

RACIONALIZACE PROGRAMÁTORSKÝCH PRACÍ

Každé výpočtové středisko se dostává při budování informačního systému do situací, kdy se značně projevuje nedostatek programátorské kapacity. Proto jsou stále aktuální tendence racionalizovat práce programátorů, a tím šetřit jejich kapacity. Používané cesty k dosažení těchto cílů jsou různé. Někde je to maximální využívání základního software počítače, především utilitních programů, jinde to jsou metody normového a strukturního programování nebo jiné cesty. Jednou z cest dosud málo využívaných je vytváření universálních programů, které jsou schopny zajišťovat nejčastěji se vyskytující funkce uživatelských programů. Předmětem tohoto příspěvku je popsat principy tvorby a využívání takovýchto programů v našem výpočtovém středisku v AZNP Mladá Boleslav, kde je instalován počítač IBM 360/30 s kapacitou vnitřní paměti 64 Kb, vybavený velkokapacitními diskovými jednotkami /kapacita jednoho diskového svazku je 29 mil. bytů/, malými disky/ 7,25 mil. bytů/, magneticko-páskovými jednotkami, snímači a děrovačem děrných štítků, snímači a děrovačem děrné pásky a širokořádkovými tiskárnami.

Velmi často se vyskytujícími funkcemi programu bývají zpravidla údržby kmenových souborů, výběry jejich dat podle různých podmínek, jejich seřídění a výpisy. Universální třídící programy jsou většinou součástí základního software dodávaného výrobcem počítače. Výběry, výpisy a údržba souborů jsou převážně předmětem uživatelských programů. Především v těchto oblastech jsme viděli zdroj úspory programátorské práce, strojového času na příklady

a ladění programů a hlavně možnost dosáhnout požadovaných výsledků v pokud možno nejkratším čase. Snaha dát uživateli žádané sestavy v nejkratší době byla hlavním důvodem pro vypracování těchto programů:

Univerzální programy vznikaly v našem výpočtovém středisku /VS/ z potřeb subsystému personálního úseku, a to jmenovitě z potřeb využívání souboru Jednotné evidence pracujících /JEP/. Téměř denně docházejí do VS požadavky na různé výběry, seřídění a výpisy dat. Požadované termíny na dodání sestav bývají 1 až 2 dny, a v mnohých případech kratší než 24 hodin. Při tradičním způsobu programování to představuje jen v oblasti JEP kapacitu nejméně 1 programátora nemluvě o tom, že u počítače plně využívaného ve třech směnách by se termíny splnění takovýchto požadavků prodloužily v průměru na 1 až 2 týdny. Především tyto skutečnosti vedly skupinu pracovníků JEP ke hledání nových forem, jak splnit požadavky na pořizování různých sestav bez vypracování desítek a stovek speciálních výběrových a výpisových programů. Hlavně tato skupina pracovníků definovala požadavky na funkce univerzálních programů i na formy výstupních sestav. Původní požadavky na funkce univerzálních programů byly v průběhu využívání těchto programů dále rozšířeny až do stávající podoby. Několikaletá zkušenost z vývoje a využívání univerzálních programů však říká, že rozvoj těchto programů bude pokračovat.

Při analýze problému, jak řešit vypracování takovýchto programů, jsme vycházeli z těchto kritérií:

- 1) pro každou funkci /údržba, výběr, parametrické zadání údajů pro třídění, výpis/ bude vypracován samostatný program;
- 2) možnost zpracovávat soubory sekvenčního a index-sekvenčního uspořádání s libovolnou délkou fyzické i logické věty /délnka bloku, věty/;
- 3) možnost zpracovávat soubory na různých typech vstupních a výstupních zařízení;

- 4) optimální spotřeba strojového času;
- 5) minimální rozsah údajů uváděných na vstupních parametrických štítcích.

Základním principem řešených programů, psaných v jazyce ASSEMBLER, jsou bloky fiktivních instrukcí, které jsou modifikovány na základě údajů z parametrických štítků a údajů z pomocného souboru, který obsahuje údaje o struktuře zpracovávaného souboru. Uvedený pomocný soubor, dále nazývaný souborem KODY, tvoří základ celého systému vypracovaných univerzálních programů.

I. Soubor KODY

Soubor KODY obsahuje údaje, které jsou nezbytné pro činnost dále popisovaných univerzálních programů. Z tohoto důvodu, a pro lepší orientaci v tomto materiálu, uvádíme strukturu věty a stručný popis jednotlivých údajů:

1. Struktura index-sequenčního souboru KODY:

Symbol: název.	Způsob uložení	počet znaků	Délka v bytech	Význam údaje
KOD	CHAR	4	4	kód údaje
DISP	FIXED	4	3	posice údaje ve větě /KEYLOC/
ROZLIS	FIXED	3	2	kód způsobu uložení údaje, délky a počtu desetinných míst
POCMIST	FIXED	3	2	počet sloupců na sestavě
ZAHL1	CHAR	16	16	1. řádek záhlaví
ZAHL2	CHAR	16	16	2. řádek záhlaví

2. Charakteristika jednotlivých údajů souboru KODY:

KOD - čtyřznakový řetězec, který identifikuje údaj popisované struktury souboru dat a má funkci klíče. Může obsahovat libovolné znaky včetně blanků.

DISP - dekadické číslo v pevné řádové čárce /DEC FIXED/
udává pozici 1. bytu popisovaného údaje /KEYLOC/.

ROZLIS - údaj, který je deklarován v popisovaném souboru jako znakový řetězec /CHAR/, je v údají ROZLIS vyjádřen počtem deklarovaných bytů. Např. pro desetiznakový řetězec /CAR(10)/ má ROZLIS hodnotu 10. Údaj deklarovaný v popisovaném souboru jako číslo ve zhuštěné formě /DEX FIXED/ je v údají ROZLIS vyjádřen dvojmístným sporným číslem. První číslice vyjadřuje délku údaje v bytech, druhá číslice udává počet desetinných míst. Např. pro číslo deklarované jako FIXED(9,2) má ROZLIS hodnotu -52, pro deklaraci FIXED(3) má ROZLIS hodnotu -20.

POCMIST - tento údaj vyjadřuje buď rozsah vlastního údaje zvýšený po počet požadovaných mezer mezi údají nebo rozsah názvu údaje pro tisk v záhlaví. Do POCMIST se uvádí údaj s větším rozsahem.

ZAHL1 - 1. řádek názvu údaje pro záhlaví sestavy.

ZAHL2 - 2. řádek názvu údaje pro záhlaví sestavy.

Takovýmto způsobem lze v souboru KODY popsat celou řadu datových souborů /zpravidla souborů dotýkajících se jedné agendy, skupiny úloh ap./: Pro maximální využití všech funkcí jednotlivých univerzálních programů je vhodné popisovat některé údaje několika větami v souboru KODY. Např. údaj o jméně ze souboru JEP je popsán třemi kódy, které umožňují tisk jména na 40, 30 nebo 20 míst podle toho, kolik máme místa na sestavě. Takto popsaných údajů je celá řada:

Příklad popisu údajů matričního souboru JEP v souboru KODY:

KOD	DISP	ROZLIS	POCMIST	ZAHL1	ZAHL2
104	1	10	11	104-RODNE	CISLO
101	11	30	31	101-	JMENO
101A	11	40	41	101-	JMENO
101B	11	20	21	101-	JMENO

KOD	DISP	ROZLIS	POCMIST	ZAH11	ZAH12
107	86	-50	10	107-	BYDLISTE
107S	86	-52	10	CELKOVY	PLAT
447	248	-42	9	447-PLAT	PUVODNI
447S	248	-42	9	ZAKLADNI	PLAT
998A	314	1	2		
998E	314	1	10		
998H	314	1	16		

Tohoto několikanásobného popisu údajů je využíváno především výběrovým a výpisovým programem:

II. Stručná charakteristika jednotlivých universálních programů

1. Universální údržbový program

1.1. Funkce programu

Program zajišťuje změny libovolných údajů v libovolném index-sekvenčním souboru a vypisuje protokol o provedených změnách.

1.2. Základní princip programu

Program je zpracován v jazyku Assembler IBM 360 a využitím fiktivních instrukcí, do nichž jsou na základě parametrických štítků a pomocného souboru KODY vkládány délky údajů a příslušné displacementy.

Vstupní data změn mohou být uložena na děrných štítcích /80 sloupců/, magnetických páskách, magnetických discích IBM 2311 nebo 2314:

Činnost programu je řízena dvěma parametrickými štítky, z nichž první popisuje index-sekvenční soubor, do kterého budou realizovány změny, druhý pak popisuje soubor změn. Nejběžněji používaným nositelem změn jsou děrné štítky. Tyto datové štítky obsahují v nejjednodušším použití tyto základní údaje:

- a) indentifikace štítku
- b) klíč věty, které má být předmětem údržby
- c) kód údaje, který má být změněn
- d) nový údaj - nahradí původní obsah měněného údaje
- e) kontrolní číslo - pokud je uvedeno, provede se změna ve větě nalezené podle klíče jen tehdy, když toto kontrolní číslo je shodné se zvoleným údajem ve větě.

Při realizaci takto zadaných změn vzniká protokol, který obsahuje: klíč měněné věty, kód změněného údaje, hodnotu původního údaje, hodnotu nového údaje a kontrolní číslo /pokud bylo použito/. Program je postaven tak, aby umožňoval i změny jen části údaje.

Závěrem k tomuto programu lze říci, že byl vypracován na základě požadavků operativního provádění změn /oprav/ v rozsáhlých matričních index-sequenčních souborech. Pravidelná změnová služba je řešena jiným způsobem.

Příklad zadání změn údajů v matričním souboru JEP:

```
// JOB S262UNIV
// DLBL KODY, 'ISQ SOUBOR KODY-XXXX',99/365/ISE
// EXTENT SYS014,DSK502,4,1,2039,1
// EXTENT SYS014,DSK502,1,2,2040,20
// DLBL UNSOUB, 'MATRIKA JEP-XXXX',99/365,ISE
// EXTENT SYS014,DSK502,4,1,418,2
// EXTENT SYS014,DSK502,1,2,420,1560
// EXTENT SYS014,DSK502,2,3,1980,40
// EXEC S26AUNIV
P1 21 270 630 315 10 1 1 1312SH01JEP 51 6
P2 4 260 80 80
UD2556131000 309 STORNO
UD3004190120 422 5 08192
UD4312210990 422 5 A04121
UD4959222900 452B 77 A01287
```

/*
18

Při dalším provádění změn v daném souboru se budou měnit pouze parametrické štítky označené "UD".

2. Universální výběrový program

2.1. Funkce programu

Základní funkcí tohoto programu je výběr vět na základě splnění podmínek uvedených na parametrických štítcích. Program byl postupně rozšiřován o řadu dalších funkcí a ve své dnešní podobě zajišťuje:

- a) výpočet rozdílu mezi systémovým a zvoleným datem
- b) korekci takto získaného výsledku pomocí konstanty z parametrického štítku
- c) výběr vět na základě splnění zadaných podmínek
- d) uložení, přičtení a odečtení konstant do vybraných vět na základě splnění zadaných podmínek
- e) násobení údajů uložených ve zhuštěné formě
- f) slučování libovolných údajů uložených ve zhuštěné formě
- g) seřetězení údajů uložených ve znakové i zhuštěné formě do jediného řetězce
- h) převod volených údajů ze vstupní do výstupní věty
- i) vertikální slučování vybraných vět s kumulací zvolených údajů
- j) vytvoření pracovního souboru KODY pro výstupní soubor zkrácených vět.

2.2. Základní princip programu

Program je zpracován v jazyku Assembler IBM 360 a využívá fiktivních instrukcí, do nichž jsou na základě parametrických štítků a pomocného souboru KODY vkládány příslušné displacementy a délky údajů. Jsou vypracovány dvě verze programu. První neumožňuje krácení vybraných vět. Druhá verze tohoto krácení umožňuje a zároveň zajišťuje vytvoření pracovního souboru KODY, který popisuje všechny údaje zkrácené věty. Pracovní soubor KODY je tedy podmnožinou vstupního souboru KODY, od kterého se liší počtem

vět a rozdílnými displacementy. Při vertikálním slučování jsou rozdíly i v délkách slučovaných údajů. Automatické vytvoření pracovního souboru KODY je nezbytné pro následné využití řídicího a výpisového programu.

Činnost programu je řízena devíti druhy parametrických štítků. Pořadí parametrických štítků je pevně stanoveno. Při zadání úlohy, která bude využívat všech funkcí programu, je pořadí řídicích a parametrických štítků následující:

```
// JOB XXXXXXXX
// DLBL KODYI, 'XXXXX',XX,ISE      /vstupní soubor KODY/
// EXTENT SYSXXX,XXXXXX,4,1,XXX,X
// EXTENT SYSXXX,XXXXXX,1,2,XXX,XX
// DLBL KODYO, 'XXXXX',XX,ISC      - výstupní soubor KODY/
// EXTENT SYSXXX,XXXXXX,4,1,XXX,X
// EXTENT SYSXXX,XXXXXX,1,2,XXX,XX
// DLBL UIN, 'XXXXXX',XX          /zpracováváný vstupní soubor/
// EXTENT SYSXXX,XXXXXX,1,0,XXX,XXX
// TLBL UOUT, 'XXXXXX',XX,XXXXXX /výst.soubor vybraných vět/
// EXEC S28AUNIV
```

Takto jednou vytvořené řídicí štítky lze využít při libovolném zpracování daného vstupního souboru univerzálním výběrovým programem. Za těmito řídicími štítky následují parametrické v následujícím pořadí:

P1 - popis vstupního souboru /viz DLBL UIN.../

P2 - popis výstupního souboru a slučování

H04 - parametrický štítek zadávající násobení údajů

&H00, &H01, &H02 - param. štítky výběru údajů do zkrácené výstupní věty /povinný je pouze &H00/

++ - parametrický štítek pro vytvoření znakového řetězce

&H03 - param. štítek pro vertikální slučování vět

PODMÍNKOVÉ ŠTÍTKY - zadání podmínek pro výběr vět.

Těchto několik druhů parametrických štítků umožňuje zpracovat libovolné požadavky na výběr vět, údajů z nich a provedení požadovaných funkcí. V tomto zřízení rozsahu vět a

a údajů, spolu s jejich předzpracováním před tříděním a výpisem, je hlavní význam tohoto programu.

3. Parametrické zadání údajů pro třídící program SORT

3.1. Funkce programu

Program je vypracován v programovacím jazyku PL/I. Je to nezbytný doplněk do systému universálních programů. Umožňuje jednoduché zadání požadavků na třídící program, který je zpravidla používán mezi universálním výběrem a universálním výpisovým programem. Nejvýhodnějším využitím je v případech, kdy má být tříděn soubor vytvořený universálním výběrem při současném krácení vybraných vět. Program využívá soubor KODY a na základě třídících hledisek, zadaných pomocí kódů, zajišťuje setřídění sekvenčních souborů.

3.2. Základní princip programu

Program vytváří dle parametrických štítků OBSAZ a TRIDENI diskový soubor /cca 1 stopa na 2314/ řídicích i datových štítků pro třídící program SORT. Třídící hlediska pro SORT FIELDS se vytvářejí pomocí kódů uvedených na parametrickém štítku TRIDENI. Na parametrickém štítku OBSAZ se uvádí umístění vstupního, výstupního a pracovních souborů. Program předpokládá u vstupního souboru jmenovku "UNIVERZALNI VYBER" a vytváří soubor se jmenovkou "SETR, UNIV.VYBER".

Program zajišťuje následující možnosti zadání třídění:

- a) disk - stejný disk; vstupní soubor i setříděný soubor jsou umístěny na jednom disku. Jako pracovní je použit druhý disk.
- b) disk - jiný disk; vstupní soubor je umístěn na jednom disku, výstupní na druhém. Část druhého a třetí disk jsou použity jako pracovní oblasti.
- c) disk - páska; soubor z jednoho disku je tříděn přes druhý pracovní disk a výstupní soubor je ukládán na magnetickou pásku.

- d) disk - páska; vstupní soubor z jednoho disku je tříděn přes dva pracovní disky a setříděný výstupní soubor je ukládán na magnetickou pásku.
- e) páska - disk; vstupní soubor z magnetické pásky je tříděn přes dva pracovní disky a výstupní soubor je ukládán na jeden z pracovních disků.
- f) páska - stejné páska; vstupní soubor z magnetické pásky je tříděn přes jeden až čtyři pracovní disky a výstupní soubor se ukládá jako druhý na stejnou magnetickou pásku.
- g) páska - jiná páska; obdoba předchozího případu - setříděný soubor je však ukládán na jinou magnetickou pásku.

Hlavním významem tohoto programu je to, že nám umožňuje spolehlivé třídění souborů, u nichž v okamžiku tohoto zadání ani nevíme, jakou strukturu budou mít. Praktické využití tohoto programu bude uvedeno v příkladech využití univerzálních programů:

4. Univerzální výpisový program

4.1. Funkce programu

Program byl postupně doplňován a nyní zajišťuje tyto základní funkce:

- a) výpis jednotlivých vět v rozsahu těch údajů, které byly specifikovány na parametrickém štítku. Maximální rozsah je 15 údajů, v součtu nejvýše 132 znaků na jedné řádce na tiskárně
- b) součtování volených údajů ve třech stupních součtů /2 stupně po změně volitelně zadaných údajů + celkový součet/, při jednom chodu tohoto programu je možno součtovat 1 - 5 údajů
- c) průměrování součtovaných údajů
- d) výpis počtu zpracovaných vět
- e) stránkování sestavy při změně libovolného údaje nebo jeho části
- f) alternativní výpis - jednotlivých vět /bez součtových po-

ložek/

- součtových vět /bez detailních vět/
- jednotlivých i součtových vět

g) výpis jednotlivých i součtovaných vět s vynecháváním jedné volné řádky.

4.2. Základní princip programu

Program je zpracován v jazyku Assembler a jeho činnost je řízena údaji dodanými pomocí parametrických štítků. Při činnosti programu je opět nezbytný pomocný soubor KODY. Program může zpracovávat libovolné magnetopáskové a diskové sekvenčně organizované soubory a index-sekvenční soubory zpracovávané sekvenčně. Program pracuje s osmi druhy parametrických štítků, z nichž sedm je zastoupeno jen jedním štítkem.

P1 - zadání čísla sestavy a rozsahu nápisu .

P2 - popis vstupního souboru

H1, H2, H3 - zadání nápisu sestavy

VYPIS - výběr údajů pro tisk do sestavy

STRANKOVANI - zadání stránkování, součtování, průměrování

F1 - specifikace záhlaví do předtištěného formuláře

P3 - názvový štítek pro 1. stupeň součtování

P4 - názvový štítek pro 2. stupeň součtování.

Základní význam tohoto programu spočívá v tom, že nahrazuje stereotypní programování výstupních sestav.

III. Příklady využití universálních programů

Výše popsané universální programy je možné využívat zcela samostatně. Nejběžnějším je však využití universálního výběru, třídění a výpisu v jedné sérii. Universální programy lze použít prakticky ve všech oblastech zpracování dat. Jen v oblasti JEP je ročně zpracováváno cca 150 jednorázových požadavků na pořízení nejrůznějších sestav. Rovněž roste počet periodicky zpracovávaných sestav. Některé jsou předmětem dále uváděných příkladů.

A. Oblast zpracování JEP

A.1. Formulace požadavku

Pro rozборы a výkaznictví požaduje osobní oddělení měsíčně zpracovávané přehledy o výstupech zaměstnanců ze závodu. Z archivu jsou vybírány výstupy realizované v požadovaném měsíci. Do tohoto výběru jsou dále zahrnuty výstupy z posledního dne předchozího měsíce a naopak nejsou požadovány výstupy posledního dne sledovaného měsíce. Před tiskem má být vybráný soubor seříděn podle důvodu výstupu, kmenného střediska a osobního čísla. V tomto případě není ještě použito universálního zadávání parametrů pro třídění. Výpis je realizován v podobě zadané parametrickými štítky H1 - VYPIS. V sestavě jsou vypsané detailní položky a jsou součtovány výstupy podle jednotlivých důvodů.

A.2. Zadání řídicích a parametrických děrných štítků

```
// JOB RHAL12 VYSTUPY MESICNI
```

```
ASSGN SYSCLB,UA
```

```
// DLBL KODY, 'ISQ SOUBOR KODY-XXXX',99/365,ISE
```

```
// EXTENT SYSQ14,DSK502,4,1,2039,1
```

```
// EXTENT SYSQ14,DSK502,1,2,2040,20
```

```
// TLBL UIN, 'VYST.ARCH.-KONV.',,000219
```

```
// TLBL UOUT, 'VYSTUPY MESICNI',,MT0000
```

```
// EXEC S19AUNIV
```

```
P1      36    277    630    315
```

```
P2       5    278   1575    315
```

```
309          E    310576      +
```

```
309     2     4     E    0676      +
```

```
309     0     2     NE    30      *
```

```
/*
```

```
// ASSGN SYSQ01,X'282'
```

```
// ASSGN SYSQ02,X'284'
```

```
// ASSGN SYSQ04,X'192'
```

```
// TLBL SORTINI, 'VYSTUPY MESICNI',,MT0000
```

```
// DLBL SORTWK1,,0
```

```
// EXTENT SYS003,111111,1,0,10,500
// DLBL SORTWK2,,0
// EXTENT SYS004,111111,1,0,10,500
// TLBL SORTOUT,'SETR.VYSTUPY-BABL',,MTC000
// IBLTYP TAPE
// EXEC SORT
```

```
SORT FIELDS=(275,2,A,130,5,A,51,6,A),FORMAT=BI,SIZE=500
WORK=2
```

```
RECORD TYPE=F,LENGTH=513
INPFIL BLKSIZE=1575
OUTFIL BLKSIZE=1575
END
```

-*

RESET PROG

```
// DLBL KODY,'ISQ SOUBOR KODY-KYXX',99/365/ISE
// EXTENT SYS014,DSK502,4,1,2039,1
// EXTENT SYS014,DSK502,1,2,2040,20
// TLBL UNSOUB,'SETR.VYSTUPY-BABL',,MTC000
// EXEC SLBAUNIV
```

```
H1          *XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*
H2          * RH112 - STATISTIKA VYSTUPU *
H3          *XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*
F1 SHS18A1/V04 45          8302-OSOBNÍ ODDELENI
P2          4 276 1575 315
STRANKOVANI          VS 310
VYPIS 104 101 401 402 103 301 309 310
```

/*

12

Požadavky na pořizeni dalších sestav z vybraného souboru výstupů jsou realizovány pouhou změnou štítku SORT FIELDS a štítků H1 až VYPIS.

B. Oblast zpracování garančních nákladů

B.1. Formulace požadavku

Při zpracování garančních nákladů je kromě jiných požadována sestava, která bude vyjedřovat finanční podíl jednotlivých závad na garančních nákladech. Celková částka má být členěna na materiálové a mzdivé náklady. Z příslušného souboru budou vybírány všechny věty, ale jen ty údaje, které budou požadovány v dalším zpracování /krácení vět/. Setřídění má být provedeno podle čísla závady - A210 a typu věty - A100 /materiál, mzdy/. Výpisem má být součtovaná a průměrovaná sestava, kde se má 1. součet tisknout po změně typu věty a 2. součet po změně čísla závady. Součtované a průměrované mají být údaje: SMC_KCS - A270 a procentní podíl na celkových garančních nákladech - A330.

B.2. Zadání řídicích a parametrických děrných štítků

```
// DLBL KODY1, 'ISQ-KODY-REKLAMACE', 99/365, ISE
// EXTENT SYS014, DSK507, 4, 1, 2879, 1
// EXTENT SYS014, DSK507, 1, 2, 2880, 20
// DLBL KODY0, 'ISQ-KODY-PRACOVNI-REKLAMACE', 0, ISC
// EXTENT SYS014, DSK507, 4, 1, 2876, 1
// EXTENT SYS014, DSK507, 1, 2, 2840, 20
// TLBL UIN, 'ASOUB REKL-KONV', , MPO000
// EXEC S28AUNIV
```

```
P1      35  278  1106  79
```

```
P2      5  277
```

```
&H00  A300 A100 A270 A210 A330
```

```
A100      NE      *
```

/*

```
ASSGN SYSC1B, X'171'
```

```
// DLBL KODY, 'ISQ-KODY-PRACOVNI-REKLAMACE', 0, ISE
// EXTENT SYS014, DSK507, 4, 1, 2876, 1
// EXTENT SYS014, DSK507, 1, 2, 2840, 20
// DLBL VSTUP, 'UNIV.TRIDENI', 99/365
// EXTENT SYS014, DSK507, 1, 0, 2878, 1
// DLBL TRID, 'UNIVERZALNI TRIDENI', 0
// EXTENT SYS014, DSK507, 1, 0, 2877, 1
```

// EXEC S072TRID

TRIDENI	A210	A100	
OBSAZ	191	283	190

/*

ASSGN SYSCLB,UA

// DLBL IJSYSIN, 'UNIVERZALNI TRIDENI',0

// EXTENT SYSOL4,DSK507,1,0,2877,1

ASSGN SYSIN,X'171'

// DLBL KODY, 'ISQ-KODY-PRACOVNI-REKLAMACE',0,ISE

// EXTENT SYSOL4,DSK507,4,1,2876,1

// EXTENT SYSOL4,DSK507,1,2,2840,20

// DLBL UNSOUB, 'SETR.UNIV.VYBER',0

// EXTENT SYSOOL,111111,1,0,10,1990

// EXEC S18AUNIV

H1 P O D I L Z A V A D

H2 NA GARANČNICH NAKLADECH ZE SLEDOVANÝCH OPRAVEN

F1 SHS18A1/UN3 46 C 8510 - OES

P2 4 275

STRANKOVANI S A100 A270 A330

/*

/R

Celá úloha byla prakticky vyřešena jednoduchým zadáním devíti parametrických štítků:

C. Vyhodnocení automobilové soutěže ŠKODA RALLYE

C.1. Formulace požadavku

Soubor obsahuje 145 identifikačních a hodnotových údajů. Z nich mají být vypsány časy prvních 10 rychlostních zkoušek a údaje identifikující posádku a typ vozu. Požadovaný výpis má být seříděn podle startovních čísel a v plném rozsahu, t.j. i včetně odpadlých posádek. Protože soubor je organizován index-sekvenčně podle startovních čísel, není v tomto případě nutné použít výběrový a třídící program.

C.2: Zadání řídicích a parametrických děrných štítků

```
// JOB RO4ARALI  STARTOVNI LISTINA JEDN.  
// OPTION NOLOG  
// DLBL KODY, 'SOUB.KODY PRO RALLYE SKODA-XXXX',99/365,ISE  
// EXTENT SYS014,DSK521,4,1,700,2  
// EXTENT SYS014,DSK521,1,2,680,20  
// DLBL UNSOUB, 'RALLYE-JEDNOTLIVCI',0,ISE  
// EXTENT SYS014,DSK521,4,1,600,1  
// EXTENT SYS014,DSK521,1,2,620,40  
// EXEC S18AUNIV
```

```
F1 SY518A1      33  
P2      6  270  760  760    3    1  
H1      S K O D A    R A L L Y E    76  
H2      CASY RYCHLOSTNICH ZKOUSEK /V SEC./  
H3      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
VYPIS CISP JKEN ZEME ZNAC RZ01 RZ02 RZ03 RZ04 RZ05 RZ06  
                                RZ07 RZ08 RZ09 RZ10
```

STRANKOVANI VS

/*

/8

Universální programy se používají i v dalších oblastech zpracování dat. Jsou řešeny úlohy s daleko složitějším algoritmem zpracování.

IV. Závěr

Pro nedostatek místa musel být popis universálních programů veden prakticky v obecné rovině. Nebyly popsány moduly, které umožňují zpracovávat soubory různého uspořádání /sekvenční a index-sekvenční/, různé délky fyzické a logické věty, různého blokování i různých medií: Není popsán konkrétní obsah jednotlivých parametrických štítků zadávaných do jednotlivých programů. Nejsou popsány chybové zprávy, které vypisují jednotlivé programy při chyb-

ném zadání údajů na parametrických štítcích. Nejsou přiloženy ani vzory sestav, které se esatně od běžných nijak nelíší.

Naším příspěvkem jsme chtěli upozornit na existenci systémů univerzálních programů, jednoduché zadávání parametrů s jejich snadnou úpravou při změně zadání úlohy. Tím vyzvednout úsporu programátorské kapacity a strojového času na překlady a ladění programů, možnost okamžitého vyhovění uživatelům /zpracování většiny požadavků do příslušných parametrů trvá jen 10 - 20 minut/. Systém univerzálních programů prakticky umožňuje promítat logiku běžných speciálních programů do parametrického vyjádření se všemi výhodami, které z toho plynou. Díky těmto programům je jen v oblasti zpracování personální agendy trvale spotřena kapacita nejméně jednoho programátora. Ročně je možno vykázat více jak 60 hodin uspořádaného strojového času. Zájem o využívání univerzálních programů vedl k jejich úspěšnému nasazení i ve výpočtových střediscích vybavených počítači EC 1030 a EC 1040.

V současné době je v našem výpočtovém středisku připravován program, který bude poskytovat uživatelům univerzálních programů další možnosti. Takovýmto způsobem se snažíme přispívat k efektivnějšímu využití lidských a strojových kapacit v našem výpočtovém středisku.