

# Типы данных. 64-битные компиляторы (Windows x64, Linux x86\_64)

(Размер регистров процессора 8 байт)

Компиляторы: Linux GCC x86\_64; 64-битные версии Microsoft Visual C++ (\*)

Автор: Третьяков Андрей, МФТИ

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

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 %g, %G – как %f или %e, %E ( <u>реком.</u> )	—  — %f, %e, %E, %g, %G – 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)	—  — все эквивалентны
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)	—  — все эквивалентны
char *, char []	–	строковые выраж.	%s	%s (до пробела), gets() ( <u>реком.</u> )
%			%%	
\			\\	
'			\'	
"			\"	
?			\?	
<New Line>			\n	
<TAB>			\t	

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

Основное соотношение между целыми типами в различных компиляторах:

**`sizeof(short) ≤ sizeof(int) ≤ sizeof(long) ≤ sizeof(long long)`**

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

Т.о., ***signed char*** эквивалентно ***char***, а ***signed long*** эквивалентно ***long*** и т.д.

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

В 64-битных системах Linux в GCC тип ***long long*** эквивалентен типу ***long*** (соотв., тип ***unsigned long long*** – типу ***unsigned long***), поэтому для него рекомендуется использовать спецификаторы %li, %ld, %lu, %lo, %lx, %lX. Т.о., тип ***long*** занимает 8 байт, а тип ***int*** так и остался 4-байтным.

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***.