

Типы данных. 32-битные компиляторы (Windows x86, Linux x86)

(Размер регистров процессора 4 байта)

Компиляторы: Microsoft Visual C++, Borland C++ Builder 6.0, Linux GCC x86

Автор: Третьяков Андрей, МФТИ

Тип	Аналог в Delphi	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	Char	8 bits / 1 byte	'\0' .. '\37' '0' .. '9' 'A' .. 'Z', 'a' .. 'z' 'А' .. 'Я', 'а' .. 'я' и другие знаки	%c	— —
char	Shortint	8 bits / 1 byte	−128 .. 127	нет (%hi)	— —
unsigned char	Byte	8 bits / 1 byte	0 .. 255	нет (%hu)	— —
short (short int)	Smallint	16 bits / 2 bytes	−32 768 .. 32 767	%hi или %hd	%hi – любая СС %hd – 10-чная СС
unsigned short (unsigned short int)	Word	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	Integer	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)	Cardinal	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)	Longint	32 bits / 4 bytes	−2 147 483 648 .. 2 147 483 647	%li или %ld	%li – любая СС %ld – 10-чная СС
unsigned long (unsigned long int)	Longword	32 bits / 4 bytes	0 .. 4 294 967 295	%lu, %lo, %lx, %lX (см. unsigned)	— — %lx и %lX эквивалентны
long long (long long int, __int64)	Int64	64 bits / 8 bytes	$-2^{63} .. 2^{63}-1$	%Li или %Ld (Borland) %I64i или %I64d (MS) %lli или %lld (GCC)	не документировано
unsigned long long (unsigned long long int, unsigned __int64)	–	64 bits / 8 bytes	$0 .. 2^{64}-1$	%Lu,%Lo,%Lx,%LX или %I64u, ...; или %llu, ... (см. выше) (см. unsigned)	не документировано
void * (int*, char*,...)	Pointer	32 bits / 4 bytes	0x00000000 .. 0xFFFFFFFF	%p	— —

Тип	Аналог в Delphi	Размер (<i>n bits или n/8 bytes</i>)	Диапазон	printf	scanf
float	Single	32 bits / 4 bytes	$\pm 1.18\text{E-}38 \dots \pm 3.40\text{E+}38$	%f – в форме [-]dddd.dxxxxx %e, %E – в форме [-]d.dxxxxxE[+/-]dd %g, %G – как %f или %e, %E (<i>реком.</i>)	— — %f, %e, %E, %g, %G – все эквивалентны
double	Double	64 bits / 8 bytes	$\pm 2.23\text{E-}308 \dots \pm 1.79\text{E+}308$	%lf, %le, %lE, %lg, %lG (<i>реком.</i>) (см. float)	— — все эквивалентны
long double	Extended	80 bits / 10 bytes	$\pm 3.37\text{E-}4932 \dots \pm 1.18\text{E+}4932$	%Lf, %Le, %LE, %Lg, %LG (<i>реком.</i>) (см. float)	— — все эквивалентны
char *, char []	String	–	строковые выраж.	%s	%s (до пробела), gets() (<i>реком.</i>)
%				%%	
\				\\	
'				\'	
"				\"	
?				\?	
<New Line>				\n	
<TAB>				\t	

Запись, например, $3.4\text{E+}38$ 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 32-битных системах тип **long** эквивалентен типу **int** (соотв., тип **unsigned long** – типу **unsigned int**), поэтому для него допустимо использовать **int**-овские спецификаторы %i, %d, %u, %o, %x, %X, однако в целях **большой** переносимости лучше привыкать использовать соотв. %li, %ld, %lu, %lo, %lx, %lX.

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

Самый платформозависимый тип – **long long**. В одних средах его нет вообще, в других вместо него используется **__int64**.

Использование переменных типа **long long** в **scanf** весьма проблематично, т.к. спецификаторы для него не документированы, но иногда можно использовать соответствующие спецификаторы **printf**.