

Programovanie 10

Znaky a reťazce v jazyku C

Znakové premenné/konštanty

- Znaký sú reprezentované číselnými kódmi (v závislosti na systéme, štandardne ASCII)
 - môžeme používať celočíselné typy, default je typ `char`
 - znakové konštanty sú v apostrofoch, prekladač sám priradí príslušný kód (napr. `'A'` preloží ako číslo 65, ak sa používa ASCII)

```
char znak = 'A';
```

```
znak = znak + 1; //kod písmena 'B'
```

```
if ('A' <= znak && znak <= 'Z') ...
```

Znakové I/O

- scanf/printf, formátovací reťazec %c (s typom char):

```
char znak;
```

```
scanf("%c", &znak);
```

```
printf("Znak %c ma kod %i\n", znak, znak);
```

- špeciálne funkcie getchar, putchar (s typom int):

```
int znak;
```

```
while ( (znak=getchar()) != '\n' )
```

```
    putchar(znak);
```

Znakové funkcie

ctype.h

- testy typu znaku :
 - `isupper`, `islower`, `isalpha...`
- konverzie znakov :
 - `toupper`, `tolower`

```
if (isalnum(znak) )  
    putchar(toupper(znak) );
```

Reťazce

- V jazyku C: reťazec je pole znakov, ktoré je ukončené znakom s kódom 0 (znakový zápis: '\0')

```
char text[] = {'t', 'e', 'x', 't', '\0'};  
    //platny retazec
```

```
char pole[] = {'t', 'e', 'x', 't'};  
//iba pole znakov,  
//nema ukoncovaci znak = neplatny retazec
```

Reťazcová konštanta

- Postupnosť znakov v úvodzovkách: "text"
 - prekladač vytvorí reťazec obsahujúci zadané znaky a ukončovací znak

- Môže sa použiť pri inicializácii poľa:

```
char text[] = "text"; //5 znakové pole
```

- Na iných miestach programu: vytvorí statické pole (niektoré prekladače: read-only) a vráti jeho adresu

```
printf("text");
```

```
//vstupom printf je adresa pola v specialnej pamati
```

Reťazce ako polia/smerníky

- Inicializované (lokálne/globálne) pole:

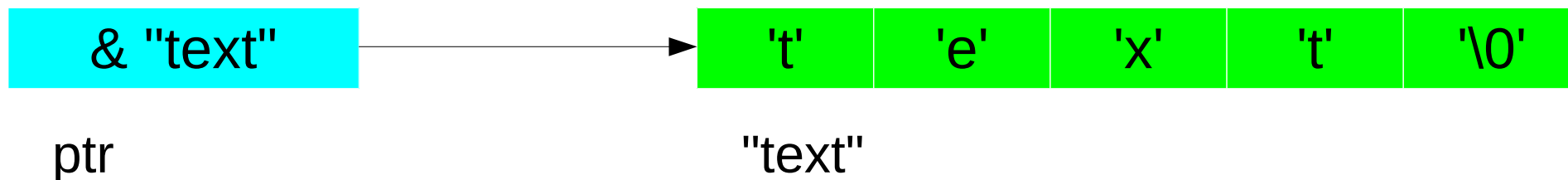
```
char pole[] = "text";
```

't'	'e'	'x'	't'	'\0'
-----	-----	-----	-----	------

pole

- Smerník odkazujúci na reťazcovú konštantu:

```
char *ptr = "text";
```



Pole reťazcov: 2D verzia (zadanie)

```
char pole[3][5] = {"text", "a", "peter"};
```

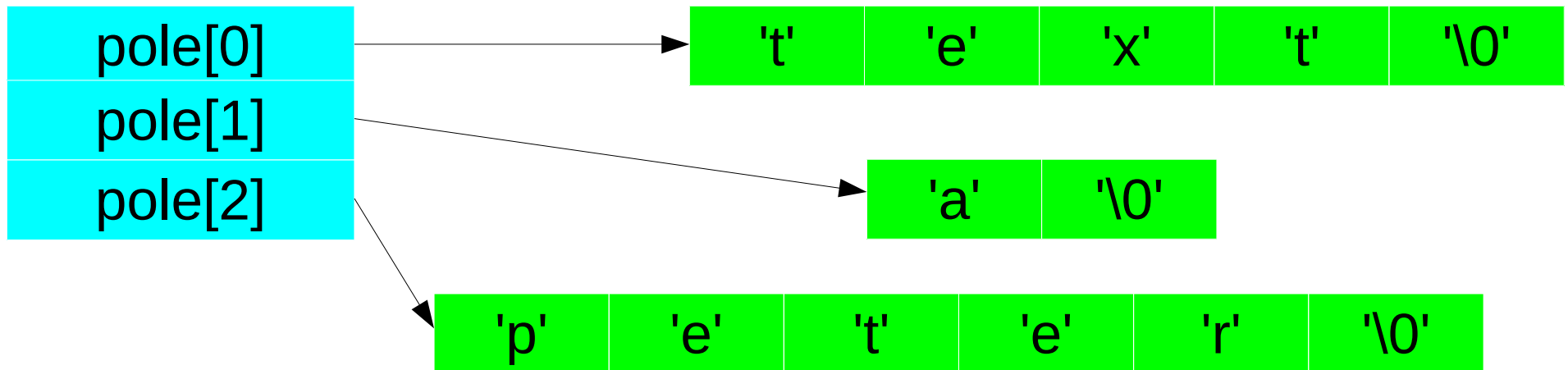
't'	'e'	'x'	't'	'\0'
'a'	'\0'			
'p'	'e'	't'	'e'	'r'

Pozn.: posledný reťazec je neplatný, lebo sa nezmestil posledný znak! Ale ako pole znakov je OK...

- `pole[1]` ... druhý reťazec ("a")
- `pole[0][2]` ... tretí znak prvého reťazca ('x')

Pole reťazcov: smerníky

```
char *pole[3] = {"text", "a", "peter"};
```



- `pole[1]` ... druhý reťazec ("a")
- `pole[0][2]` ... tretí znak prvého reťazca ('x')

Reťazce a I/O

- `printf`, formátovací reťazec `%s` (vstupom je adresa, `char*`):

```
char* mesiace[] = {"januar", "februar", ...};  
printf("Mesiac %i sa nazyva %s\n",  
      i, mesiace[i]);
```

- `scanf`, `%s` – potrebuje adresu dostatočne veľkého poľa znakov, načítava slovo (po biely znak), dopĺňa `'\0'` automaticky:

```
char pole[10];  
scanf("%9s", pole);  
//nacitaj maximalne 9 znakov a ukoncovaci
```

Užitečné reťazcové funkcie

- string.h:
 - strlen : zistí dĺžku reťazca (počet platných znakov okrem '\0')
 - strcmp : porovná reťazce abecedne
 - strcpy : kopíruje obsah reťazca do pripraveného poľa
 - strcat : spojí dva reťazce (musí mať rezervované miesto)
 - ...
- stdio.h
 - sprintf : vytvorí reťazec podobne na základe formátového vstupu
 - ...
- ...