

Семинар №5

ФАКИ 2017

Бирюков В. А.

November 23, 2017

Кодировки. Символы. Тип
char

Кодировка – таблица в которой символам сопоставлены числовые коды.

- ASCII (American standard code for information interchange) – разработана в 1963 году, 7 битовая – можно закодировать 128 символов, но занимает 1 байт.
- Unicode (UTF-8, UTF-16, UTF-32) – можно закодировать 1112064 различных символов. UTF-8 – самая распространённая кодировка, символ занимает от 1 до 4 байт.

char – тип данных, предназначенный для хранения одного символа в кодировке ASCII. Представляет собой целочисленный тип. Диапазон от -128 до 127 (или от 0 до 255).

```
char a = 105;  
char b = 'N';  
printf("%c\n", '+');
```

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

'0' = 48	'A' = 65
'1' = 49	'B' = 66
'2' = 50	...
...	'Z' = 90
'9' = 57	'a' = 97
' ' = 32	...
	'z' = 122

нулевой символ '\0' = 0
 символ переноса строки '\n' = 10
 табуляция '\t' = 9
 backspace '\b' = 8
 звуковой сигнал '\a' = 7
 возврат каретки '\r' = 13

Строки

- Строки в языке C – на самом деле массивы из элементов типа char
- В конце строки должен стоять символ '\0'
- Объявление:

```
char s[10];
```

- Доступ к элементу (Нумерация в строке тоже начинается с 0):

```
printf("%c\n", s[0]);
```

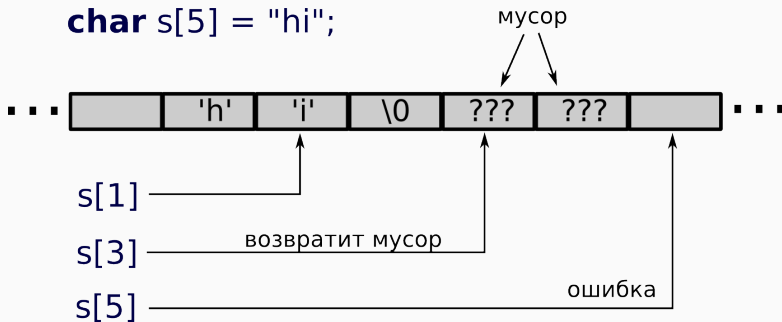


```
char hello_str[10] = {'h', 'e', 'l', 'l',  
                      'o', '\0'};
```

```
char hello_str[10] = "hello";
```

Во втором случае нулевой символ задаётся автоматически

```
char s[5] = "hi";
```



Чтение строки:

```
char string[10];  
scanf("%s", string);
```

Считывание из строки (функция sscanf()):

```
int string[10] = "15:20";  
int h, m;  
sscanf(string, "%d:%d", &h, &m);
```

Можно использовать для конвертирования строки в число.

Функции для работы со
строками (библиотека
string.h)

```
size_t strlen(const char*);
```

возвращает длину строки (size_t – беззнаковый целый тип)

```
#include <string.h>
int string[10] = "hello";
int n = strlen(string);
```

В этом примере n будет равно 5.

```
char* strcpy(char* dest, const char* src);
```

Функция strcpy копирует строку из исходной строки src в строку dest побайтово. Строка dest должна быть достаточного размера.

```
char* strstr(const char* str, const char*  
            substr);
```

Функция strstr находит первое вхождение строки substr в строке str. Возвращает указатель на первый символ этого вхождения. Если такого вхождения нет, то она возвращает NULL.

```
int strcmp(const char*, const char*);
```

лексикографическое сравнение строк (возвращает 0, если строки одинаковые, положительное, если первая строка больше, и отрицательное, если меньше).

```
#include <string.h>
int s1[10] = "kangaroo";
int s2[10] = "kitten";
int n = strcmp(s1, s2);
```

В этом примере $n < 0$ так как строка "kangaroo" меньше строки "kitten" (потому что 'a' < 'i').

Поиск наибольшей общей подпоследовательности

Поиск наибольшей общей
подпоследовательности
Longest common subsequence, LCS

ABBCDAB

BDCABA

Поиск наибольшей общей подпоследовательности

A**B****B****C****D****A****B**

B**D****C****A****B****A**

Наибольшая общая подпоследовательность равна BCAB.

Поиск наибольшей общей подпоследовательности

i
A**B****B****C****D****A****B**

j
B**D****C****A****B****A**

$$C(i, j) = \begin{cases} C(i-1, j-1) + 1, & \text{если } X[i] = Y[j] \\ \max(C(i-1, j), C(i, j-1)), & \text{иначе} \end{cases}$$

		A	B	B	C	D	A	B
	0	0	0	0	0	0	0	0
B	0							
D	0							
C	0							
A	0							
B	0							
A	0							

		A	B	B	C	D	A	B
B	0	0	0	0	0	0	0	0
D	0	0	1					
C	0							
A	0							
B	0							
A	0							

		A	B	B	C	D	A	B
	0	0	0	0	0	0	0	0
B	0	0	1	1				
D	0							
C	0							
A	0							
B	0							
A	0							

		A	B	B	C	D	A	B
	0	0	0	0	0	0	0	0
B	0	0	1	1	1			
D	0							
C	0							
A	0							
B	0							
A	0							

max ↓

→

		A	B	B	C	D	A	B
	0	0	0	0	0	0	0	0
B	0	0	1	1	1	1		
D	0							
C	0							
A	0							
B	0							
A	0							

max ↓

→

		A	B	B	C	D	A	B
	0	0	0	0	0	0	0	0
B	0	0	1	1	1	1	1	1
D	0	0	1	1	1	2		
C	0							
A	0							
B	0							
A	0							

+1



		A	B	B	C	D	A	B
	0	0	0	0	0	0	0	0
B	0	0	1	1	1	1	1	1
D	0	0	1	1	1	2	2	2
C	0	0	1	1	2	2	2	2
A	0	1	1	1	2	2	3	
B	0							
A	0							



		A	B	B	C	D	A	B
	0	0	0	0	0	0	0	0
B	0	0	1	1	1	1	1	1
D	0	0	1	1	1	2	2	2
C	0	0	1	1	2	2	2	2
A	0	1	1	1	2	2	3	3
B	0	1	2	2	2	2	3	4
A	0	1	2	2	2	2	3	4

		A	B	B	C	D	A	B
	0	0	0	0	0	0	0	0
B	0	v	d	d	h	h	h	d
D	0	v	v	v	v	d	h	h
C	0	v	v	v	d	v	v	v
A	0	d	v	v	v	v	d	h
B	0	v	d	d	v	v	v	d
A	0	d	v	v	v	v	d	v

		A	B	B	C	D	A	B
	0	0	0	0	0	0	0	0
B	0	v	d	d	h	h	h	d
D	0	v	v	v	v	d	h	h
C	0	v	v	v	d	v	v	v
A	0	d	v	v	v	v	d	h
B	0	v	d	d	v	v	v	d
A	0	d	v	v	v	v	d	v

		A	B	B	C	D	A	B
		0	0	0	0	0	0	0
B	0	v	d	d	h	h	h	d
D	0	v	v	v	v	d	h	h
C	0	v	v	v	d	v	v	v
A	0	d	v	v	v	v	d	h
B	0	v	d	d	v	v	v	d
A	0	d	v	v	v	v	d	v

Примеры кода

```
#include <stdio.h>
int main(){
    int n;
    scanf("%d", &n);
    for (int i = 0; i < n; ++i){
        int a;
        scanf("%d", &a);
        printf("%d ", a+1);
    }
    printf("\n");
}
```



```
#include <stdio.h>
#include <string.h>
int main()
{
    char s[1001];
    scanf("%1001s", s);
    for (int i = 0; i < strlen(s); ++i){
        if (s[i] == 'a')
            s[i] = 'o';
    }
    printf("%s\n", s);
}
```