Programowanie C++

Wykład 8 (20.04.2016)

```
cout << "Podaj liczbe calkowita\n";
```

Stała napisowa (c-string)

Litery reprezentujemy za pomocą kodów ASCII

```
char litera = 'e';
```

```
char a = 101;
int b = a*5 + 17;
```

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

128	Ç	144	É	160	á	176	▒	192	└	208	╨	224	α	240	≡
129	ü	145	æ	161	í	177	▒	193	┴	209	╤	225	ß	241	±
130	é	146	Æ	162	ó	178	▓	194	┬	210	╥	226	Γ	242	≥
131	â	147	ô	163	ú	179	│	195	├	211	╙	227	π	243	≤
132	ä	148	ö	164	ñ	180	┤	196	─	212	╘	228	Σ	244	⌠
133	à	149	ò	165	Ñ	181	╡	197	┼	213	╒	229	σ	245	⌡
134	å	150	û	166	ª	182	╢	198	╞	214	╓	230	µ	246	÷
135	ç	151	ù	167	º	183	╖	199	╟	215	╫	231	τ	247	≈
136	ê	152	ÿ	168	¿	184	╕	200	╚	216	╪	232	Φ	248	°
137	ë	153	Ö	169	⌐	185	╣	201	╔	217	┘	233	Θ	249	∙
138	è	154	Ü	170	¬	186	║	202	╩	218	┌	234	Ω	250	·
139	ï	155	¢	171	½	187	╗	203	╦	219	█	235	δ	251	√
140	î	156	£	172	¼	188	╝	204	╠	220	▄	236	∞	252	n
141	ì	157	¥	173	¡	189	╜	205	═	221	▌	237	φ	253	²
142	Ä	158	₧	174	«	190	╛	206	╬	222	▐	238	ε	254	■
143	Å	159	ƒ	175	»	191	┐	207	╧	223	▀	239	∩	255	

\b backspace
\f form feed
\n newline
\r carriage return
\t horizontal tab
\" double quote
\' single quote
\0 null
\\ backslash
\v vertical tab
\a bell
\N octal constant where N is an octal constant
\xN hexadecimal constant where N is a hexidecimal constant

```
char apostrof = '\'';
```

```
char napis[6] = {'H', 'e', 'l', 'l', 'o', '\0'};

char napis[] = "Hello";

char *napis = "Hello";
```

Co wypisze następujący kod?

```
napis[2] = 'x';
cout << napis;
```

Co wypisze następujący kod?

```
napis[2] = 0;
cout << napis;
```

Funkcje na c-stringach:

```
#include <cstring>
```

strcpy(s1, s2); kopiuje napis s2 na s1

strcat(s1, s2); dołącza napis s2 na koniec napisu s1

strlen(s1); zwraca długość napisu s1

strcmp(s1, s2); Zwraca 0 jeśli s1 jest identyczne z s2; Mniej niż 0 jeśli s1<s2; Więcej niż 0 jeśli s1>s2

strchr(s1, ch); Zwraca wskaźnik do pierwszego wystąpienia znaku *ch* w napisie s1

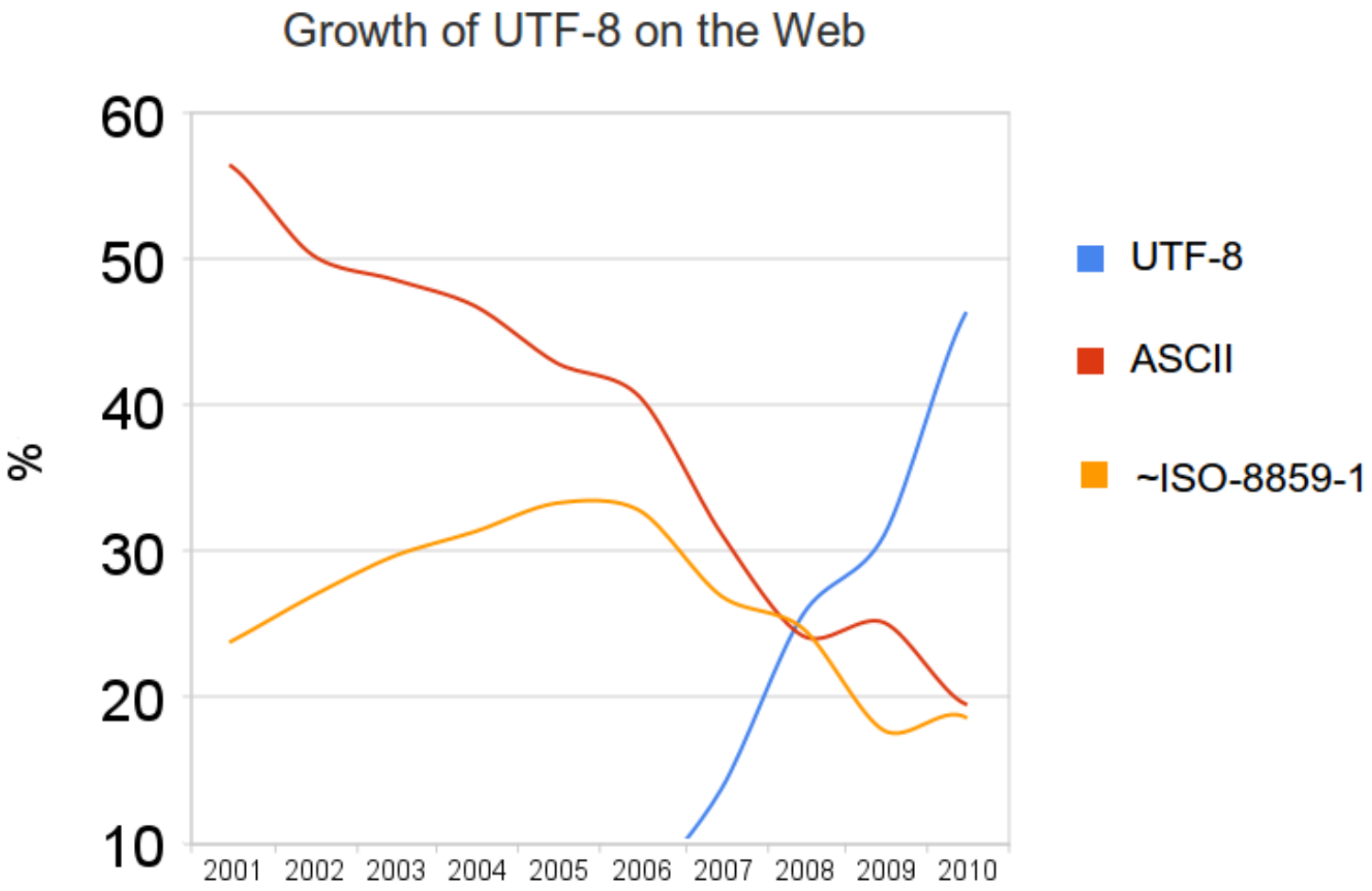
strstr(s1, s2); zwraca wskażnik do pierwszego wystąpienia napisu s2 w napisie s1

Lepsza alternatywa

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string str1 = "Hello";
    string str2 = "World";
    string str3;

    // kopiowanie
    str3 = str1;
    cout << "str3 : " << str3 << endl;
    // dolaczanie
    str3 = str1 + str2;
    cout << "str1 + str2 : " << str3 << endl;
    // calkowita dlugosc:
    int len = str3.size();
    cout << "str3.size() :  " << len << endl;
    return 0;
}
```



Źródło: Wikipedia

Kodowanie UTF-8

Bits of code point	First code point	Last code point	Bytes in sequence	Byte 1	Byte 2	Byte 3	Byte 4
7	U+0000	U+007F	1	0xxxxxxx			
11	U+0080	U+07FF	2	110xxxxx	10xxxxxx		
16	U+0800	U+FFFF	3	1110xxxx	10xxxxxx	10xxxxxx	
21	U+10000	U+1FFFFF	4	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

Wystarczy string oparty na **unsigned char**

Źródło: Wikipedia

Kodowanie UTF-16

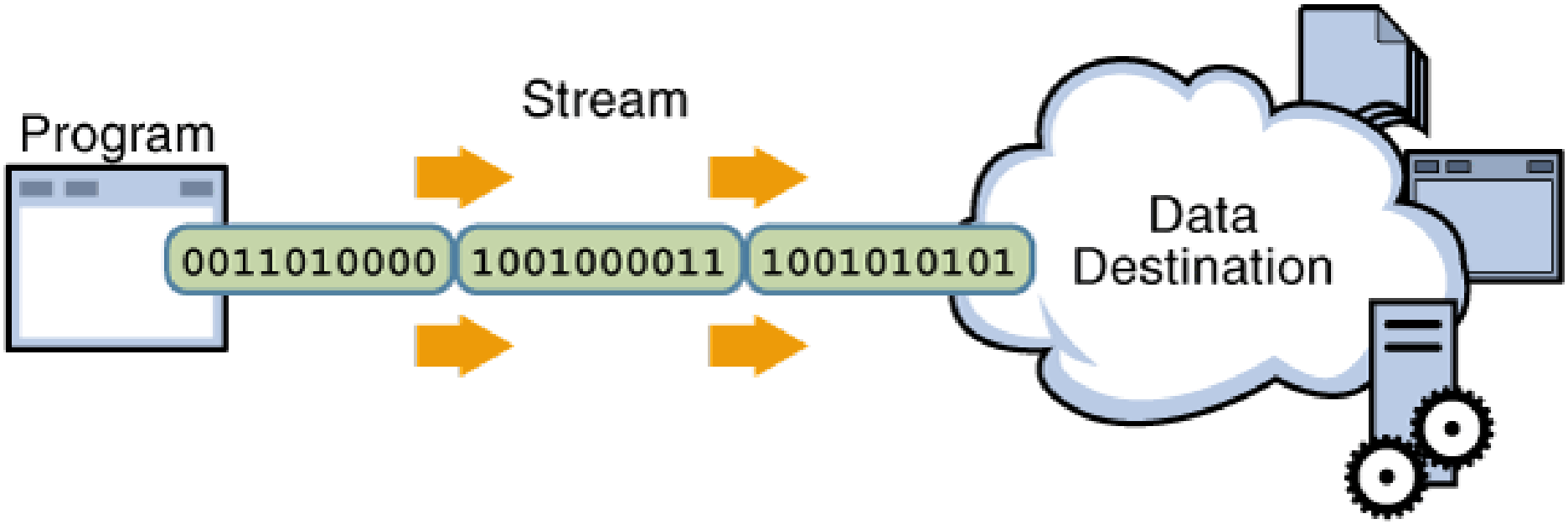
Unicode		Unicode binarnie	UTF-16 binarnie	UTF-16 szesnastkowo
$	U+0024	00000000 00100100	00000000 00100100	0024
€	U+20AC	00100000 10101100	00100000 10101100	20AC
𐐷	U+10437	0001 00000100 00110111	11011000 00000001 11011100 00110111	D801 DC37
𤭢	U+24B62	0010 01001011 01100010	11011000 01010010 11011111 01100010	D852 DF62

```cpp
#include <iostream>
#include <string>
using namespace std;

int main()
{
    wchar_t oldschool = L"Hello";

    wstring str1 = L"Hello";
    wstring str2 = L"World";
    wstring str3;

    // kopiowanie
    str3 = str1;
    wcout << "str3 : " << str3 << endl;
    // dolaczanie
    str3 = str1 + str2;
    wcout << "str1 + str2 : " << str3 << endl;
    // calkowita dlugosc:
    int len = str3.size();
    wcout << "str3.size() :  " << len << endl;
    return 0;
}
```


Program
Stream
0011010000
1001000011
1001010101
Data
Destination

Strumienie

Strumień standardowy
biblioteka:
`<iostream>`

Pobieranie danych ze strumienia

operator lub funkcja	Uwagi
>>	Operator >> domyślnie pomija białe znaki. Wczytuje ciąg znaków do pojawiania się kolejnego białego znaku
`get(char* gdzie, int ile, char ogran = '\n')`	Ogranicznik nie jest wyjmowany ze strumienia
`getline(char* gdzie, int ile, char ogran = '\n')`	Ogranicznik jest wyjmowany ze strumienia
`read(char* gdzie, int ile)`	Nie dopisuje na koncu stringu znaku NULL

Zapisywanie danych do strumienia

operator lub funkcja	Uwagi
<<	przekierowanie do strmienia
`put(char znak)`	Wstawia do strumienia jeden znak
`write(char* skąd, int ile)`	Wstawia do strumienia określoną ilość danych

```cpp
#include<iostream>
#include<sstream>
#include<string>


using namespace std;

// wypisanie pierwszej kolumny z 2-kol pliku
int main(int argc, char *argv[])
{
    string line;
    double x,y;
    while(getline(cin, line)){
    istringstream stream(line); // zamiana string na strumien

        stream >> x >> y;
        cout << x << endl;
  }
    return 0;
}
```

```cpp
#include <iostream>
#include <fstream>
#include <string>
using namespace std;


int main()
{
    string a;
    cout << "Nacisnij Enter aby zakonczyc zapis.\n";
    ofstream f("log.txt");
    cin >> a;
    if (f.good())
    {
        f << a;
        f.close();
    }
    return 0;
}
```

```cpp
#include <iostream>
#include <sstream>
using namespace std;

int main ()
{
    long x;
    string napis;
    sstringstream ss;
    cout << "Podaj dowolna liczbe calkowita: ";
    cin >> x;
    ss << x;
    napis = ss.str();
    cout << "Dlugosc napisu wynosi "
              << napis.size() << " znakow." << endl;
    return 0;
}
```

```
void wypisz_date(ostream &strumien, int dzien, int miesiac, int rok) {
    strumien << dzien << "." << miesiac << "." << rok << endl;
}
```

```
#include <iostream>
using namespace std;

class Date
{
    int mo, da, yr;
public:
    Date(int m, int d, int y)  {
        mo = m; da = d; yr = y;
    }
    friend ostream& operator<<(ostream& os, const Date& dt);
};
ostream& operator<<(ostream& os, const Date& dt)
{
    os << dt.da << '.' << dt.mo << '.' << dt.yr;
    return os;
}
```

```
int main()
{
    Date dt(5, 6, 92);
    cout << dt;
}
```