

Лабораторная работа 1 (0001 = 1)

Ввод-вывод при помощи `libc`

Версия 2024 г. — засчитывается только в 2024 г. Актуальная версия — в <https://gitlab.com/illinc/gnu-asm>

Цель работы: изучить размеры стандартных типов C/C++ на выбранной платформе; сопоставить размеры стандартных типов C/C++ на различных платформах; научиться использовать функции ввода-вывода `libc`.

Все задания данной лабораторной работы выполняются на чистом C/C++, без использования ассемблера. Как и для всего курса — если иное не указано явно, компилятор должен быть из коллекции GCC (среда Microsoft Visual Studio недопустима), подробнее в разделе «Компилятор, IDE, отладчик» регламента.

Штраф за одно пропущенное обязательное задание:

- 1 балл, если лабораторная работа сдаётся на первом занятии;
- 2 балла — если на втором и последующих.

Дополнительный бонус за сдачу на первом занятии: +2 балла.

Задание на лабораторную работу

Задание Л1.31. Разработайте программу, выводящую на стандартный вывод группу, номер и состав команды при помощи функции `puts()` библиотеки `libc`.

При работе в ОС MS Windows возможны проблемы с кодировкой русского языка. Если они возникли — используйте транслит или любые доступные вам способы настройки.

Задание Л1.32. Укажите для платформы, где выполняется работа:

- ОС и разрядность ОС;
- компилятор (должен относиться к коллекции GCC/MinGW) и его версию;
- разрядность сборки (собираемая программа может работать в 32-битном режиме даже под 64-битной ОС — в режиме совместимости);
- архитектуру процессора, назначение платформы.

Компьютер с процессором x86/x86-64 под управлением GNU/Linux, BSD (в том числе Mac OS X) или MS Windows — платформа общего назначения.

При помощи оператора `sizeof` языка C/C++ выясните, сколько байтов занимают на выбранной платформе переменные следующих типов: `char`, `signed char`, `unsigned char`, `char*`, `bool`, `wchar_t`, `short`, `int`, `long`, `long long`, `float`, `double`, `long double`, `size_t`, `ptrdiff_t`, `void*`.

Штраф – 2 балла, если выводятся только числа, без пояснений, и непонятно, где размер какого типа.

Бонус +2 балла, если при помощи макроса пояснения выводятся так, что в коде каждое имя типа в Л1.32 встречается единожды.

Задание Л1.33. Бонус +2 балла. Выполните измерения согласно заданию Л1.32 на платформах, доступных на ВЦ (таблица Л1.1).

Платформы для измерения

Таблица Л1.1

Процессор	ОС	Компилятор	разрядность сборки
x86-64	GNU/Linux 64	GCC	64
x86-64	GNU/Linux 64	clang	64
x86-64	GNU/Linux 64	Intel	64
x86-64	MS Windows 64	GCC (MinGW)	64
x86-64	MS Windows 64	Microsoft	64
x86-64	MS Windows 64	Microsoft	32

Связка GNU/Linux 64 + GCC 64 широко используется в онлайн-компиляторах. На `godbolt.org` (ОС GNU/Linux 64) доступны сборка компиляторами GCC 64, clang 64 и ICC (Intel C++ Compiler) 64 с возможностью запуска; а также сборка без запуска для множества других компиляторов, в том числе для не-x86 процессоров.

ОС MS Windows 64 и компиляторы GCC и Microsoft доступны на ВЦ локально (для дистанционных занятий — на терминале ВЦ).

Не возбраняется использование инструментов, установленных дома.

Штраф — 2 балла за платформу таблицы Л1.1, если в аудитории она доступна¹, а данных по ней нет.

Бонус +2 балла за платформу. При подготовке к работе выполните измерения на платформе, отсутствующей в таблице Л1.1 (укажите ОС, компилятор, режим (разрядность) сборки, архитектуру процессора, назначение платформы — без этих сведений баллы не начисляются).

Обратите внимание на размеры целочисленных типов и типов с плавающей запятой. Какие из них **на всех платформах таблицы Л1.1** имеют разрядность 16, 32, 64 бита, учитывая, что байт x86/x86-64 — октет (8 бит)?

Задание Л1.34. Разработайте программу на языке C/C++, создающую массивы из $N = 5$ чисел и инициализирующую их N одинаковыми значениями x :

- Ms из 16-битных целых чисел ($x = 0xFADE$);
- Ml из 32-битных целых чисел ($x = 0xADE1A1DA$);
- Mq из 64-битных целых чисел ($x = 0xC1A551F1AB1E$);
- Mfs из 32-битных чисел с плавающей запятой (x из таблицы Л1.2);
- Mfl из 64-битных чисел с плавающей запятой (x из таблицы Л1.2).

Тип элементов каждого из массивов определите по результатам заданий Л1.32/Л1.33.

¹если платформа недоступна в лабораторной аудитории либо её убрали с ВЦ вообще (в 2021 году с ВЦ исчезла связка MS Windows 64 + GCC 32), штраф не начисляется

Варианты начальных значений элементов с плавающей запятой

Таблица Л1.2

$(N^0 - 1) \% 2 + 1$	Вариант
1	$x = \frac{5}{3}$
2	$x = -\frac{8}{5}$

Не используйте тип *long* на 32/64-битных платформах, так как его размер нестабилен. На 16-битной платформе (если найдёте такую) *long* может быть использован как 32-битный тип.

Не используйте типы фиксированной разрядности *intX_t/uintX_t*, так как модификаторы размера форматных полей *printf()/scanf()* определены не для них, а для *short, long long* и *double*.

Выведите каждый из массивов на экран при помощи функции `libc printf()`:

- каждый из целочисленных массивов дважды — как в знаковом десятичном (формат *d*), так и в шестнадцатеричном (*X*) виде, чтобы убедиться, что короткие значения не расширены до 32 бит, а длинные — не усечены; в шестнадцатеричном виде дополняйте код ведущими нулями до необходимого количества цифр (то есть: для 16-битных *short* используйте *04hX*, для 32-битных на 32/64-битных платформах *int* — *08X*, для 64-битных на 32/64-битных платформах *long long* — *016llX*);
- каждый из массивов с плавающей запятой также выведите дважды — с двумя знаками после десятичной запятой (формат *f*: для *float* используйте *.2f*, для *double* — *.2lf*) и в экспоненциальной форме (формат *e*: *e* и *le*).

Обратите внимание, что для типов, отличных от *int/unsigned/float*, необходимо указывать размер при помощи модификатора перед форматом ввода/вывода.

Штраф – 1 балл, если вместо именованной константы *N* здесь и/или позже используется литерал 5.

Бонус +1 балл, если вывод массива в двух формах описан как функция и в последующих заданиях используется вызов этой функции, а не копирование и вставка; **+2 балла**, если эта функция описана как единый для всех массивов шаблон и принимает тип как параметр шаблона, а адрес начала *M*, длину *N* и форматы с модификатором размера как параметры функции; **+3 балла**, если вывод описан как единый для всех массивов макрос с соответствующими параметрами.

Задание Л1.35. Для одного из массивов *M* (по варианту согласно таблице Л1.3) выведите на экран *адреса*:

Варианты массива M

Таблица Л1.3

$(N^0 - 1)\%5 + 1$	Вариант
1	Ms
2	Ml
3	Mq
4	Mfs
5	Mfl

- начала массива — M ;
- начального (нулевого) элемента массива — $\&(M[0])$;
- следующего (с индексом 1) элемента массива — $\&(M[1])$;

при помощи функции `libc printf()` как указатели (формат p). Сравните полученные значения между собой и с размером элемента массива M .

Задание Л1.36. Для каждого массива M из пяти созданных введите с клавиатуры новое значение элемента $M[i]$, $i = 2$ при помощи функции `libc scanf()`.

Проанализировав возвращённое `scanf()` значение, определите корректность ввода; при необходимости отобразите сообщение об ошибке при помощи функции `libc puts()`. Очистка буфера после некорректного ввода во всех заданиях данной лабораторной работы необязательна.

Выведите массивы на экран до и после ввода, каждый раз — в обеих формах, описанных в Л1.34; убедитесь, что элемент $M[i]$ приобрёл ожидаемое значение, а другие элементы массива не изменились (если изменились — проверьте, верно ли вы указали модификатор размера).

В данном задании необходимо передать функции `scanf()` адрес $M[i]$, а не промежуточной переменной — иначе нет смысла контролировать значение соседних элементов массива. **Штраф –2 балла**, если используется промежуточная переменная для ввода-вывода.

Задание Л1.37. Бонус +1 балл. Если поддерживается модификатор размера hh , выполните задания Л1.34 и Л1.36 также для массива Mb из N 8-битных целых чисел (0xED).

Проверяйте перед защитой, действительно ли hh поддерживается! Если некорректность ввода/вывода выяснится в процессе защиты, задание не засчитывается.

Обходной способ ввода-вывода байта (с использованием либо потоков, либо `printf()/scanf()`) может быть оценен на +1 балл независимо от Л1.37, но вы-

полнением Л1.37 не является, так как неизбежно использование промежуточных переменных.

Задание Л1.38. Бонус +1 балл. Для одного из массивов M (по варианту согласно таблице Л1.3) введите с клавиатуры новое значение всех пяти элементов при помощи одного вызова функции `libc scanf()`.

Проанализировав возвращённое `scanf()` значение, определите корректность ввода; при необходимости отобразите сообщение о количестве введённых и не введённых элементов.

Выведите массив на экран до и после ввода; убедитесь, что количество изменившихся элементов соответствует ожиданиям.

Задание Л1.39. Бонус +2 балла для пар, обязательное для троек. Введите с клавиатуры (каждую строку — одним вызовом `scanf()`):

- слово (строку без пробелов) $s1$ (формат s без модификаторов);
- слово $s2$ таким образом, чтобы принимающий его буфер гарантированно не переполнился: если буфер длины k — вводить не более $k - 1$ символов (ширина поля ввода задаётся аналогично ширине поля вывода);
- строку, возможно, содержащую пробелы $s3$ (формат `[]` — регулярное выражение Perl).

Выведите на экран при помощи функций `libc` строки «*** $s1$ ***», «*** $s2$ ***», «*** $s3$ ***» (между звёздочками должна быть введённая строка, а не литералы $s1$ - $s3$) и убедитесь, что ввод корректен.

Л1.1. Дополнительные бонусные и штрафные баллы

—3 балла за утечку памяти (выделенные, но не освобождённые блоки динамической памяти).

Л1.2. Ссылки на теоретические сведения

7.6.2. Ввод-вывод с помощью `libc`

Л1.3. Вопросы

- Какие функции `libc` используются для форматированного ввода/вывода?
- Как задаётся формат ввода/вывода для `scanf()/printf()`?
- Как задаётся размер вводимых/выводимых чисел (а для строк — размер символа `char/wchar_t`) для `scanf()/printf()`?