

Programovanie 10

Dátové typy a výrazy

Význam a delenie DT

- Dátový typ:
 - všetky dáta sú fyzicky len sekvencie bitov
 - DT umožňuje priradiť týmto sekvenciám určitý význam
- Príklad: 01000001
 - binárne zapísané číslo 65
 - ASCII kód znaku 'A'
 - vlastná reprezentácia zlomku: $4 + 1/16 = 4.0625$

Dátové typy v jazyku C

- Jednoduché:
 - základné /číselné/: char / int / float / double
 - smerníky
 - enumerácie
- Zložené:
 - štruktúry
 - uniony
- Polia

Rozšírenia dátových typov

- Typ celé číslo:
 - `int`
- Varianty s rôznou veľkosťou:
 - `short / long / long long int`
- Varianty so znamienkom / bez znamienka:
 - `signed / unsigned int`
- Špeciálne typy, závislé na prekladači (`stdint.h`):
 - `int8_t, uint32_t, uint_least64_t, int_fast16_t, ...`

Operátory využívané s typmi

- vlastné pomenovanie typu: `typedef`

```
typedef unsigned short int word;  
word x;
```

- zistenie veľkosti údaju (daného typu): `sizeof`

– vracia veľkosť zabranú v pamäti v bajtoch

```
printf("x zabera %i B\n", sizeof(x));  
potrebna_pamat = 1000*sizeof(word);
```

Použitie dátových typov v programe

- Deklarácia premenných

```
T ident;
```

- Deklarácia funkcií

```
ret_T fun_id( T1 par1, T2 par2, ... );
```

- Zmena dátového typu

```
(dest_T) vyraz
```

- Zistenie veľkosti premennej daného typu

```
sizeof(T)
```

Dátový typ a program

- Dátový typ určuje:
 - Množstvo vyhradenej pamäte
 - Povolené operátory a ich význam
- Dátový typ je určený:
 - deklaráciou premennej / funkcie
 - syntaxou konštanty
 - pri vyhodnocovaní výrazov pri implicitnej konverzii

Dátové typy a I/O

- Funkcie formátovaného vstupu a výstupu (printf/scanf) nepoznajú typ premenných!
- Musíme určiť sami vo formátovacom reťazci
 - pozor na chyby! (pri scanf hrozí fatálne zrušenie)
 - niekedy sa dá využiť na extra funkcionality

```
printf("%i %c", 65, 65); //vypise raz ako  
cislo, raz ako znak
```

```
printf("%x", 0.73); //z hex vystupu vieme  
overit, ako vyzerá binárna reprezentácia 0.73
```


Základné dátové typy a I/O

char	%c	Z
int	%i / %d %x %o	90 5A 112
short int	%hi	90
long int	%li	90
unsigned int	%u	90
float / double	%f (%lf) ...	3.141592
long double	%Lf	3.141592

<https://www.lix.polytechnique.fr/~liberti/public/computing/prog/c/C/FUNCTIONS/format.html>
<http://people.tuke.sk/igor.podlubny/C/Kap6.htm>

Vyhodnocovanie výrazov

- Výraz:
 - kombinácia konštant, identifikátorov (premenných a funkcií) a operátorov
- Výraz sa za behu vyhodnocuje:
 - na základe hodnôt premenných a výsledkov funkcií
 - výsledok je hodnota nejakého typu
 - vyhodnocovanie výrazu môže mať postranné efekty (napr. pri volaní funkcií, ale aj zvýšením hodnôt premenných, priradením hodnôt premenným, ...)

Vyhodnocovanie výrazov

- Operátory:
 - arita (unárny, binárny, ternárny)
 - združovanie (priorita a smer združovania)
 - postranné efekty (++, --, a priradenia)
- Združovanie meníme zátvorkovaním a rozdelením výrazov na jednoduchšie medzikroky
- Aj priradenie je operátor:
 - pri vyhodnocovaní má iný význam má pravá a ľavá strana (bude neskôr), výsledok je priradená hodnota

Vyhodnocovanie výrazov a funkcie

- Ak je súčasť výrazu volanie funkcie:
 - najprv sa musia vypočítať parametre:
 - vyhodnotia sa príslušné výrazy
 - nie je garantované poradie vyhodnotenia!
 - po výpočte hodnôt parametrov sa začne vykonávať funkcia (s dosadenými hodnotami parametrov), zvyšok programu „čaká“
 - výsledok funkcie (hodnota return) sa dosadí do výrazu a pokračuje jeho vyhodnotenie

Vyhodnocovanie a typy

- Operátory (a funkcie) pracujú len s údajmi správneho typu
 - napr. delenie: celočíselné vs. floating point
- Ak sú vstupy iného typu, prekladač sa pokúsi o implicitnú konverziu:
 - pri čítaní vždy na „väčší“ / presnejší typ
 - pri zápise na cieľový typ

`int + float --> float + float`

`int = float --> int = int`

math.h

- Obsahuje užitočné matematické funkcie
 - pow, log, sqrt, ...
 - ceil, floor, ...
 - sin, cos, ... (pozor, pracujú v radiánoch)
- mat. konštanty (závislé na platforme, vid' ďalej):
 - M_PI, M_E, M_SQRT2

```
#define _USE_MATH_DEFINES
```

```
#include <math.h>
```