

PARAMETRICKÉ PROGRAMY PRO MANIPULACI S DATY

Ing. Jiří Volák

Příspěvek popisuje kolekci parametrických programů pro nejčastěji se vyskytující funkce ve zpracování hromadných dat. Kolekci tvoří 6 parametrických programů SELECT, KUMUL, SKUMUL, RESTRUKT, ACT a EDT. Programy pracují se sekvenčními soubory pod operačními systémy DOS ASVT (M-4030) a BS-1000 (Siemens 4004 a 7000).

1. Úvod

Ke vzniku kolekce parametrických programů, které jsou tématem tohoto příspěvku vedly následující úvahy:

Ve zpracování dat se s různými obměnami opakují některé funkce, jejichž logika je v zásadě vždy stejná. Jsou to zejména:

- výběr vět podle výběrových kritérií
- kumulace údajů ve větách se stejnou hodnotou řídících klíčů
- změna struktury, přeskupení dat
- zatřídění sekvenčních souborů, aktualizace kmenového souboru
- třídění dat
- provádění aritmetických operací
- tisk sestav, zejména součtových

Podají-li se tyto funkce vhodným způsobem zobecnit formou parametrických programů, vznikne aparát, který je schopen převzít značnou část úlohy zpracování dat.

Vyjmenované (a ařejmě ještě další) funkce se vyskytnou převážně v okrajových částech úlohy, tj. na vstupech a výstupech. Centrální část (jádro) bývá již pro danou úlohu specifická a realizuje se v některém programovacím jazyku.

2. Metoda realizace

Všechny parametrické programy z kolekce jsou neprocedurální, tj. uživatel pouze zadává prostřednictvím pseudopříkazů, co má být výsledkem funkce. Vlastní algoritmus je již záležitostí parametrického programu. Parametrické programy jsou vytvořeny jako překladače, tj. každý pseudopříkaz je převeden do úseku strojo-

vých instrukcí, který se pro daný běh programu stává jeho součástí. Tím je dosaženo vysoké operační rychlosti, srovnatelné s dobře napsanými programy v assembleru.

3. Omezení

Operační prostředí	DOS ASVT (M-4030), BS-1000 (Siemens)
Organizace souborů	Sekvenční se standardním návěštím, MF nebo disk, fixní nebo var. věty
Délka bloku	max. 4000 bytů
Délka věty	max. 1000 bytů
Zobrazení údaje	znakově (X) nebo zhuštěně (P)
Max. počet pseudopříkazů	cca 100

4. Obsluha parametrických programů

Parametrické programy jsou řízeny pseudopříkazy. Každý pseudopříkaz se děruje do samostatného děrného štítku. Na začátku štítku děrujeme dvě lomítka a mezeru (podobně jako u jobových štítků), následuje klíčové slovo, jedna mezera a za ní poziční parametry, které danou instrukci upřesňují. Za posledním parametrem je jedna mezera, za níž mohou následovat libovolné poznámky až do konce štítku. Pokračovací štítky nejsou povoleny.

// operace $par_1, par_2, \dots, par_n$ poznámky

Symbolika pozičních parametrů:

- r - relativní adresa údaje ve větě (první byte má $r=0$ u fixních i variabilních vět)
- l - délka údaje v bytech; u dekadických zhuštěných údajů předchází údají délky písmeno "P" (packed)
- m - maska pro tisk, podobně jako v cobolu (např. ZBZZ9V9)
- t_1 - text do 3. řádku záhlaví nad údajem (EDT)
- t_2 - text do 4. řádku záhlaví nad údajem (EDT)
- b - délka bloku v bytech
- v - délka věty v bytech
- i - relativní pozice v tiskovém řádku nebo výstupní větě
- h_1 - hodnota pole
- d - distance, počet mezer mezi tiskovými poli (default=2)

Popis základních pseudopříkazů:

- // KEY r,l,text Poloha a délka řídicího (trždicího, zatřizovacího) klíče ve vstupní větě, text k příslušné úrovni součtů (EDT)
- // RECORD r,l,h₁,h₂, ... h_n
Výběr věty určitého typu. Věta se vybere, má-li specifikované pole některou z hodnot h_i. Je povoleno zadat i rozmezí h_m-h_n.
- // SELECT r,l,h₁,h₂, ... h_n
Doplňující výběrové kritérium. Parametry a jejich význam je stejný jako u příkazu // RECORD. Příkaz SELECT smí být jen za příkazem RECORD nebo SELECT, počet příkazů není omezen. Příkazy RECORD a SELECT jsou spojeny logickou spojkou AND. Je-li soustava podmínek RECORD/SELECT splněna, přechází se na následující pseudopříkaz, jinak se vyhledá nejbližší následující příkaz RECORD.
- // MASK m
Explicitní zadání tiskové masky. Masky je platná pro všechny následující příkazy FIELD, SPIELD a EDIT až po další instrukci MASK.
- // FIELD r,l,m,t₁,t₂
Poloha, délka, maska pro tisk a texty do záhlaví pro tiskové pole nescoučtované. Nemá-li zadána maska, platí maska z předcházející specifikace MASK.
- // SPIELD r,l,m,t₁,t₂ Totéž pro součtovaná tisková pole
- // DISTANCE d Stanovení počtu mezer mezi tiskovými poli
- // SET i Nastavení ukazatele do tiskové pozice i.
Příkazy FIELD, SPIELD, EDIT nastavují automaticky $i = i + 1 + d$
- // VARIB b,v Vstupní soubor variable (default)
- // FIXIN b,v Vstupní soubor fix
- // VAROUT b,v Výstupní soubor variable (default)
- // FIXOUT b,v Výstupní soubor fix

// LOAD r,1	Naplnění aritmetického registru
// ADD r,1	Přičtení k aritmetickému registru
// SUB r,1	Odečtení od aritmetického registru
// MULT r,1	Násobení aritmetického registru
// DIV r,1	Dělení aritmetického registru. Výsledek dělení nulou je nula. Chybí-li u aritmetické operace údaj délky <u>l</u> , chápe se <u>r</u> jako konstanta (např. // MULT 10 značí násobení deseti).
// PRINT	Tisk připravené řádky
// WRITE	Zápis věty do výstupního souboru
// EXIT	Přechod na čtení další věty
// DELETE	Vypuštění kmenové věty (ACT)
// INSERT	Vložení změnové věty (ACT)
// REPLACE	Výměna kmenové věty změnovou (ACT)
// COPY člen	Parametry jsou ve zdrojové knihovně pod názvem "člen"

5. Parametrické programy

Všechny parametrické programy z kolekce se vyznačují těmito společnými rysy:

- obsahují funkci RECORD/SELECT (program SELECT jen tuto funkci)
 - parametry mohou být uloženy na zdrojové knihovně (COPY)
 - programy mají jeden vstup a jeden výstup (s výjimkou ACT)
- Ostatní funkce jsou již specifické pro daný parametrický program.

5.1. Program SELECT

Program je určen pro výběr vět. Výběr se provádí na základě specifikace v příkazech RECORD a SELECT. Vybrané věty se zapisují do výstupního souboru v původní nebo specifikované délce (FIXOUT, VAROUT).

Parametry:

1. Organizace, blokování (FIXIN, VARIN, FIXOUT, VAROUT)
2. Výběr věty (RECORD, SELECT)

Příklad: žádný.

5.2. Program KUMUL

Provádí kumulaci vět o stejných řídicích klíčích. Kumulace probíhá tak, že specifikovaná součtová pole se sčítají do odpovídajících polí v součtové větě, která má stejnou strukturu jako věta původní. Součtová věta se vydává na výstup při změně některého z řídicích klíčů. Ostatní (nesoučtová) pole se do výstupní věty převezmou z poslední věty, která má ještě stejnou hodnotu řídicích klíčů. Pokud by došlo k přeplnění některého součtového pole, zapíše se tzv. přetoková věta, takže na výstupu budou dvě nebo více vět se stejnou hodnotou řídicích klíčů.

Parametry:

1. Organizace, blokování (FIXIN, VARIN, FIXOUT, VAROUT)
2. Výběr věty (RECORD, SELECT)
3. Klíče pro kumulaci (KEY)
4. Specifikace součtových polí (SFIELD)

Poznámka: Nepoužijeme-li ani jeden příkaz SFIELD, provede se pouze eliminace duplicitních vět a na výstupu bude vždy jen poslední.

Příklad:

```
// OPTION PARAM
// EXEC KUMUL
// KEY 9,6      číslo záznamu
// KEY 111,6    osobní číslo
// KEY 66,2     typ mzdy
// COPY KUMUL27 popis kumulovaných polí
// SFIELD 302,9 mzdový fond
// SFIELD 309,8 výkon pro mzdu
// END
```

Z vět, které následují za sebou a mají stejné číslo záznamu, osobní číslo a typ mzdy se vytvoří jediné výstupní věta, která ve specifikovaných polích SFIELD obsahuje součty za všechny věty.

5.3. Program SKUMUL

Provádí třídění a kumulaci vět souboru. Třídění se provádí interně vyvolanou procedurou systému SORT. Ve vstupní i výstupní fázi se provádí kumulace vět, které mají shodné řídící klíče. Podle stejných klíčů se provádí i třídění. Specifikovaná součtová pole se sčítají do součtové věty, která má stejnou strukturu jako vstupní věta. Součtová věta se zapisuje na výstup při změně některého z řídících klíčů. Při přeplnění některého součtového pole se zapisuje přetečková věta.

Parametry:

1. Organizace, blokování (FIXIN, VARIN, FIXOUT, VAROUT)
2. Výběr věty (RECORD, SELECT)
3. Klíče pro kumulaci a třídění (KEY)
4. Klíče jen pro třídění (SKEY)
5. Součtová pole (SFIELD)

Poznámka: Neuvedeme-li v parametrech ani jediný příkaz SFIELD, provede se jen třídění s eliminací duplicitních vět - na výstup se zapíše jen poslední věta shodných třídících klíčů. Uvedeme-li pouze příkazy SKEY a žádný KEY, provede se čisté třídění a ke kumulaci nedojde. Program SKUMUL je výhodně používat i místo parametrického SORTu, protože má jednodušší zadání a zachovává pořadí vět se shodnými třídícími klíči.

Příklad: Stejný jako v kap. 5.2 - tak bychom postupovali v případě, že by soubor měl být předem vhodně seřazen.

5.4. Program RESTRUKT

Změna struktury souboru, přeskupení dat. Výstupní větu je možno vytvořit v libovolné struktuře z libovolného počtu vstupních vět. Naopak z jedné vstupní věty je možno vytvořit libovolný počet výstupních vět v libovolné struktuře.

Parametry:

1. Organizace, blokování (FIXIN, VARIN, FIXOUT, VAROUT)
2. Výběr věty (RECORD, SELECT)
3. Příprava výstupní věty (SET, FIELD, STORE)
4. Aritmetické operace (LOAD, ADD, SUB, MULT, DIV)
5. Řízení vstupu a výstupu (EXIT, WRITE)

5.5. Program ACT

Aktualizace kmenového souboru jedním změnovým souborem.

Na základě parametrů se provádí porovnání řídicích klíčů obou vstupních souborů a podle výsledku porovnání se provede vlastní aktualizace, tj.:

- vložení nové věty (INSERT)
- zrušení existující věty (DELETE)
- výměna existující věty (REPLACE)
- oprava existující věty (REPLACE r,1)
- tisk protokolu o provedené aktualizaci (EDIT,PRINT)

Oba vstupní soubory musí být seřazeny podle zadaných řídicích klíčů a musí mít shodnou strukturu.

Parametry:

1. Organizace, blokování (FIXIN,VARIN,FIXOUT,VARCUT)
2. Výběr změnové věty (RECORD,SELECT)
3. Vlastní aktualizace (INSERT,REPLACE,DELETE)
4. Tisk protokolu o aktualizaci (EDIT,PRINT)

Příklad:

```
// OPTION PARAM
// EXEC ACT
// KEY 10,5      číslo záznamu
// KEY 20,2      číslo řádku
// RECORD 0,3 001  typ věty 001
// INSERT                vložit
// RECORD 0,3,002      typ věty 002
// DELETE                vypustit kmenovou větu
// RECORD 0,3,003      typ věty 003
// REPLACE              vyměnit celou větu
// RECORD 0,3,004      typ věty 004
// REPLACE 30,6        vyměnit údaj v kmenové větě
// END
```

5.6. Program EDT

Vytvoření součtové sestavy. EDT produkuje tisky typu opis, výběr, seznam, součtová sestava s možností provádění jednoduchých aritmetických operací. Sestava se tvoří z jediného vstupního souboru. EDT je z popísané kolekce klíčovým programem. V podstatě umožňuje vytisknout téměř každou sestavu s minimální programátorskou námahou. Sestavu není třeba kreslit do tiskových formulářů, za normálních okolností si tiskové sloupce odpočítá EDT sám.

Parametry:

1. Organizace, blokování (FIXIN,VARIN)
2. Uživatelské texty do záhlaví (TITLE,HEADER,SET,FIELD)
3. Výběr věty (RECORD,SELECT)
4. Sledování změn řídicích klíčů, editace součtů (KEY,PKEY)
5. Generování tiskových řádků (SET,FIELD,SFIELD,EDIT,DISTANCE)
6. Provádění aritmetických operací (LOAD,ADD,SUB,MULT,DIV)
7. Řízení tisku (PRINT,CLEAR,DROP,EXIT)

Příklad:

```
// OPTION PARAM
// EXEC EDT
// DISTANCE 6
// TITLE REKAPITULACE PODLE ZAVODU A TYPU MEZD
// KEY 137,2,CELKEM ZAVOD součty za závod + totál
// FIELD 137,2,99,ZAV
// FIELD 68,3,2,TYP,MZDY
// SFIELD 71,4,ZBZZ9V9,POCET,HODIN
// MASK ZBZZBZZ9V99
// SFIELD 197,7,,ZAKLADNI,MZDA
// SFIELD 239,7,,HRUBA,MZDA
// SFIELD 204,7,,PREMIE
// SFIELD 253,7,,MZDA,CELKEM
// LOAD 253,7 mzda celkem
// MULT 10 x 10
// DIV 71,4 : počtem hodin
// EDIT 7,,HODINOVY,PRUMER
// PRINT
// END
```


6. Závěr

V závěru bych chtěl zdůraznit přednosti v používání parametrických programů. Jsou to zejména:

- snadné vytvoření manipulačních a tiskových procedur
- unifikace standardních funkcí v rámci střediska
- snadná údržba úloh, řešených parametrickými programy
- odbourání kompilací uživatelských programů
- úspora knihoven operačního systému
- vysoká operační rychlost parametrických programů

Z uvedeného výčtu je zřejmé, že parametrické programy přinášejí úspory v řešitelské oblasti i v provozu počítače. Parametrické programy však ovlivňují i oblast analýzy a navrhování projektů. Analytický návrh projektu, předpokládající maximální využití parametrických programů bude dosti odlišný od projektu programovaného klasicky, tj. v některém z programovacích jazyků. Je třeba vhodně navrhnout strukturu souborů, aby použití parametrických programů bylo co nejhezčí a nejjednodušší. Předpokládáme-li např. kumulaci vět, je vhodné nechat dostatečnou rezervu v rozsahu součtových polí, aby nevznikaly přetokové věty, které snižují přehlednost tiskových sestav. Řadu funkcí (zatřídění, výběr, tisk) je možno z klasických programů vyčlenit a provádět je parametricky.