

PPA – BI-WSI-SI-13

Rozdělení paměti při implementaci programovacích jazyků: statické části, zásobník, halda. Aktivační záznamy, mechanismus implementace volání funkcí.

Obsah

1	Rozdělení paměti při implementaci programovacích jazyků	2
1.1	Statické objekty	2
1.2	Objekty na zásobníku	2
1.3	Halda	2
1.4	Aktivační záznam	2
1.5	Základní typy předávání parametrů funkcí	3
1.6	Mechanismus implementace volání funkcí	3

1 Rozdělení paměti při implementaci programovacích jazyků

Běh programu běží v souvislém logickém adresovém prostoru, který je poskytován operačním systémem. Operační systém mapuje logické adresy na fyzické adresy (které nemusí tvořit souvislý celek).

Kód programu
Statické objekty
Halda (Heap)
Volná paměť
Zásobník (Stack)

1.1 Statické objekty

- Globální konstanty, globální proměnné, data generované překladačem,...
- Výhody
 - Nepotřebuje správu paměti při běhu programu
 - Rychlý přístup (překládá se přímo adresou v paměti)
 - Není nebezpečí out-of-memory
- Nevýhody
 - Velikost a počty musí být známy předem

1.2 Objekty na zásobníku

- Každé volání funkce má svůj aktivační záznam (activation record). Podobně každý vnitřní blok.
- LIFO přístup podporuje volací a návratový mechanismus volání funkcí a procedur.
- Přirozená podpora volání funkcí a návratu z funkcí. Podpora rekurze.
- Objekty jsou vloženy na zásobník při vstupu do funkce a jsou ze zásobníku odebrány při návratu.
- Volání funkce je implementováno volací sekvencí, návrat je implementován návratovou sekvencí.

1.3 Halda

- Obsahuje dynamické objekty/proměnné vytvořené během běhu programu, např. pomocí příkazů malloc a free (C) nebo new a delete (C++).
- V některých programovacích jazycích je halda udržována pomocí garbage collection.

1.4 Aktivační záznam

- S každým voláním funkce se na zásobníku vytvoří aktivační záznam.
- Jazyky bez vnořených procedur, např. C
 - Lokální statické proměnné jsou uloženy v lokálním aktivačním záznamu na zásobníku.
- Jazyky s vnořenými procedurami, např. Pascal, také Lisp a funkcionální jazyky obecně, kde funkce typicky vytváří jinou funkci

- Link mezi aktivačními záznamy procedur (funkcí).

1.5 Základní typy předávání parametrů funkcí

- Hodnotou
 - Skutečná hodnota je vypočítána a následně zkopírována.
 - Vstupní parametr, chová se jako lokální proměnná.
- Odkazem
 - Volající určuje adresu v paměti.
 - Vstupní/výstupní proměnné.

1.6 Mechanismus implementace volání funkcí

Skutečné parametry
Návratové hodnoty
Řídící link (link na starý aktivační záznam)
Přístupový link (link na aktivační záznam vyšší procedury)
Uložené údaje počítače (registry, atd...)
Lokální proměnné
Pomocné dočasné proměnné

- Stack pointer SP - Odkaz na vrchol zásobníku
- Frame Pointer FP - Odkaz na začátek aktuálního záznamu