

Википедия

NULL (Си)

Материал из Википедии – свободной энциклопедии

NULL в языках программирования **Си** и **С++** — **макрос**, объявленный в **заголовочном файле** `stddef.h` (и других заголовочных файлах). Значением этого макроса является зависящая от реализации константа нулевого указателя (англ. *null pointer constant*).

Константа нулевого указателя — это целочисленное константное выражение со значением 0 или (только в Си) такое же выражение, но приведённое к типу `void*`. Константа нулевого указателя, приведённая к любому типу указателей, является **нулевым указателем**. Гарантируется, что нулевой указатель не равен указателю на любой объект (в широком смысле слова, любые данные) или функцию. Гарантируется, что любые два нулевых указателя равны между собой. Разыменовывание нулевого указателя является операцией с неопределённым поведением.

Иначе говоря, реализация предоставляет специальное значение — константу нулевого указателя, которую можно присвоить любому указателю и такой указатель при сравнении не будет равен любому «корректному» указателю. То есть, можно считать, что нулевой указатель не содержит корректного адреса в памяти.

Содержание

Использования

Размерность указателя

Разыменовывание нулевых указателей

Нулевые указатели в С++

См. также

Примечания

Ссылки

Использования

Нулевые указатели придуманы как удобный способ «отметить» указатели, которые заведомо не указывают на корректный адрес в памяти. Например, при объявлении указателя как автоматической переменной его значение не определено. Чтобы отметить, что этот указатель ещё не содержит корректный адрес в памяти, такому указателю присваивают константу нулевого указателя:

```
void f(void)
{
    int *x = NULL;
    /* ... */
}
```

Хорошим стилем программирования является присваивание указателю после освобождения памяти, на которую он ссылался, нулевого указателя. Кроме этого, применение обнуления указателей актуально для безопасности освобождения памяти: операция **delete** в C++ (**free** в Си) безопасна для нулевого указателя. Например:

```
TYPE *foo = new TYPE();
//использование foo
delete foo; // foo != NULL
//некоторый код программы
delete foo; //ОШИБКА! память уже недоступна
```

в то время как в таком варианте ошибки не будет

```
TYPE *foo = new TYPE();
//использование foo
delete foo; // foo != NULL
foo = NULL; // foo == NULL
//некоторый код программы
delete foo; //ошибки нет: delete проверяет значение foo
```

Размерность указателя

При вызове функции в один из аргументов может быть передан NULL. Макрос NULL в разных компиляторах может быть определен различными способами, в том числе

```
#define NULL 0
```

```
#define NULL ((void *)0)
```

В первом случае NULL имеет тип `int`, а во втором случае – тип `void*`. Существуют архитектуры, где `sizeof(int) != sizeof(void*)`, тогда на разных платформах в функцию будет приходит разное количество байт, что может нарушить её работу. В настоящее время предпринимается попытка решить эту проблему в Си путём введения `nullptr`, см предложение N 2394^[1].

Разыменовывание нулевых указателей

Разыменовывание нулевого указателя является операцией с неопределённым поведением. На реализацию не накладывается *никаких* ограничений: может произойти, например, обращение к памяти, не предназначенной для использования данной программой (то есть при чтении будет считан «мусор», а при записи – значение будет записано в область памяти, не принадлежащую программе). Например, в DOS запись по нулевому адресу затрёт как минимум нулевой вектор прерываний, так что следующий вызов `int 0` приведёт, скорее всего, к зависанию системы. Однако чаще всего это приводит к ошибке времени выполнения (если в операционной системе реализована защита памяти и доступ в невыделенную процессу память блокируется). Например, в Windows 9x сообщение «Общая ошибка защиты» – «Программа выполнила недопустимую операцию и будет закрыта» (англ. *general protection fault, GPF*) выдаётся чаще всего в тех случаях, когда программа обращается в память по некорректному (в том числе неинициализированному или уже освобождённому) указателю. В Unix-подобных операционных системах в таких ситуациях процесс получает сигнал SIGSEGV и его обработчик выводит сообщение «Segmentation fault».

Нулевые указатели в C++

Если брать конкретную реализацию NULL по исходным файлам, то он может быть определен как `(void*)0` или как `0`. Использование NULL в проектах на языке C++ может приводить к ошибкам. Например

```
int (ClassName::*pf)() = NULL;
```

приведет к ошибке компиляции в случае, если NULL определен как `(void*)0` (например опосредованно был включен заголовок, где стандартное для C++ определение NULL перекрывается). Поэтому в программах на C++ не рекомендуется использовать NULL в виде `((void *)0)`. В стандарте C++11 для обозначения нулевого указателя добавлено новое ключевое слово `nullptr`^{[2][3]}.

См. также

- [Ошибка сегментации](#)
- [Вариативная функция](#)

Примечания

- 1 N 2394 (<http://www.open-std.org/jtc1/sc22/wg14/www/docs/n2394.pdf>) (англ.) ?. *Open Standards*. Дата обращения: 22 мая 2020. Архивировано (<https://web.archive.org/web/20200727055706/http://www.open-std.org/jtc1/sc22/wg14/www/docs/n2394.pdf>) 27 июля 2020 года.
- 2 *JTC1/SC22/WG21 — The C++ Standards Committee* *SC22/WG21/N2431 = J16/07-0301 «A name for the null pointer: nullptr»* (<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2007/n2431.pdf>) (англ.) (PDF) (недоступная ссылка). *JTC1.22.32. The C++ Standards Committee* (2 октября 2007). Дата обращения: 4 октября 2010. Архивировано (<https://www.webcitation.org/65NGO4mjK?url=http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2007/n2431.pdf>) 11 февраля 2012 года. (англ.)
- 3 Scott Meyers, Summary of C++11 Feature Availability in gcc and MSVC (<http://www.aristeia.com/C++0x/C++0xFeatureAvailability.htm>) Архивная копия (<http://web.archive.org/web/20111026204345/http://www.aristeia.com/C++0x/C++0xFeatureAvailability.htm>) от 26 октября 2011 на Wayback Machine, 16 August 2011

Ссылки

- <http://c-faq.com/null/null1.html> Архивировано (<https://www.webcitation.org/6EibLyMXR?url=http://c-faq.com/null/null1.html>) 26 февраля 2013 года.
- <http://c-faq.com/null/varieties.html> Архивная копия (<http://web.archive.org/web/20110818052648/http://c-faq.com/null/varieties.html>) от 18 августа 2011 на Wayback Machine

Источник — [https://ru.wikipedia.org/w/index.php?title=NULL_\(Си\)&oldid=120661383](https://ru.wikipedia.org/w/index.php?title=NULL_(Си)&oldid=120661383)

Эта страница в последний раз была отредактирована 15 марта 2022 в 12:22.

Текст доступен по лицензии Creative Commons Attribution-ShareAlike; в отдельных случаях могут действовать дополнительные условия.

Wikipedia® — зарегистрированный товарный знак некоммерческой организации Wikimedia Foundation, Inc.