной (**d**ecimal) СС

Построчный разбор кода


```
printf("%d\t%d\n", fahr, celsius);
```

- `%d` – спецификатор для знакового целого в десятичной (**d**ecimal) СС
- `\t` – символ табуляции

Построчный разбор кода


```
printf("%d\t%d\n", fahr, celsius);
```

- `%d` – спецификатор для знакового целого в десятичной (**d**ecimal) СС
- `\t` – символ табуляции
- `\n` – символ перевода строки

Построчный разбор кода


```
celsius = 5 * (fahr - 32) / 9;
```

Почему не написали `5 / 9 * (fahr - 32)`?

$$^\circ C = \frac{5}{9} \cdot (^\circ F - 32)$$

Арифметичесие операторы С

Оператор	Описание
`++ --`	Постфиксные инкремент и декремент
`++ --`	Префиксные инкремент и декремент
`* / %`	Умножение, деление, остаток
`+ -`	Сложение, вычитание
`=`	Присваивание
`+= -=` `*= /=` `%=`	Присваивание через сумму, разность, произведение, частное, остаток

http://en.cppreference.com/w/c/language/operator_precedence

Компиляция и запуск


```
$ gcc -o temperature -Wall temperature.c
$ ./temperature
0       -17
20      -6
40      4
60      15
80      26
100     37
120     48
140     60
160     71
180     82
200     93
...
```

Какие проблемы и недоработки есть в приложении?

Компиляция и запуск


```
$ gcc -o temperature -Wall temperature.c
$ ./temperature
0       -17
20      -6
40      4
60      15
80      26
100     37
120     48
140     60
160     71
180     82
200     93
...
```

- Не подписаны столбцы
- Числа выровнены по левому краю
- Изменение верхней и нижней границы требует перекомпиляции
- * Теряется точность из-за целочисленных вычислений

Ввод данных

Было:

```
lower = 0;
upper = 300;
```

Стало:

```
printf("Input lower and upper bound (°F): ");
scanf("%d", &lower);
scanf("%d", &upper);
```

Ввод данных

Было:

```
lower = 0;
upper = 300;
```

Стало:

```
printf("Input lower and upper bound (°F): ");
scanf("%d %d", &lower, &upper);
```

Форматирование вывода

Было:

```
printf("%d\t%d\n", fahr, celsius);
```

Стало:

```
printf("%3d\t%3d\n", fahr, celsius);
```

Форматирование вывода


```
$ ./temperature
  0      -17
 20       -6
 40        4
 60       15
 80       26
100       37
120       48
140       60
160       71
180       82
200       93
...
```

Присваивание через сумму


```
01. #include <stdio.h>
02.
03. int main()
04. {
05.     int fahr, celsius;
06.     int lower, upper, step;
07.
08.     step = 20;
09.
10.     printf("Input lower and upper bound (°F): ");
11.     scanf("%d %d", &lower, &upper);
12.
13.     fahr = lower;
14.
15.     while (fahr <= upper) {
16.         celsius = 5 * (fahr - 32) / 9;
17.         printf("%3d\t%3d\n", fahr, celsius);
18.         fahr = fahr + step;
19.     }
20.
21.     return 0;
22. }
```

Присваивание через сумму


```
01. #include <stdio.h>
02.
03. int main()
04. {
05.     int fahr, celsius;
06.     int lower, upper, step;
07.
08.     step = 20;
09.
10.     printf("Input lower and upper bound (°F): ");
11.     scanf("%d %d", &lower, &upper);
12.
13.     fahr = lower;
14.
15.     while (fahr <= upper) {
16.         celsius = 5 * (fahr - 32) / 9;
17.         printf("%3d\t%3d\n", fahr, celsius);
18.         fahr += step;
19.     }
20.
21.     return 0;
22. }
```

Финальный вариант программы


```
01. #include <stdio.h>
02.
03. int main()
04. {
05.     int fahr, celsius;
06.     int lower, upper, step;
07.
08.     step = 20;
09.
10.     printf("Input lower and upper bound (°F): ");
11.     scanf("%d %d", &lower, &upper);
12.
13.     fahr = lower;
14.
15.     while (fahr <= upper) {
16.         celsius = 5 * (fahr - 32) / 9;
17.         printf("%3d\t%3d\n", fahr, celsius);
18.         fahr += step;
19.     }
20.
21.     return 0;
22. }
```

Задание

Задание 1

Скомпилировать hello_world, запуская каждый этап компиляции вручную

Задание 2

1. Добавить заголовки F и C к выводу таблицы температур
2. Изменить порядок вывода строк: от больших значений температур к меньшим

Домашнее задание

1. Разработать приложение для перевода температуры из °C в °F
2. Разработать приложение для перевода секунд в дни, часы, минуты и секунды. Пример использования:

```
$ ./time-converter
Input seconds: 10000
1d 3h 46m 40s
```