

Содержание

Лекция 2: Система типов в С

1 Общие сведения о языке С

- С — регистрозависимый язык программирования.
- Разрядность процессора обычно соответствует размеру указателя (адреса).

2 Система типов в С

Типы данных в С можно разделить на несколько категорий: целочисленные, дробные и специализированные.

2.1 Целочисленные типы

- **int** — целочисленный тип (гарантированный диапазон $[-32767, 32767]$, но фактически 32-битный на большинстве платформ).
- **short** — 16-битное целое число.
- **long** — 32-битное или 64-битное число в зависимости от компилятора и ОС.
- **long long** — 64-битное целое число.
- **char** — 8 бит, диапазон может быть знаковым ($-128..127$) или беззнаковым ($0..255$).
- **_Bool** — логический тип (занимает 8 бит, но хранит 0 или 1).

Дополнительно к этим типам можно использовать модификатор **unsigned** для хранения только положительных значений.

2.2 Дополнительные целочисленные типы

Эти типы требуют подключения заголовочных файлов:

- **size_t** и **ptrdiff_t** (`#include <stddef.h>`) — типы, соответствующие разрядности кода.
- **uintX_t** (`#include <stdint.h>`) — гарантированная битность: 8, 16, 32, 64 бит.
- **(unsigned) _BitInt(X)** (`#include <stdalign.h>`) — гарантированная длина X бит.

2.3 Форматный ввод/вывод целых чисел

```
1 printf("%d", x);    // d - Signed integer (int)
2 printf("%u", x);    // u - Unsigned integer (unsigned int)
3 printf("%x", x);    // x - Hexadecimal (lowercase letters)
4 printf("%X", x);    // X - Hexadecimal (uppercase letters)
5 printf("%hd", x);   // hd - Short integer (short)
6 printf("%hhd", x);  // hhd - Character (char)
7 printf("%ld", x);   // ld - Long integer (long)
8 printf("%lld", x);  // lld - Long long integer (long long)
```

2.4 Дробные типы

- **float** — одинарная точность IEEE-754 ($10^{\pm 38}$), 32 бита.
- **double** — двойная точность, 64 бита.
- **long double** — 128 бит (или 80 бит на некоторых системах), не рекомендуется.

Форматный ввод/вывод дробных чисел:

```
1 printf("%f", x);    // float
2 printf("%lf", y);   // double
```

2.5 Комплексные числа

```
1 #include <complex.h>
2 _Complex float x = 1.5f - 2.3if;
```

3 Дополнительные возможности

Создание пользовательских типов через **typedef**:

```
1 typedef unsigned int uint;
2 uint x;
```

4 Особенности констант

- Отрицательные числа в C — это представление со знаком, но отрицательных констант не существует.
- Ведущий 0 указывает на восьмеричное число.
- Префикс 0b используется для двоичных констант.

5 Заголовочные файлы

В языке C существуют различные стандартные заголовочные файлы, которые подключаются для расширения возможностей работы с типами данных, математическими операциями и многими другими аспектами. Вот основные из них, использованные в данном конспекте:

- `#include <stddef.h>` — этот заголовочный файл включает определения для стандартных типов данных и макросов, таких как `size_t`, `ptrdiff_t` и `NULL`.
- `#include <stdint.h>` — определяет типы с фиксированным размером, такие как `int32_t`, `uint64_t` и другие.
- `#include <complex.h>` — включает поддержку комплексных чисел в языке C, позволяя работать с комплексными числами через типы `_Complex float`, `_Complex double`.
- `#include <stdbool.h>` — добавляет поддержку логического типа `bool` и значений `true` и `false`. Этот заголовочный файл используется для работы с булевыми значениями, что значительно упрощает код.
- `#include <math.h>` — предоставляет математические функции, такие как `sin`, `cos`, `sqrt` и другие.
- `#include <stdio.h>` — предоставляет функции ввода/вывода, такие как `printf`, `scanf`, `fopen` и другие для работы с текстовыми и бинарными файлами.