

Ing. Zdeněk Tomka

Ústav vývoje a racionalizace železničního  
stavebnictví, Brno

## SROVNÁNÍ PROGRESIVNÍCH METOD VYTVÁŘENÍ TISKOVÝCH SESTAV

V příspěvku jsou porovnány tyto metody vytváření tiskových sestav:

- užití obecných tiskových parametrických programů
- užití REPORT-WRITERu jazyka COBOL
- využití normovaného programování a jeho generátoru.

Srovnání je provedeno z hlediska:

- pracnosti a srozumitelnosti kódování,
- rychlosti odladění,
- potřebě operační paměti.

Sestavy byly odladěny na počítači EC-1033 v operačním systému OS.

### 1. ÚVOD.

Mezi nejracionalnější metody vytváření tiskových sestav patří:

- využití obecných parametrických tiskových programů
- užití cobolského REPORT-WRITERu
- využití normovaného programování a generátoru RP
- kódování v jazyku RPG.

Jelikož se ve výpočetních střediscích aplikuje většinou jeden hlavní programovací jazyk, v našem případě COBOL, nebude se sestavami vytvořenými v jazyku RPG zabývat.

Pro posouzení vhodnosti uvedených metod si rozdělíme tiskové sestavy do 3 skupin:

- sestava je vytvářena z jednoho vstupního souboru
- sestava je vytvářena z více vstupních souborů, kde výběr vět se provádí dle srovnání hodnot klíčů, podle kterých jsou soubory seřídány
- sestavy se speciálními požadavky.

## 2. Sestava vytvořena z jednoho vstupního souboru.

U tohoto typu úloh provedeme srovnání vytváření sestav z jednoho vstupního, seříděného souboru a řešení řídících změn dle klíčů metodou:

- normovaného programování
- normovaného programování s generátorem MP
- cobolského REPORT-WRITERu
- obecného parametrického programu OPPT
- obecného parametrického programu T105. *veřejl. CULPRIT*

Srovnání bylo provedeno na jednoduchém zadání běžné úlohy hromadného zpracování dat.

Zadání: Vstupní soubor je na MP a je seříděn dle klíčů:

- podnik

- závod

- středisko a má cobolevskou strukturu :

02	PODNIK	PIC	X(2).
02	ZAVOD	PIC	X(3).
02	STREDISKO	PIC	X(5).
02	MATERIAL	PIC	X(8).
02	POCET-KUSU	PIC	9(3).
02	CENA	PIC	9(6)V9(2).

Máme tisknout položkový výpis souboru se sumami za středisko, závod, podnik a celkovou sumou. Tisk položek pro jiný podnik začíná na jiné straně. Sumujeme údaj KCS.  $KCS = CENA \times POCET-KUSU$ .

Sestava má tvar:

SESTAVA: VÝDAJE ZA MATERIAL  
 XXXXXXXXXXXXXXX

POD ZAVOD STREDISKO MATERIAL POCET-KUSU CENA SUMA  
 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

.. ... ..  
 x SUMA STREDISKO : .....  
 xx SUMA ZAVOD : .....  
 xxx SUMA PODNIK : .....  
 xxxx SUMA CELKEM : .....

Tabulka výsledků po odzkoušení jednotlivých metod:

Způsob kódování	Počet kódovaných DS	Počet procedurálních příkazů	Spotřeba paměti v B	Počet vygenerovaných řádků	
				celkem	proced. část
Normované programování	170	66	4536	-	-
Norm. progr. s generátorem NP	157	121	5101	162	67
Cebelovský REPORT-WRITER	84	10	6080		
Obecný par. progr. OPPT	14				
Obecný par. progr. T105	24				

analog. RPG,  
 (včetně leptů)

## Hlediska hodnocení vhodnosti metod vytváření sestav.

Faktory vhodnosti a efektivity metod vytváření sestav:

1. Pracnost kódování
  - a) počet nutných kódovaných DŠ
  - b) jednoduchost kódování.
2. Rychlost odladění.
3. Opravitelnost a přispůsobitelnost textu.
4. Srozumitelnost a dokumentárnost.
5. Rychlost výpočtu programu.
6. Spotřeba operační paměti.

Jedním z nejpálčivějších problémů v současné době ve výpočetní technice je otázka programátorských kapacit a fluktua-  
ce. Současně jsou kladeny stále větší nároky na se možno  
nejvčasnější vytvoření programových celků a výstupů z počítače. Uživatel se také dožaduje častých změn.

Musíme proto vytvářet programy srozumitelné, lehce odla-  
ditelné a opravitelné. Programy se musí kódovat jednoduchým  
způsobem tak, aby kódování zvládl i začátečník.

Proto nejvýznamnějšími faktorem při posouzení efektivity vy-  
tváření sestav se jeví počet kódovaných děrných štítků tex-  
tu programu nebo parametrů obecných programů. S tímto fak-  
torem souvisí přibližně přímo úměrně pracnost vytvoření pro-  
gramu, počet nutných pokusů při odladění, jednoduchost kó-  
dování a opravitelnost. Srozumitelnost obolovských textů je  
velmi dobrá zvláště při dodržení metodiky NP. Srozumitelnost  
parametrických programů závisí na jejich znalosti. Text pa-  
rametrů není samodokumentující.

Dalším faktorem k posouzení vhodnosti způsobu vytváření  
sestav je rychlost výpočtu programu. Při malých souborech se  
rozdíl rychlostí mezi různými způsoby zpracová-  
ní neprojevuje. Pro posouzení by se musel provést výpočet  
s velkými soubory. Na dnešních rychlých počítačích nebude

tento rozdíl v rychlosti mít velký význam.

Jako poslední hledisko pro hodnocení způsobu vytváření sestav je spotřeba operační paměti. Výsledky jsou uvedeny v tabulce. U velkých počítačů 3. generace je však i toto hledisko bez většího významu.

#### Peřadí vhodnosti metod pro tisk sestav:

##### 1. Nejvýhodnější se jeví použití obecných parametrických programů ( OPPT, T105, CULPRIT-IDMS atd.)

Počet kódovaných DŠ byl u OPPT 14, u T105 24. Zápis je velmi srozumitelný, sestavu může vytvořit po krátkém seznámení s užitím parametrů i neprogramátor. Odpadá kompilace a sestavení. Odladění sestavy je jednoduché. Opravy se provádějí snadno. Programátor by měl mít snahu, pokud takto vytvořená sestava uživateli vyhovuje, použít především této metody. Parametrické programy nebývají však zcela univerzální a mají některé nedostatky. Např. u OPPT musíme zadávat datum z parametrického DŠ. Sestavy nelze zcela obecně volit, některé tisky jsou dané programem.

##### 2. Druhým nejefektivnějším prostředkem je REPORT-WRITER jazyka COBOL.

Počet kódovaných DŠ byl v našem případě 84. Počet příkazů v procedurální části je minimální, v našem případě 10. Program je tedy velmi srozumitelný. Kódování je snadné, srozumitelné, lehce opravitelné a odladitelné. Při odladění potřebujeme 1 až 2 pokusy navíc oproti použití obecných parametrických programů. Report má zabudováno množství automatických funkcí nutných při vytváření sestav, které by se jinak musely složitě kódovat a může vytvářet velmi efektivně i ty nejsložitější sestavy. Složitější sestavy, které nemůžeme vytvářet pomocí obecných parametrických programů by se

měly vytvářet cobolovským Reportem.

3. Třetím nejvýhodnějším prostředkem je vytváření sestav pomocí metodiky normovaného programování a využitím generátoru NP.

Počet kódovaných řádků programu byl 157, počet vygenerovaných řádků 162, počet kódovaných příkazů v procedurální části byl 53. Text je srozumitelný a lehce opravitelný díky jednotné řídicí složce programu a vygenerovaných komentářových zápisů.

Tato metoda by se měla uplatňovat tam, kde nevyhovuje obec. param. program a neaplikuje se jazyk COBOL.

Uplatnění je také pro další typy sestav, viz dále.

4. Další způsob je uplatnění metodiky NP bez generátoru.

Počet kódovaných DŠ byl 170, počet příkazů 66. Uplatnění je jako u bodu 3, kde však není generátor k dispozici.

5. Kódování pro vytváření sestav normálním způsobem (v jazyku COBOL, PL1 atd.) lze snad uplatnit pro jednoduché sestavy, které neřeší řídicí směny klíčů a zpracování více souborů s výběrem vět dle klíčů.

### 3. Sestava vytvořená z více vstupních souborů.

V úlohách, kde se zpracovává více než 1 soubor a výběr vět se provádí dle hodnoty klíčů je vhodné řešit program jedním ze dvou způsobů:

- Pomocí obec. param. programů, pokud takto vytvořená sestava uživateli vyhovuje a program zvládne složitost zadání. Program OPPT je schopen zpracovat jen dva vstupní soubory.
- Pomocí metodiky normovaného programování a jeho generátorem a REPORT-WRITERu.

Generátor NP generuje část řídicího programu pro výběr vět (blok C). Pro vytváření vlastní sestavy je pak vhod-

*(RPG by byl na 2. místě hned po param. prog.)*

né použít cobolevského Reportu, který obhospodaruje zpracování změn klíčů a další funkce nutné k vytváření sestav. Vidíme tedy, že je možné také v normovaném programu užít REPORT-WRITER a vytvořit tak velmi jednoduchý a srozumitelný program. V normovaném programu pak vypustíme blok D (blok kontrol změn klíčů), který je automaticky prováděn v REPORT-WRITERu.

#### 4. Speciální požadavky na tisk sestav.

Existují speciální požadavky na vytváření sestav, při kterých není možno aplikovat ani obecné parametrické tiskové programy, ani REPORT-WRITER jazyka COBOL. V těchto případech lze však většinou s výhodou použít metodiky normovaného programování a generátoru NP. Dále je uveden příklad takového použití.

Mějme tento požadavek na sestavu. Velikost tištěné věty souboru je malá a na sestavu se vejdou dva řádky různých vět. Ušetří se tak téměř polovina papíru. Věty však musí být vzestupně dle klíče zařazeny pod sebou a až naplníme první polovinu stránky, naplňujeme druhou polovinu. Zápisy jednotlivých vět musíme uchovávat v tabulce paměti počítače, kterou nemůžeme REPORT-WRITERem naplňovat. Rutiny při změnách klíčů obhospodaruji vygenerované části normovaného programu, místo automatické funkce Reportu.