

Názvosloví

➤ Procedura

- Procedura nebo funkce

➤ Call graph

- Graf (možného) volání mezi procedurami

➤ Základní blok (BB – basic block)

- Část procedury bez větvení a smyček, se vstupem pouze na začátku a výstupem pouze na konci
- Volání procedury může a nemusí být považováno za předěl BB

➤ Tok řízení - control-flow (graph)

- Možnosti předávání řízení mezi základními bloky v proceduře
- Reprezentováno orientovaným (cyklickým) grafem

➤ Tok dat - data-flow

- Předávání dat, obvykle uvnitř jednoho základního bloku
- Pro jeden BB může být reprezentováno dagem

➤ Závislost (dependence)

- Povinnost provést jednu operaci/instrukci po jiné
- Částečné uspořádání operací/instrukcí v jednom BB

❖ Datová závislost (dependence)

- Závislost producent-konzument v toku dat

❖ Antidependence

- Read-Write: Čtení se musí stihnout před zápisem
- Write-Write: Pořadí zápisů se nesmí změnit
- Jiné důvody, obvykle nízkoúrovňového původu

➤ Dag (directed acyclic graph)

- Orientovaný acyklický graf použitý pro zaznamenání
 - data-flow
 - závislostí

➤ **Skalární/jednoduchý/atomický typ (scalar)**

- Typ, s nímž dokáže přímo pracovat cílový stroj

➤ **Složený typ (aggregate)**

- Pole, struktura, třída, řetězec apod.

➤ **Zarovnání (alignment)**

- Požadavek na umístění proměnné/paměťového místa na adrese dělitelné 2, 4, 8, nebo 16
- Striktní: Při nedodržení procesor vyvolá výjimku
- Optimalizační: Při nedodržení bude kód pomalejší

➤ **Proměnná (variable)**

- Proměnná deklarovaná ve vstupním jazyce, včetně parametrů
- Pomocná proměnná (temporary) vytvořená překladačem

❖ **Statická/globální proměnná**

- Proměnná s jedinou instancí přístupná všem procedurám

❖ **(Lokální) proměnná**

- Parametr, deklarovaná či pomocná proměnná přístupná pouze jedné proceduře
- Jazyky s vnořenými procedurami vyžadují další kategorii proměnných přístupných z vnořených procedur

➤ **Paměťové místo**

- Část proměnné nebo dynamicky alokované paměti

➤ Alias

- Situace (nebo možnost výskytu situace), kdy k jedné proměnné, její části, či paměťovému místu, vedou dvě různé přístupové cesty

```
int x;  
int * p = & x;
```

```
int a[ 20];  
a[ i] = ...;  
a[ j] = ...;
```

- Rozhodnutí, zda může jít o alias, je obecně algoritmicky neřešitelná úloha
- Pokud si překladač není jist, že o alias nejde, musí se chovat, jako by to alias byl
 - Nejistý alias je ještě horší, než jistý

❖ Proměnná bez aliasu

- Lokální proměnná, která prokazatelně nemá alias
 - Všechny přístupy k ní lze jednoznačně určit

➤ **Doba života/rozsah platnosti proměnné (live range)**

- Množina míst v proceduře, kdy je proměnná zapotřebí
 - Tedy existuje možnost, že by ještě byla čtena (před zápisem)
- Zkoumá se obvykle pouze pro skalární lokální proměnné bez aliasu

❖ **Variable splitting/renaming**

- Proměnnou s nesouvislým rozsahem platnosti lze nahradit několika jinými
 - Odpadne omezení na shodnou alokaci v jednotlivých souvislých oblastích rozsahu platnosti

➤ **Statická alokace (static allocation)**

- Pro statické proměnné
- Vyhrazení místa na „pevné“ adrese
 - Podléhá relokaci při spojování linkerem a zavádění loaderem

➤ **Registrová alokace (register allocation)**

- Pro skalární lokální proměnné bez aliasu
- Umístění do fyzického registru cílového stroje
 - Pouze po dobu života proměnné
 - Omezeno počtem fyzických registrů

➤ **Zásobníková alokace (stack allocation)**

- Umístění na zásobníku
 - Zásobník může být definován procesorem nebo emulován
- Složené nebo aliasované lokální proměnné
- Proměnné, které se nevešly do registrů

❖ **Spill-code**

- Kód „navíc“, který musel být přidán pro manipulaci s proměnnými, které se nevešly do registrů

➤ Volací konvence

- Úmluva o způsobu spolupráce volajících a volané procedury
- Definována cílovým prostředím nebo autorem překladače
- Umístění parametrů
 - Zásobník nebo registry
 - Pořadí
 - Přesné umístění je komplikováno zarovnáním
- Umístění návratové hodnoty
 - Obvykle registr
 - Složené typy se obvykle řeší jako parametry předávané odkazem
- Odpovědnost za úklid zásobníku
 - Volající/volaný
- Povinnost zachovat obsah registrů
 - Všechny, některé, nebo žádné
- Další technické definice
 - Úprava jména procedury jako symbolu pro linker
- Pokročilá interprocedurální optimalizace: Automatická úprava volací konvence podle místních podmínek volaného a všech volajících

➤ Intraprocedurální

- Uvnitř jedné procedury

❖ Lokální

- Uvnitř jednoho základního bloku
 - Příklad: Jednodušší verze schedulingu nebo CSE

❖ Globální

- Pro celou proceduru najednou
 - Příklad: Přidělování registrů, složitější CSE

➤ Interprocedurální

- Pro celý program najednou
- Při separátním překladu modulů je součástí linkeru
- Obvykle exponenciální nebo algoritmicky neřešitelné úlohy
 - Příklad: Interprocedurální analýza aliasů
- Srovnání: In-line expanze procedury patří formálně mezi intraprocedurální optimalizace

❖ Logický typ

- Datový typ definovaný vstupním jazykem
- Základní typy + typové konstrukce
- Neomezená množina typů

❖ Fyzický typ

- Typ rozeznávaný cílovým strojem
- Konečná množina vestavěných typů
 - Celá čísla několika velikostí (znaménková a bezznaménková)
 - Ukazatel (pokud nesplývá s celým číslem)
 - Reálná čísla několika velikostí/přesností
- Na ostatní typy se pohlíží jako na posloupnost bajtů
 - Rozhoduje délka, případně požadavek na zarovnání