Datové typy a proměnné

Přednáška 4

Datové typy

- Základní datové typy
- Abstraktní datové typy

- Reprezentace dat daného problému

Základní datové typy

- Dělení
  - Reálné
  - Celočíselné
    - Znaménkové
    - Bezznaménkové
  - Rozšířené, složené, strukturované
    - Pole, soubor, struktura
    - Komplexní číslo, vektor
    - Vyšší nároky na paměť i zpracování

Operace s datovými typy

- Operace spojené s daným datovým typem
- Důležité operace
  - Typické funkce (+, -, *, /, modulo …)
  - Specifické funkce pro daný typ (např. číslo komplexně sdružené)
  - Inicializace
  - Destrukce (a uvolnění zdrojů)
  - Získání hodnoty
  - Zapsání hodnoty
- Některé operace nedávají smysl pro jisté datové typy
  - Reálné typy -> bitový posun, negace bitů
  - Komplexní číslo -> relační operátory

Přesnost datového typu

- Závislá na platformě, překladači, jazyku
- Určuje
  - Rozsah hodnot
  - Přesnost/jemnost typu
  - Velikost zabrané paměti
  - Složitost a časovou náročnost operace
- U reálných typů dána počtem bitů mantisy
- Spojena se zaokrouhlovací chybou

Rozsah datového typu

- Závislá na platformě, překladači, jazyku
- U reálných typů dána počtem bitů exponentu

- Definováno v knihovně `limits.h`
  - int
    - [INT_MIN, INT_MAX]
    - Vzdálenost mezihodnotami je 1
  - double
    - [-DBL_MAX, DBL_MAX]
    - Vzdálenost mezi hodnotami?
      - Liší se podle expomentu

Celočíselné beznaménkové typy

- Různé přesnoti
- V jazyce C:
  - Není dána přesnost, ale hodnota, kterou musí datový typ reprezentovat
  - Menší datový typ se musí beze ztráty uložit do většího
  - long long int musí mít minimálně 64 bitů
- Z hlediska velikosti v paměti:
  - *char <= short <= int <= long int <= long long int (vše unsigned)*
- Z hlediska konverzního řádu
  - *char < short < int < long int < long long int (vše unsigned)*

Celočíselné znaménkové typy

- Menší rozsah v kladných číslech
- Mají záporné hodnoty
- Problém s vložením bezznaménkových čísel do znaménkových

Reálné typy, s pohyblivou desetinnou tečnou

- Z hlediska velikosti v paměti:
  - *float <= double <= long double*
- Z hlediska konverzního řádu
  - *jakýkoli celočíselný typ < float < double < long double*

Konverze datových typů

- Převody jsou v různých jazycích hodně přísné
- Možná ztráta přesnosti při převodu
- Možné přetečení, podtečení při převodu

- Pozor na převod signed na unsigned stejného typu
- Celočíselné typy s více bity než má mantisa reálného čísla se také neuloží správně

```
char c;
int i;
float f;
double d;

c / i ;      // konverze na int
f / d ;      // konverze na double
f / i ;      // konverze na float
             // mají-li oba typy float i int 32 bitů
             //    -> proměnná i se do mantisy nevejde celá
```

Typy definované výčtem

- Datový typ enum
- Zápis v definici v textové podobě
  - Nelze využít pro načtení a tisk
- V programu reprezentován celočíselným datovým typem
  - Přímé použití celočíselných hodnot se nedoporučuje
- Hodnoty možné přiřadit při inicializaci
- Výhodné při použití příkazu switch

- Př. `enum EStav {OK=0,NotInit,BadResult};`

Pole

- Více prvků stejného datového typu
- Indexace prvků -> přístup přes index ke všem prvkům stejně rychlý
- Operace přidání/ubrání prvku -> náročné
- Nutné řešit přístup mimo index (jazyk C neřeší)

Záznam, struktura

- Prvky různého typu
- Nemůže mít prvky stejného typu jako nový typ
  - Nekonečná smyčka
  - Nutné použít ukazatel
- Tvoří nový typ

Union

- Prvky různého typu
- Vždy jen jeden aktivní
- Tvoří nový typ
- Využívá se pro univerzální funkce

Soubor

- Může být
  - Typ
  - Soubor na disku
- Proměnná velikost
  - Nemá index
  - Neomezená velikostí typu indexu
- Sekvenční (sériový) přístup k datům
  - časově různá přístupová doba k datům
- Sériová linka
  - Např. data z teploměru – vždy pouze jedna hodnota, nelze se vrátit zpě
  - Nebo data zaznamenaná – lze vrátit na začátek a provést nastavení na danou pozici
- Soubor na disku
  - manipulace (přístupy) řeší operační systém
  - zápis nebo čtení lze i zaráz (opatrně)

Problémy datových typů

- Přetečení
- Podtečení

- Výsledek je „mimo“
  - Např. sjednocení dvou intervalů mohou být dva intervaly
  - Jak poté uložit výsledek?

Vlastnosti proměnných

- Umístění v druhu/typu paměti:
  - Zásobník
  - Datová část programu
  - Registr
  - Externí
- Umístění v paměti (adresa)
- Viditelnost
  - Globální
  - Lokální pro modul nebo pro funkci
- Datový typ (vliv na velikost obsazené paměti)
- Hodnota

Dělení proměnných

- Statické
  - Velikost daná při překladu
  - Neměnný rozměr
- Dynamické
  - Mohou měnit velikost po definici

- Homogenní
  - Stejné prvky
  - Např. pole
- Heterogenní
  - Struktura s různými typy

Globální proměnné

- Znemožňují současného vícenásobného použití kódu
- Vadí nejenom při použití vláken
  - V jednom procesu program nastaví proměnnou
  - Další proces nastaví „svoji hodnotu“
  - První proces bude upravovat nesprávná data
- Použití pro nastavení společných vlastností
  - Inicializační hodnoty
  - Datum
  - Adresář
  - …

Padding

- Zhušťování
- Načítání z paměti je po více bitech (mocninách 2)
- Adresy proměnných by měly začínat na mocninách dvou
- Pokud nedodrženo – hodnoty nejsou načteny jednou operací, ale dvěma
- Proměnné (např. ve struktuře) jsou v paměti doplňovány výplní, aby ležely na správných pozicích
- Výhodnější definovat menší datové typy za menšími