Materiály na SZZ pro Informační Bezpečnost

Matěj Douša

Červen 2024

Obsah

1 Počítače a systémy

1.1 SP-19 (PA1)

Datové typy v programovacích jazycích. Staticky a dynamicky alokované proměnné, spojové seznamy. Modulární programování, procedury a funkce, vstupní a výstupní parametry. Překladač, linker, debugger.

Datové typy

V programovacích jazycích používáme proměnné, tedy něco, co uchovává datovou hodnotu s nějakou vnitřní strukturou. Proměnné jsou identifikovány svými jmény — identifikátory. Datový typ proměnné definuje vnitřní strukturu/reprezentaci dat a jejich význam. Tím určuje jakých hodnot může proměnná nabývat a také jaké operace lze s proměnnou (její hodnotou) vykonávat.

Jednoduché datové typy:

- celočíselné
  - existují různé délky — short, int, long (byte, long long, ...)
  - signed (znaménkové) — umí uložit i záporné hodnoty, používá se doplňkový kód
  - unsigned (neznaménkové) — ukládá jen kladné hodnoty, přímý kód
- s pohyblivou řádovou čárkou
  - existují různé délky — float, double, long double
  - znaménko (1 bit) + mantisa (velikost=přesnost) + exponent (velikost=rozsah)
- znakové

  Znaky jsou kódovány jako čísla, používá se ASCII / extended ASCII / UNICODE.
- logická hodnota

  Není v C, ale často se v jazycích vyskytuje (boolean — true/false).

Další datové typy:

- ukazatel (pointer)

  Adresy paměti, kde je uložen datový typ pointeru (pointer vždy ukazuje na konkrétní typ/funkci, případně void).
- výčtový typ (enum)
- struktura

  Je složena z dalších datových typů, klidně dalších struktur.
- union

  Ukládá více různých datových typů na stejné místo.
- třída (ve vyšších jazycích)

Statická a dynamická alokace

Staticky alokované proměnné:

- vzniknou běžnou deklarací
- ukládají se na zásobník (lokální proměnné) či do části .BSS (neinicializované globální proměnné) a .DATA (inicializované globální proměnné)
- v případě pole je nutno znát v době kompilace velikost (statická velikost)

Dynamicky alokované proměnné

- vzniknou použitím speciální funkcí/operátorem
- ukládají se na haldě (heap)
- přistupujeme přes pointer
- je možné alokovat paměť podle hodnot spočítaných za běhu programu

Spojové seznamy

- oproti poli nejsou položky seřazeny v paměti, ale každý prvek seznamu obsahuje ukazatel na další prvek.
- podobně jako v dynamicky alokovaném poli lze ukládat předem neznámý objem dat
- nelze jednoduše indexovat, ale lze libovolně přidávat či ubírat prvky z jakékoliv pozice v seznamu

Modulární programování

- složitější programy mohou být rozděleny do modulů
- tyto moduly lze použít v různých dalších částech programu
- modul má svou specifikační část (deklarace poskytovaných prostředků/rozhraní) a implementační část (definice/implementace poskytovaných prostředků)
- v C/C++ typicky hlavičkový soubor (.h/.hpp) a implementační soubor (.c/.cpp)

Procedury, funkce a parametry

- procedura/funkce je posloupnost příkazů uložených v paměti programu
  - procedura — bez návratové hodnoty (typ void)
  - funkce — s návratovou hodnotou
- použijeme ji zavoláním přes její jméno
- deklarace je specifikace jejího rozhraní — parametrů a typu návratové hodnoty
- definice je samotný kód funkce
- vstupní paramtery jsou informace, které využije kód funkce
- výstupní parametry jsou výsledkem běhu funkce — typicky se nějak změní a tím nám dají výsledek

Překladač

- překládá vyšší programovací jazyky do nižších
- ze zdrojového kódu vzniká objektový soubor — modul se strojovým kódem
- front-end přeloží konkrétní jazyk do vnitřní reprezentace (abstrakce nezávislá ani na platformě ani na jazyku)
- back-end přeloží vnitřní reprezentaci do strojového kódu konkrétní platformy

Linker

- spojuje přeložené moduly do výsledného celku — programu
- výstupem je spustitelný soubor

Debugger

- usnadňuje hledání chyb v kódu, také usnadňuje pochopení programu
- je vhodné kompilovat s informacemi pro ladění
- je možné si na nějakém místě běh programu zastavit a např. sledovat obsah proměnných, pouštět každý krok programu postupně...

1.2 SP-30 (SAP)

Kódy pro zobrazení čísel se znaménkem a realizace aritmetických operací (paralelní sčítačka/odčítačka, realizace aritmetických posuvů, dekodér, multiplexor, čítač). Reprezentace čísel v pohyblivé řádové čárce.

Čísla se znaménkem

Existuje několik možností, jak v počítači ukládat celá čísla:

- Přímý kód
  - první bit je znaménkový — určuje tedy, zda je hodnota za ním kladná či záporná
  - ostatní bity představují absolutní hodnotu čísla
  - existuj zde kladná i záporná nula
- Doplňkový kód
  - dle prvního bitu lze poznat znaménko čísla
  - převod kladné $\leftrightarrow$ záporné lze vysvětlit jako inverze bitů a následné přičtení jedničky
  - 0111 (7) $\rightarrow$ 1001 (-7)
  - není zde záporná nula
- Aditivní kód
  - uložené číslo je posunuto o nějakou konstantu, typicky polovina rozsahu
  - pro 4 bity určeme nulu jako 1000 — pak 1111 je 7, 0000 je -8
  - nula není zobrazena jako nula
  - není zde záporná nula

Čísla v pohyblivé řádové čárce

Reprezentace pohyblivé řádové čárky vychází ze zobrazení $A = M * z^e$ používaném např. ve fyzice, kde z je základ soustavy (zde 2), e je exponent jako celé číslo, M je mantisa.

- používá se normalizovaný tvar, tedy mantisa je zapsána tak, že ji nelze "posunout"více doleva.
- v přímém kódu mantisy je vlevo vždy jednička, která se skrývá (zvýšení přesnosti)
- pro mantisu se typicky používá přímý kód, pro exponent aditivní
- float (32b) typicky vypadá jako 1b znaménko, pak 8b exponent a nakonec 23b mantisa (tedy přesnost 24b)

Realizace aritmetických operací

- Paralelní sčítačka

  Tvořena více jednobitovými sčítačkami. Jednobitová sčítačka má 3 vstupy: A, B (sčítané bity) a vstupní přenos (carry — např. ze sčítačky nižšího řádu). Výstupy jsou S (výsledek) a výstupní přenos.
- Aritmetické posuvy

  Posun čísla vlevo/vpravo. Realizuje posuvný registr. Existuje více různých posuvů:
  - logický posuv — doplňuje nuly
  - cyklický posuv — doplňuje co vylezlo na druhé straně
  - aritmetický posuv — doplňuje 1 nebo 0 podle znaménka čísla
- Dekodér

  Kombinační logický obvod, který má méně bitů na vstupu než na výstupu, a podle tabulky převádí. Kodér má opačnou funkci.
- Multiplexor

  Na základě řídícího signálu vybere, který ze vstupů pošle na výstup (má několik vstupů + řídící vstup, a jeden výstup).
- Čítač

  Registr s funkcí inkrementu/dekrementu, může čítat nahoru a/nebo dolů. Existují úplné čítače (do mocnin 2) či neúplné (do jiných čísel). Typicky čítají v binárním kódu, lze i např. v Grayově kódu.

1.3 SP-28 (SAP)

2 Šifrování a sítě

3 Obecná bezpečnostní teorie

4 Matematika

5 Programování