

Типы данных. 64-битные компиляторы (Linux x86_64, Windows x64)

(Размер регистров процессора 8 байт)

Disclaimer: размеры типов в Стандарте Си не зафиксированы, ниже приведены значения для популярных современных компиляторов.

Компиляторы: Linux GCC x86_64; 64-битные версии Microsoft Visual C++ (*)

Автор: Третьяков Андрей, МФТИ

Тип	Размер (<i>n bits</i> или <i>n/8 bytes</i>)	Диапазон ($-2^{n-1} .. 2^{n-1}-1$ или $0 .. 2^n-1$ для целых)	printf	scanf
char	8 bits / 1 byte	'\0' .. '\37' '0' .. '9' 'A' .. 'Z', 'a' .. 'z' 'А' .. 'Я', 'а' .. 'я' и другие знаки	%c	
char	8 bits / 1 byte	-128 .. 127	только GCC: %hhhd, %hhi	см. int
unsigned char	8 bits / 1 byte	0 .. 255	только GCC: %hhu, %hho, %hhx, %hhX	см. unsigned
short (= short int)	16 bits / 2 bytes	-32 768 .. 32 767	%hd, %hi	
unsigned short int	16 bits / 2 bytes	0 .. 65 535	%hu, %ho, %hx, %hX	
int	32 bits / 4 bytes	-2 147 483 648 .. 2 147 483 647	%d или %i – 10-чная СС	%d – 10-чная СС %i – любая СС
unsigned int	32 bits / 4 bytes	0 .. 4 294 967 295	%u – 10-чная СС %o – 8-чная СС %x – 16-чная (ab cd ef) %X – 16-чная (AB CD EF)	— — %x и %X эквивалентны
long int	64 bits / 8 bytes	$-2^{63} .. 2^{63}-1$	%ld, %li	
unsigned long int	64 bits / 8 bytes	$0 .. 2^{64}-1$	%lu, %lo, %lx, %lX	
long long int (__int64)	64 bits / 8 bytes	$-2^{63} .. 2^{63}-1$	%lld или %lli (GCC) %I64d или %I64i (MS)	%lld, %lli (GCC) не документировано
unsigned long long int (unsigned __int64)	64 bits / 8 bytes	$0 .. 2^{64}-1$	%llu, %llo, %llx, %llX или %I64u, %I64o, %I64x, %I64X (см. long long и unsigned)	%llu, %llo, %llx, %llX (GCC) не документировано
void * (int*, char*, ...)	64 bits / 8 bytes	0x0000000000000000 .. 0xFFFFFFFFFFFFFFFF	%p	

Основное соотношение между целыми типами в различных компиляторах:

sizeof(char) ≤ sizeof(short) ≤ sizeof(int) ≤ sizeof(long) ≤ sizeof(long long)

Минимальные размеры типов: **char** – 8 бит, **short** – 16, **int** – 16, **long** – 32, **long long** – 64

По умолчанию, все целые типы, кроме **char**, знаковые, т.е. имеют спецификатор **signed**, поэтому его можно (и даже лучше) опускать.

Т.о., **signed int** эквивалентно **int**, а **signed long** эквивалентно **long** и т.д. Тип **char** эквивалентен **signed char** или **unsigned char**, в зависимости от [настроек] компилятора.

Тип	Размер (<i>n bits или n/8 bytes</i>)	Диапазон	printf	scanf
float	32 bits / 4 bytes	$\pm 1.18E-38 \dots \pm 3.40E+38$	%f – в форме [-]dddd.ddddd %e, %E – в форме [-]d.dddddE[+/-]dd (<i>экспоненц.форма</i>) %g, %G (<u>реком.</u>) – автом. как %f или как %e,%E scanf – все эквивалентны	
double	64 bits / 8 bytes	$\pm 2.23E-308 \dots \pm 1.79E+308$	%lf, %le, %lE, %lg, %lG (<u>реком.</u>) (см. float)	
long double	128 bits / 16 bytes	$\pm 3.37E-4932 \dots \pm 1.18E+4932$	%Lf, %Le, %LE, %Lg, %LG (<u>реком.</u>) (см. float)	
char *, char []	–	строковые выражения	%s	%s (до пробела, tab'a), gets(), fgets() (<u>реком.</u>)
%			%%	
\			\\	
'			\'	
"			\"	
?			\?	
<New Line>			\n	
<TAB>			\t	

Запись, например, 3.4E+38 означает $3.4 \cdot 10^{38}$.

Ключевое слово **int** может опускаться в следующих декларациях: **signed int**, **unsigned int**, **short int**, **long int**, **const int** и их производных.

В средах Microsoft Visual C++ тип **long double** эквивалентен типу **double**.

(*) 64-битные компиляторы MS Visual C++ ведут себя аналогично 32-битным версиям, за исключением размера указателей, равного 8 байтам. Поэтому в этом случае тип **long** занимает 4 байта, как и тип **int**.