

Informační a komunikační technologie

Řetězce v C

David Weber

Kabinet K13

weber3@spsejecna.cz

Co dnes probereme

- Datový typ `char`
- Práce s řetězcí
- Užitečné funkce (knihovna `string.h`)

Zpět k datovým typům...

- Zatím jsme pracovali s číselnými proměnnými \Rightarrow `int`, `float`, `double`.
- Avšak např. u funkcí `printf` a `scanf` jsme jako parametr předávali "text".
- \Rightarrow To jsou tzv. **řetězce** (angl. *string*).
`printf("Hello World!");`
- Řetězce se skládají z tzv. **znaků** (angl. *char*).

Znaky I

- Znaky jsou ukládány do datového typu `char`
 - 8 bitů
- Znaky vždy uvádíme mezi apostrofy, např. `'t'`.

```
char znak = 'c';
```

- Pracujeme s nimi stejně, jako s ostatními datovými typy, tzn. lze do nich *přiřazovat*, porovnávat je pomocí `==`, ...
- Pro výpis a načtení znaku používáme formátovou specifikaci `%c`.

```
scanf("%c", &znak);  
printf("Ulozeny znak: %c", znak);
```

Znaky II

- Znaky, které lze do datového typu `char` ukládat nemusí být nutně **tisknutelné**.
- Lze např. uložit i znak pro *odřádkování*.

```
char newLine = '\\n';
```

⇒ Pozor! Ač `\` a `n` představují *dva znaky*, dohromady reprezentují **jeden znak**!

Znaky III

- Datový typ `char` je ve skutečnosti také číselný!
- Např. příkaz

```
printf("%d", 'c');
```

vypíše do konzole číslo 99.

- *Proč tomu tak je?* \Rightarrow Každý znak má svůj kód v tzv. tabulce **ASCII**.

ASCII

- ASCII = *American Standard for Information Interchange*
- Definuje znaky *anglické abecedy* a jiné znaky.
- Každý znak má v tabulce svůj (dnes *osmibitový*) kód, kterým je identifikován.
- Např.
 - a → kód 0x61 = 97
 - b → kód 0x62 = 98
 - c → kód 0x63 = 99
 - M → kód 0x4D = 77
 - M → kód 0x4E = 78
 - ...

Kódování v ASCII

- Každý znak je reprezentován 8 bity = 2 číslice v hexadecimální soustavě.
- Každá číslice určuje **řádek**, resp. **sloupec** v tabulce.
- \Rightarrow V proměnné datového typu `char` je tak uložena **ASCII hodnota daného znaku**.
- Přehled všech znaků v ASCII např. [zde](#).

Tabulka ASCII

Prvních 127 znaků tabulky ASCII (obrázek převzat z [Wikipedie](#)).

ASCII Code Chart

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Řetězce I

- Řetězec představuje jistou **posloupnost znaků**.
- V jazyce C jsou reprezentovány jako **pole znaků**.
 - \Rightarrow Jako datový typ uvádíme tedy `char[]`.
 - Každý řetězec je zakončen znakem tzv. *nulovým znakem* `'\0'` reprezentující konec slova.
- Řetězec lze inicializovat přímo jako pole

```
char pozdrav[] = { 'A', 'h', 'o', 'j', '\0' };
```

\Rightarrow Celkem nepraktické!

Řetězce II

- Lepší znaky daného řetězce uvést do uvozovek, tj. "...".

```
char pozdrav[] = "Ahoj";
```

- Znak `\0` nemusíme uvádět \Rightarrow je doplněn automaticky.
- Pro výpis používáme formátovou specifikaci `%s`.

```
printf("%s", pozdrav);
```

Řetězce III

- Řetězce (stringy) jsou pole znaků \Rightarrow lze s nimi manipulovat pomocí indexů.
- Např. program

```
char pozdrav[] = "Ahoj";  
pozdrav[1] = 't';  
printf("%s", pozdrav)
```

vypíše slovo Atoj.

Knihovna `string.h`

- Pro práci s řetězci se nám bude hodit knihovna `string.h`
- Některé užitečné funkce:
 - `strlen()` → vrátí délku řetězce (počet znaků)
 - `strcat()` → spojí dvojici řetězců do jednoho.
 - `strcpy()` → zkopíruje obsah jednoho řetězce do druhého.
 - `strcmp()` → porovná dvojici řetězců.

Funkce strlen

- STRing LENgth
- Vrací délku řetězce (počet znaků).
- Návrátový datový typ je `int`.

```
char alphabet[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
printf("%d", strlen(alphabet));    // 26
```

Funkce strcat

- ConCATenate STRings
- Jako parametry přijímá **dvojici řetězců** str1 a str2, které **spojí do jednoho** a **výsledek uloží do prvního**.

```
char str1[] = "Hello ";  
char str2[] = "World!";
```

```
strcat(str1, str2);
```

```
printf("%s", str1); // Vypise "Hello World!"
```

Funkce strcpy

- CoPY STRings
- Zkopíruje obsah **druhého řetězce do prvního** (tj. $\text{str2} \rightarrow \text{str1}$)

```
char str1[] = "Hello World!";  
char str2[];    // Prazdny retezec
```

```
strcpy(str2, str1);
```

```
printf("%s", str2); // Vypise "Hello World!"
```


Funkce strcmp

- CoMPare STRings
- Porovná dvojici řetězců, vrátí hodnotu typu `int`
 - 1 = řetězce jsou stejné,
 - jiné číslo = řetězce jsou odlišné.

```
char str1[] = "Hello";  
char str2[] = "Hello";  
char str3[] = "Hi";
```

```
printf("%d\n", strcmp(str1, str2)); // 0 (řetězce jsou  
    stejné)  
printf("%d\n", strcmp(str1, str3)); // -4 (řetězce  
    nejsou stejné)
```

Otázky?

