

Богатов Р.Н.

Программирование на языке высокого уровня

C++ ► Лекция 5 ► Символы. Кодировки. Строки

Кафедра АСОИУ ОмГТУ, 2012

«Ходовые» типы данных

Тип	Размер (байт)	Диапазон значений
int	4	-2 млрд ... 2 млрд
float	4	$\pm 10^{\pm 38}$, точность – 7 разрядов
double	8	$\pm 10^{\pm 308}$, точность – 15 разрядов
bool	1	true или false
char	1	-128 ... 127 (код символа)

СИМВОЛЫ

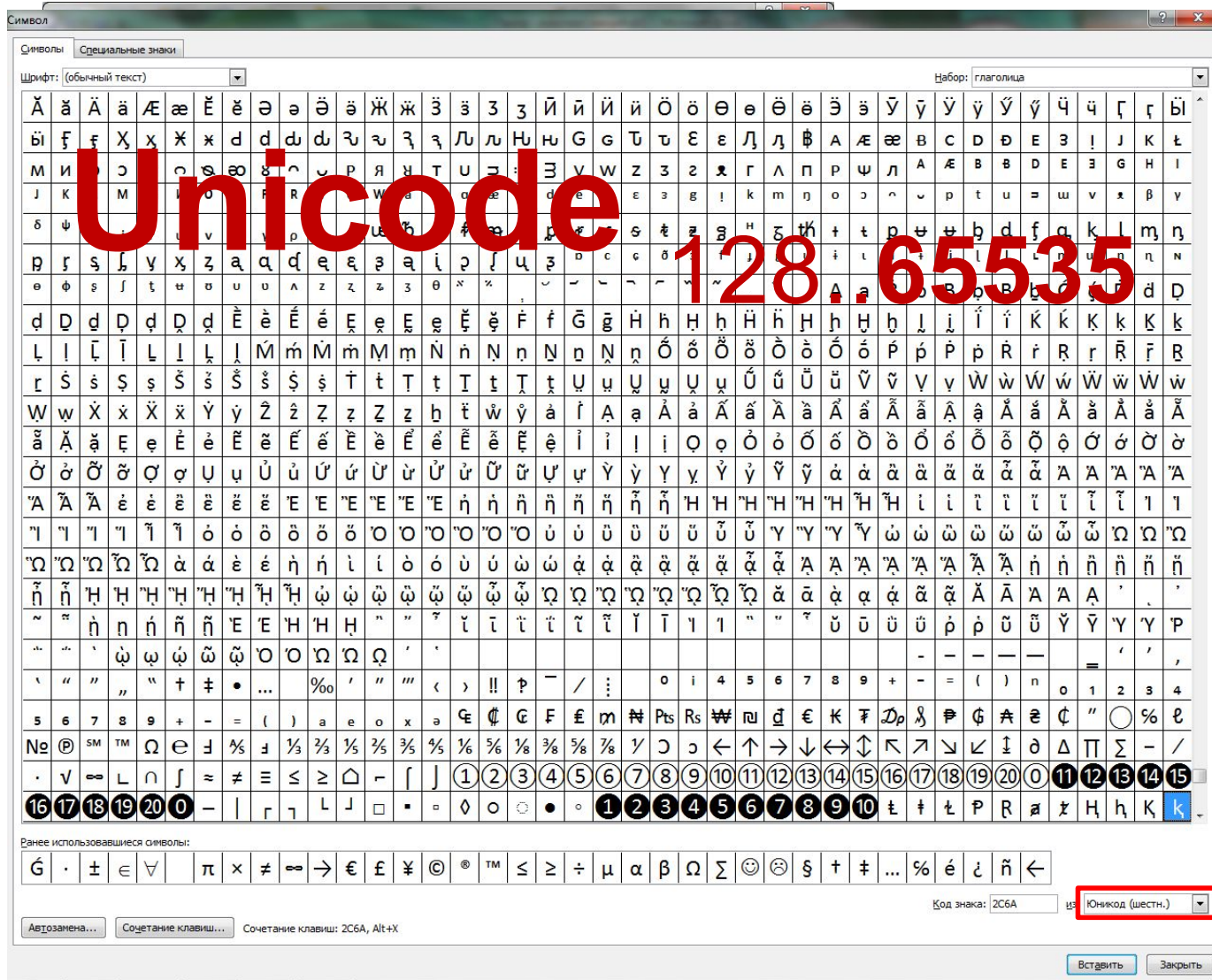
- Сколько-значный код потребуется для представления одного символа текста?
- **A...Z a...z 0...9 " !?.,:;№_ -+*/% \ = <> () {} [] @ # \$ ^ & ~ | '**
– это $27+27+10+32+\dots \geq 96$ символов
- 6 бит = 64 значения
7 бит = 128 значений
8 бит = 256 значений
- Изначальная идея: **один символ – один байт**. Коды 0...127 – основной набор символов, коды 128...255 – для национального алфавита.
- ASCII – базовая 7-битовая кодировка для латинского алфавита и **ОСНОВНЫХ СИМВОЛОВ**. Кириллические кодировки (CP866, Windows CP1251, KOI8-R и другие) отличаются только размещением русских букв в диапазоне кодов 128...255.

Кодировка Windows CP1251

0		16		32		48	0	64	@	80	P	96	`	112	p
1		17		33	!	49	1	65	A	81	Q	97	a	113	q
2		18		34	"	50	2	66	B	82	R	98	b	114	r
3		19		35	#	51	3	67	C	83	S	99	c	115	s
4		20		36	\$	52	4	68	D	84	T	100	d	116	t
5		21		37	%	53	5	69	E	85	U	101	e	117	u
6		22		38	&	54	6	70	F	86	V	102	f	118	v
7		23		39	'	55	7	71	G	87	W	103	g	119	w
8		24		40	(56	8	72	H	88	X	104	h	120	x
9		25		41)	57	9	73	I	89	Y	105	i	121	y
10		26		42	*	58	:	74	J	90	Z	106	j	122	z
11		27		43	+	59	;	75	K	91	[107	k	123	{
12		28		44	,	60	<	76	L	92	\	108	l	124	
13		29		45	-	61	=	77	M	93]	109	m	125	}
14		30		46	.	62	>	78	N	94	^	110	n	126	~
15		31		47	/	63	?	79	O	95	_	111	o	127	□
128	Ђ	144	ђ	160	°	176	°	192	А	208	Р	224	а	240	р
129	Ѓ	145	ѓ	161	±	177	±	193	Б	209	С	225	б	241	с
130	Ѕ	146	ѕ	162	Љ	178	Љ	194	В	210	Т	226	в	242	т
131	Ї	147	ї	163	Ј	179	ј	195	Г	211	У	227	г	243	у
132	„	148	”	164	Ѡ	180	ѧ	196	Д	212	Ф	228	д	244	ф
133	…	149	•	165	Ѓ	181	μ	197	Е	213	Х	229	е	245	х
134	†	150	—	166	‡	182	¶	198	Ж	214	Ц	230	ж	246	ц
135	‡	151	—	167	§	183	·	199	З	215	Ч	231	з	247	ч
136	€	152	□	168	Ё	184	ё	200	И	216	Ш	232	и	248	ш
137	‰	153	™	169	©	185	№	201	Й	217	Щ	233	й	249	щ
138	Љ	154	љ	170	Є	186	є	202	К	218	Ђ	234	к	250	ь
139	‹	155	›	171	«	187	»	203	Л	219	Ы	235	л	251	ы
140	Њ	156	њ	172	–	188	–	204	М	220	Ь	236	м	252	ь
141	Ќ	157	ќ	173	–	189	–	205	Н	221	Э	237	н	253	э
142	Ћ	158	ћ	174	®	190	®	206	О	222	Ю	238	о	254	ю
143	Ќ	159	ќ	175	İ	191	ı	207	П	223	Я	239	п	255	я

ASCII

Вставка символа в Microsoft Word



Строки в C++

```
char s[6];  
s[0] = 72;  
s[1] = 101;  
s[2] = 108;  
s[3] = 108;  
s[4] = 111;  
s[5] = 0;
```

```
char s[6];  
s[0] = 'H';  
s[1] = 'e';  
s[2] = 'l';  
s[3] = 'l';  
s[4] = 'o';  
s[5] = 0;
```

```
char s[] = "Hello";
```

```
printf("What's your name? ");  
char x[20];  
scanf("%s", x);  
printf("Hi, %s!", x);
```

fo: d:\Visual Studio 2008\Projects\C++\proga\Debug\proga.exe

```
What's your name? John  
Hi, John!
```


Строки в C++ – нуль-терминированные

```
// для продвинутых: создание своей функции
int mystrlen(char s[])
{
    int i = 0;
    for(;; i++)
        if (s[i]==0) break;
    return i;
}

void main()
{
    printf("Введите текст: ");
    char t[2000];
    scanf("%1999[^\n]s", t);
    printf("В тексте %d символов!", mystrlen(t));
}
```

```
ки
ст: ");
```

```
#include
<string.h>
```

```
СИМВОЛОВ!", strlen(t));
```



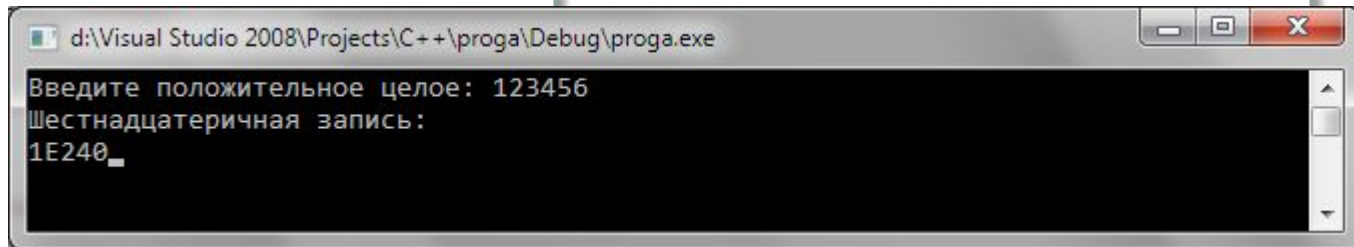
Перевод в шестнадцатеричную систему

СНДСЛОЖИЯ

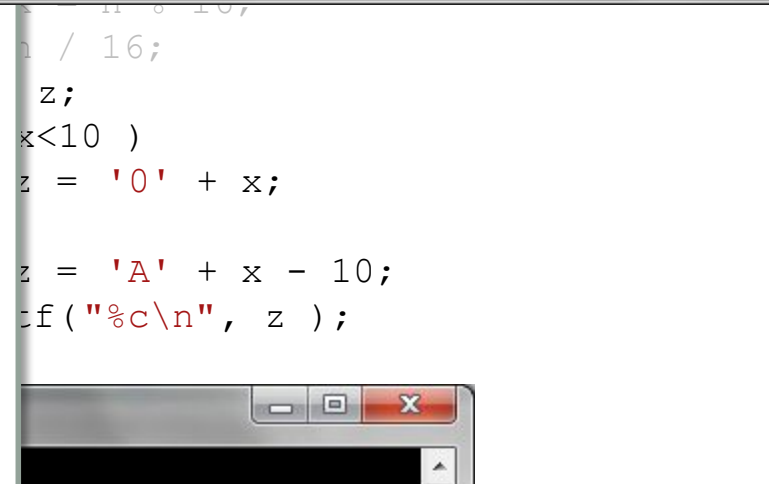
```
char s[8];
int i=0;

do {
    int x = n % 16;
    n = n / 16;
    char z;
    if( x<10 )
        z = '0' + x;
    else
        z = 'A' + x - 10;
    s[i] = z;
    i++;
} while (n>0);
// после цикла i - количество полученных цифр

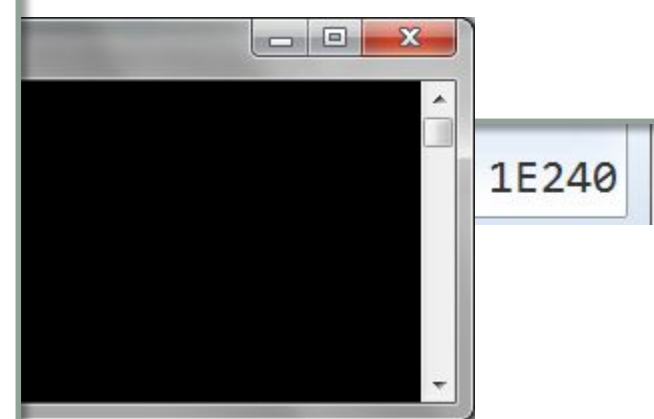
// переворачиваем содержимое строки
for(int j=0; j<i/2; j++)
{
    char c = s[j];
    s[j]= s[i-j-1];
    s[i-j-1] = c;
}
s[i]=0; // терминируем строку и выводим
printf("%s", s);
```



```
d:\Visual Studio 2008\Projects\C++\proga\Debug\proga.exe
Введите положительное целое: 123456
Шестнадцатеричная запись:
1E240_
```



```
int x = n % 16;
n = n / 16;
char z;
if( x<10 )
    z = '0' + x;
else
    z = 'A' + x - 10;
printf("%c\n", z );
```



```
1E240
```

Подсчёт слов, начинающихся с заглавной буквы

```
// оптимальное решение:  
// заранее добавить перед текстом лишний пробел  
printf("Input text: ");  
char s[2000] = " ";  
scanf("%[^\n]s", s+1);  
  
int count = 0;  
for(int i=0; i<strlen(s)-1; i++)  
    if ( s[i]==' ' && s[i+1]>='A' && s[i+1]<='Z' )  
        count++;  
  
printf("There're %d words that start by a capital letter\n", count);
```

```
print  
There're 2 words that start by a capital letter  
);
```

d:\Visual Studio 2008\Projects\C++\proga\Debug\proga.exe

Input text: William Shakespeare

There're 2 words that start by a capital letter

Использование string.h

```
#include  
<string.h>
```

```
printf("Login: ");  
char login[200];  
scanf("%s", login);  
  
// проверка на наличие подстроки  
if( strstr(login, "putin")!=NULL )  
{  
    printf("Unknown error. Terminating...");  
    return;  
}
```

d:\Visual Studio 2008\Projects\C++\proga\Debug\proga.exe

```
...  
Login: Rasputin  
Unknown error. Terminating...
```

Использование string.h

```
#include  
<string.h>
```

```
// Задача: в строке, содержащей путь и имя файла,  
// изменить расширение в имени файла на "txt"  
printf("Введите имя файла: ");  
char filename[200];  
scanf("%[^\n]s", filename);  
int n = strlen(filename);  
printf("Длина строки: %d\n", n );
```

```
// отсечение расширения вручную
```

```
// отсечение расширения с помощью strchr  
char *p;  
// поиск первой точки при просмотре справа-налево  
p = strchr(filename, '.');  
if( p!=NULL )  
    *p = 0; // затираем точку кодом конца строки
```

```
printf("Имя файла без расширения: %s\n", filename);
```

```
strcat(filename, ".txt"); // приклеиваем новое расширение  
printf("Имя файла с новым расширением: %s\n", filename);
```

```
d:\Visual Studio 2008\Projects\C++\proga\Debug\proga.exe  
Введите имя файла: c:\windows\hacked.mp3  
Длина строки: 21  
Имя файла без расширения: c:\windows\hacked  
Имя файла с новым расширением: c:\windows\hacked.txt
```

Домашнее задание

Подсчитать количество «слов» в тексте, который введёт пользователь. За «слово» считать любую последовательность символов, не содержащую пробелов.

Пример текста: *Точка $(x1; y1)$ находится в ... квадранте.*

Ответ: в этом тексте 7 «слов»

Очевидная структура алгоритма:

- общий цикл, пока не кончится текст
 - цикл пропуска пробелов (одного или нескольких идущих подряд)
 - цикл пропуска не-пробелов (составляющих «слово»)
 - инкрементирование счётчика слов