

INTERAKTIVNÍ PŘEKLADAČ PARAMETRICKÉHO PROGRAMOVÉHO SYSTÉMU PARAM/II

RNDr. Jaroslav Lizner, CSc.

Článek se pokouší o obecnou charakteristiku interaktivního překladače na základě jeho funkcí, o nástin jeho možné stavby s využitím existujících firemních softwarových produktů (pro práci se znakovými řetězci). Jako základ pro tyto úvahy slouží interaktivní překladač realizovaný v rámci parametrického programového systému PARAM/II, který je v tomto článku popsán.

Příspěvek volně navazuje na referát o parametrickém programovém systému PARAM/II, který byl přednesen Ing. Andrejem Boldišem a autorem tohoto článku na tomto semináři v minulém roce.

1. Úvod

Současná etapa vývoje aplikace výpočetní techniky je charakterizována snahou o zefektivnění analyticko-programovacích prací. Ve velké většině výpočetních středisek, převážně ve spojitosti s problematikou hromadného zpracování dat, vzniká celá řada obecněji použitelných "pomůcek" - od jednoduchých tiskových programů, přes náročnější parametrické programy až po ucelené parametrické programové systémy pokrývající komplexně řešenou problematiku.

Výhody budování takovýchto programů či celých systémů jsou zřejmé a obecně známé. V neposlední řadě je jí minimali-

zace rutinní, monotenní práce v této profesi, práce ovšem náročná na speciální znalosti řešitelů. Z tohoto hlediska je vhodné spojit budování takovýchto programů a systémů se snahou o tvorbu "komfortních" programů (v anglické terminologii "friendly", v německé "freundlich", český odborný termín není dosud pevně stanoven a je utvořen autorem) jako určitého protikladu k tzv. "intelligentním" programům. Řada těchto programů na straně jedné sice umožňuje uživateli mnoho obecných i speciálních možností, ale na straně druhé vyžaduje i náročnou odbornou přípravu jako podmínku pro jejich efektivní využití. V praxi jsou ovšem takovéto programy (ale obecněji i např. tzv. inteligentní terminály atd.) potom využívány neoptimálně, často pouze ve svých základních funkcích.

Neproti tomu, či lépe souběžně, je potřeba propagovat tvorbu takových programových systémů, které uživateli "ulehčují" práci s nimi, t.j. chovají se vůči němu přátelsky. Právě do této kategorie lze zařadit snahu o tvorbu interaktivního překladače, který byl vytvořen jako součást parametrického programového systému PARAM/II. Možnost opravy chyby prostým najetím kurzorem na zobrazený chybný překládaný příkaz či oprava celého zdrojového programu pomocí příkazů "inteligentního" editovacího programu a to vše v součinnosti s postupným překladem při jediném natažení zdrojového tvaru programu i překladače do virtuální paměti je práce právě onou "přátelskou" službou software jeho uživateli. Zvládnutí jeho užívání je přitom otázkou několika minut.

Při tvorbě zmíněných obecných parametrických programových systémů musíme mít na paměti i nároky na spotřebu strojového

času, t.j. musíme být vedeni snahou o minimalizaci času potřebného pro vykonávání, interpretaci tykových programů. Musíme tedy, vedle dalších zásad jako je volba programovacího jazyka atd., pracovat podle hesla "co je možno udělat jednou nelze provádět mnohokrát" (a to se týká nejen syntaktické kontroly parametrů, ale i výpočtů nejrůznějších adres atd.). Neměli bychom se tedy spokojit s oddělením části analýzy parametrů od vlastního vykonávání, interpretace programu, ale je vhodné zavést vedle zdrojové formy parametrů (zdrojového programu) i tzv. tvar interní, cílový (cílový program), t.j. orientovat se na tvorbu interpretačních programů. Jinými slovy, snaha o zefektivnění interpretace programu nás přivede k nutnosti konstruovat překladač.

Vhodnou formou komfortního překladače může být právě překladač interaktivní. Zjednodušením problému je konstrukce překladače jednopřechodového a tato cesta byla zvolena též při konstrukci popisovaného překladače.

Účelnost interaktivního překladače spočívá tedy zejména v následujícím:

- v minimalizaci doby potřebné pro odstranění triviálních chyb (PROMPTING)
- v ulehčení procesu oprav
- v úspoře strojového času.

Nelze ovšem na závěr neuvést, že sám překladač je na procesu interakce "nezávislý", vyžaduje pouze některé specifické principy, které je nutno v procesu projekce a realizace překladače zohlednit.

2. Předpoklady konstrukce interaktivního překladače

Základním předpokladem konstrukce interaktivního překladače je zjevně vhodný operační systém s terminálovými pracovišti. Jazyk překládaného programu musí mít specifikovanou elementární jednotku jazyka, pokud možno co "nejmenší" (tzv. příkaz či vhodnější obecnější termín formule). Zřejmě velmi obtížně by se realizoval interaktivní překladač pro jazyk s blokovou strukturou.

Dále předpokládáme existenci dialogového programu pro práci se znakovými řetězci, uloženými ve virtuální paměti (na př. programy EDT, využitý v prezentovaném interaktivním překladači, či EDOR fy Siemens, případně obdobné programy jiných výrobců výpočetní techniky), tzv. EDITOR. Toto není samozřejmě požadavek nezbytný, ovšem podstatně zjednodušuje proces přípravy interaktivního překladače.

Specifickým požadavkem je zde taková reentrantnost modulů provádějících syntaktickou analýzu jednotlivých formulí zdrojového jazyka, která zaručí opakované startování překladače. V případě, že se toto nepodaří u všech modulů dodržet, je zde ještě možnost "výpomoci si" při opakovaném startování tzv. dynamickým linkováním.

3. Charakteristika interaktivního překladače

Interaktivní překladač lze charakterizovat jeho pěti základními funkcemi :

- globální syntaktická kontrola
- syntaktická kontrola formule

- zobrazení chyby a interaktivní oprava formule (formulí) zdrojového jazyka
- generování interního (cílového) tvaru formulí
- vedení protokolu o překladu.

Z hlediska konstrukce interaktivního překladače je podstatné rozčlenění testů korektnosti zápisu zdrojového jazyka na testy globální syntaxe a na testy korektnosti formule. Ve vztahu k tomuto rozdělení jsou potom dvě základní akce interaktivního překladače :

- opakovaný překlad formule, u níž byla zjištěna chyba (RESUME)
- opakovaný překlad od první formule.

Pod globální chybou (t.j. chybou zjištěnou modulem globální syntaktické kontroly) rozumíme zde chybu ve formulí jazyka ve vztahu k jiným, již dříve přeloženým formulím (např. pořadí formulí, přípustnost formule v dané situaci, úplnost souboru formulí atd).

Problém je zjevně zjednodušen pro konstrukci jednopráchodového překladače; jinak je obtížnější určit, kdy možno překlad analyzované formule opakovat a kdy je nutno se vrátit a kam.

Se základním schématem interaktivního překladače PARAM/II budou posluchači seznámeni při přednášce.

4. Popis interakce překladače

Podstatou interakce překladače je nabídka několika možností (akcí) v klíčových stavech (situacích) zpracování s tím, že uživatel volí jednu z nich. Volbu akce uživatelem lze vhodně kombinovat s omezením nabídky možných akcí programem, v některých

případech potom tuto volbu provádět přímo programem.

Přehled možných akcí interaktivního překladáče :

0. Volba překládaného programu

(interaktivní překladáč umožňuje překládat více zdrojových programů při jednom volání překladáče)

1. Příímá oprava formule

(v případě opravy právě překládané formule není nutné užívat dialogový program pro práci s texty, opravu možno provést přímo, zásahem do zobrazené formule)

2. Oprava formulí pomocí Editoru

(zahrnuje též možnost opravy formulí předcházejících či následujících formulí analyzované, popřípadě rušení či vkládání formulí)

3. Pokračování v překladu od formule, u níž byla indikována chyba

(právě zde je důležitá reentrantnost modulu globální syntaxe, umožňující opakovaný průchod tímto modulem - tím je umožněno, že i při zrušení opravované formule, vložení nové, či při změně formule, lze tuto akci volit)

4. Startování (opakování) překladu programu od první formule

5. Ukončení interaktivního překladu

(překlad nadále pokračuje standardním způsobem bez zobrazování chyb a bez jejich oprav)

6. Přepis zdrojového programu jeho opravenou verzí

(jiným možným řešením by bylo např. založení souboru s vyšším číslem verze, resp. generace, ovšem přepis zdrojové formy programu je možno překládat se postup základní, v případě jiného postupu možno s výhodou využít možností Editoru)

7. Ukončení práce interaktivního překladáče

Se zřetelím na komfort budovaného překladače jsou v interaktivním překladači PARAM/II zabudovány ještě následující dvě akce :

- přechod do systémového modu
 - možnost zobrazení návodu pro práci s interaktivním překladačem, vhodně segmentovaného dla jednotlivých klíčových stavů.
- Pro úplnost akcí je možno uvést i příkaz pro návrat z dialogového programu zpět do řídicího modulu překladače a příkaz pro návrat ze systémového modu; to jsou ovšem standardní příkazy těchto programů.

Vedle těchto akcí má interaktivní překladač i další standardní akce, jichž se ovšem proces interakce nedotýká (uživatel je přímo neovlivňuje), jako je tvorba protokolu o překladu, či generování cílového tvaru programu atd.

V interaktivním překladači PARAM/II je zavedena dále tzv. "nulová akce", t.j. odpověď uživatele nulové délky (s využitím terminálové klávesy DU). Uvádí-li uživatel tohoto překladače pouze odpovědi nulové délky chová se interaktivní překladač jako dávkový (batchový). Jinak je možné užívat odpověď nulové délky pro volbu akce v dané situaci považované za základní, či jako výraz odmítnutí nabízené možnosti (např. přepis zdrojového tvaru programu po provedení překladu).

Klíčové stavy (situace) interaktivního překladače.

- A. Po natažení interaktivního překladače, resp. po ukončení práci na předcházejícím programu
(zde uživatel volí překládáný program, respektive ukončení práce interaktivního překladače)
- B. Po načtení překládaného zdrojového programu do virtuální paměti

(zde uživatel může zvolit úpravu zdrojového tvaru programu před započítáním překladu či překlad programu od první formule)

C. Po zjištění chyby

(uživatel volí buď přímou opravu příkazu či opravu příkazu pomocí Editoru, případně ukončení interaktivního překladu)

D. Po ukončení oprav

(zde uživatel volí mezi pokračováním v překladu od formule v níž byla indikována chyba a mezi opakováním překladu od první formule)

E. Po ukončení překladu programu

(zde uživatel může volit předpis zdrojového programu opravenou verzí, či práci s dialogovým programem pro práci s texty).

Z hlediska užívání interaktivního překladače je kritické místo ve správném rozhodování ve stavu D. Nesprávná volba pokračování kompilace od příkazu v němž byla zjištěna chyba vede, vzhledem k předchozím úpravám, k nekorektnímu chování kompilátoru. Řešit tento problém přímo v překladači je ovšem velmi obtížné.

5. Závěr

Konstrukce interaktivního překladače s použitím uvedených principů je reálná a není nikterak časově náročná. Přesnost vybudování interaktivního překladače v PARAM/II nepřesáhla 2 měsíce (ovšem při existenci překladače dávkového). V naší praxi se interaktivní překladač osvědčil a lze jej tedy všem tvůrcům parametrických programových systémů doporučit.