

GENEROVÁNÍ PROGRAMŮ COBOL A PROCEDUR DOS 3/4

Ing. Vlastimil Čevela, VS Ingstav n.p. Brno

Příspěvek vychází ze zkušeností s konkrétním generátorem programů Cobol a procedur Dos 3/4 na počítači EC 1026 a ukazuje v širší poloze oblast využití takového prostředku pro částečnou automatizaci programátorských prací. Současně je též vysvětlena a příklady doložena filozofie tvorby generátoru na bázi aplikačních a typových textů, automatická vazba na slovník dat přes jméno údaje dle konvencí JÚZO a možnost samodokumentujících funkčních popisů.

0. Úvod

Tvorba programů ve zdrojovém programovacím jazyku a návrhů příkazových štítků se obecně vyznačuje velikou náročností na pracnost a především na přesnost. Běžně se jedná o desítky a stovky řádků, které musí být zcela jednoznačné jak pořadím, tak i obsahově. Při analýze problematiky, která tvoří náplň této práce, je možné vytvořit jistá typová řešení, která jsou použitelná pro určitou třídu řešených problémů a s jejich pomocí pak tvorbu programového vybavení těchto úloh částečně automatizovat.

V oblasti zpracování hromadných dat můžeme v tomto snažení sledovat zhruba tři hlavní směry - podprogramy, parametrické programy a generátory programů, jejichž stručné charakteristiky uvedeme dále.

podprogram

- pracuje jako cílový modul
- velice vhodný pro řešení dílčích funkcí
- umožňuje zpracovat libovolné dílčí algoritmy, které jsou jednoznačně dány předem
- zpravidla vysoce ekonomický provoz
- nedovoluje řízení programu na vyšší úrovni

parametrický program

pracuje jako cílový program
vhodný pro řešení ucelených komplexnějších funkcí
vyžaduje vytvoření složitého univerzálního algoritmu,
který je pomocí parametrů modifikován nebo doplňován
zpravidla méně ekonomický provoz
řízení programu na vyšší úrovni je předem dáno

generátor programů

vytváří zdrojové texty
umožňuje řešení dílčích i komplexních funkcí
dovoluje stavebnicové vytvoření různých algoritmů,
sestavených z aplikačních a typových textů
běžně ekonomický provoz
zvláště vhodný pro řízení programu na vyšší úrovni

1. Filozofie tvorby generátoru programů a procedur

Počítačové prostředí, které máme k dispozici u současných počítačů a jejich operačních systémů nabízí velice zajímavé náměty v oblasti částečné automatizace programátorských prací. Umožňuje totiž pracovat se zdrojovými texty programů i s procedurami příkazových štítků jako s daty, tj. realizovat pomocí běžně zvládnutých programátorských technik jejich prakticky libovolné zoracování. V dalších odstavcích bude tato představa rozvedena na příkladě generátoru programů Cobol s procedur příkazových štítků Dos 3/4 dle /2/, konkrétně realizovaného ve VS Ingstav Brno na počítači EC 1026.

Podstatou práce generátoru je, že na základě zjednodušeného tzv. aplikačního textu vytvoří kompletní výsledný zdrojový i cílový program Cobol, včetně procedury příkazových štítků, potřebné pro jeho spuštění. Činnost generátoru probíhá ve 3 úrovních:

- 1/ stavebnicové využívání typového textu, předem připraveného v knihovně
- 2/ tvorba určitých deklarácí a příkazů přímo generátorem
- 3/ doplňování deklarácí údajů dle slovníku dat.

2. Spojování aplikačních a typových textů

Základním principem je jednoduché slučování řádků obecného typového textu s řádky konkrétního aplikačního textu. Typový text je přitom složen z řádků základního řídicího algoritmu, který je generován vždy a z řádků zdrojového textu a procedury příkazů, specifických pro jednotlivé druhy vstupů a výstupů, jako např. různý diskový či páskový soubor, fronta, knihovna a pod., které jsou jednou nebo vícekrát vybrány dle právě požadovaného zadání.

Slučování je řízeno šestimístným složeným klíčovým polem, obsahujícím v typovém textu následující údaje /viz příloha 3./:

- rozlišení zdrojového textu /S/ nebo procedury /P/
- označení bloku výsledného programu /písmeno/
- rozlišení základního algoritmu a souborů /2 znaky/
- pořadové označení řádku v rámci výsledného bloku /znak/
- příkaz k modifikaci typového textu dle zadání /znak/

Dle výše uvedených zásad je pak možno pomocí různé skladby jednoho řádku v bloku P2 aplikačního textu /viz příloha 1./ ošetřit práci se širokou paletou vstupů a výstupů, jako je klávesnice terminálu, disketa, páskový soubor dle normy Tesla i EC, pracovní třídící soubor, indexový i sekvenční diskový soubor, člen knihovny nebo fronty, terminálová tiskárna a pod.

Stejným způsobem však je možno typově naprogramovat i složitější úlohy, jako je např. čtení z jednotně organizované základny vstupních dat, změnové řízení kartoték či jiné, včetně třeba i algoritmu normovaného programu. Tak lze odštípnout práci aplikačního programátora od navrhování i psaní všech algoritmů, které je možno typizovat ať již jako řídicí, nebo ve formě podprogramů, volaných perform nebo call s parametry.

Nikoliv zanedbatelnou skutečností přitom zůstává fakt, že typové texty jsou psány běžným zdrojovým jazykem Cobol a formou normálních příkazových štítků. Při jejich psaní je pouze nutno správně vytvořit výše zmíněné složené klíčové pole a vhodně využívat proměnné, které jsou obsežovány až při generování dle konkrétního aplikačního textu.

3. Možnosti doplňovacích funkcí a slovník dat

Programové zpracování aplikačního textu generátorem dovolu-
je automatizovat psaní některých slov a symbolů jazyka Cobol, a tak
usnadnit a zkvalitnit programátorskou práci. /viz přílohy 1. a 2./

- řetězce deklarací libovolných konstant, textů, parametrů ap.
lze psát volným přehledným způsobem do více řádků či sloupců
- v rámci Cobol-Report lze generovat jednotné záhlaví sestavy
a většinu rezervovaných slov deklarací
- v datových sekcích je možno přes jméno údaje, sestavené
dle pravidel /3/ pro kód JÚZO, tj. písmeno + 3 číslice,
automaticky přiřazovat základní nebo editované deklarace
ze slovníku dat, definovaného ve formě nezávislého číselní-
kového modulu dle systému DKR, příp. indexového souboru.

4. Zkušenosti z využívání generátoru

2a hlavní výhodou lze považovat podstatné zjednodušení a zrych-
lení programátorské práce se všemi významnými druhy I/O dat a to
nejen při prvotní tvorbě programu, ale především při změnách, ově-
řováních, pomocných výběrech, výpisech a pod. Dalším přínosem pak
je zvýšení přesnosti a spolehlivosti. Tím, že je část řádků prog-
ramu a celá procedura tvořena automaticky, nejsou v těchto řádcích
nepřesnosti a tudíž se dále zvyšuje rychlost ladění a ověřování.

5. Podklady a literatura

- /1/ Dokumentace k operačním systémům Dos 3/4
- /2/ Generátor programů a procedur Dos 3/4,
2N-321/84, Ingstav n.p. Brno
- /3/ Jednotná údajová základna organizací,
Vyhl. č. 173/1980 Sb. a příslušné výnosy FSÚ

Příloha 1. - aplikační text programu K05045

Příloha 2. - výsledná procedura a program K05045

Příloha 3. - příklad typového textu

```

P1 Aplikační text
P2 (SOUČTOVANÍ)
P3
P4
P5
P6
P7
P8
P9
P10
P11
P12
P13
P14
P15
P16
P17
P18
P19
P20
P21
P22
P23
P24
P25
P26
P27
P28
P29
P30
P31
P32
P33
P34
P35
P36
P37
P38
P39
P40
P41
P42
P43
P44
P45
P46
P47
P48
P49
P50
P51
P52
P53
P54
P55
P56
P57
P58
P59
P60
P61
P62
P63
P64
P65
P66
P67
P68
P69
P70
P71
P72
P73
P74
P75
P76
P77
P78
P79
P80
P81
P82
P83
P84
P85
P86
P87
P88
P89
P90
P91
P92
P93
P94
P95
P96
P97
P98
P99
P100

```

Aplikační text

(SOUČTOVANÍ)
 1 311, 2 - 2/85 EK
 I 'UČTO TESLA'
 O 'PŘÍKLAD REKAPITULACE'
 O 'PŘÍKLAD POL. ROZPIŠU'

M05045 'ING KOLACEK & TNOJ
 KOCHOVÁ
 M00619-C T*VC 75 4 11 7
 S01741 L2F 12 1
 S01742 L3F 12 1

'TYPOVA KONVERZE'
 'PŘEVOD ZNAKOVY'
 'PŘEVOD ZHUSTENY'
 E00125
 I00001
 I00003

SEKVENČNÍ ČTENÍ MAGNETICKE PASKY SOUBORU Z TESLY
 A TVORBA VYSTUPNICH SESTAV VE SHODNEM TRIDEMI

VZROVY PŘÍKLAD NA PRACI S REPORT SECTION COBOL
 V RANCI TYPOVE KONVERZE

PŘÍKLAD BLOKU PARAMETRU

P1- 3 X 09 0000
 9 03 0000
 Y 30 0012
 Y1- 'CHYBNA DATA'
 Y2- 'NESPRAVNY KOD'

MOZNOSTI DOPLŇOVANI ZE SLOVY

* 1 P.
 2 E081
 3 E081
 4 E081

RD S 1742 CONTROL Y9.3
 PAGE 86 FIRST DETAIL 7.

* POL. SESTAVA
 * ZÁHLAVI SEST.

```

LINE S. CI TP MM.  

330 .PM.  

45 .RM.  

61 .RM.  

77 .RD.  

93 .RD.  

100 .RD.  

113 .ZA.  

121 TYPE DE.  

1 LINE PLUS 1.  

3 E0774.  

4 E0775.  

7 Y0336.  

13 Y0337.  

16 Y0660.  

19 Y0991.  

33 E0666.  

51 E0882.  

67 E0995.  

83 X0333.  

99 E0886.  

118 Y0003 GROUP.  


```

1 S01 TYPE OF Y903 NEXT GROUP NEXT PAGE.

* BEŽNY RÁDEK
 * SKOK NA NOV.
 * REKAP. SEST.
 * ZÁHLAVI SEST.

RD S 1741 CONTROL Y9.3 E976
 PAGE 86 FIRST DETAIL 7.

```

LINE S. CI TP MM.  

330 .PM.  

45 .RM.  

61 .RM.  

77 .RD.  

93 .RD.  

102 .RD.  

114 .SC.  

144 .ZA.  

151 TYPE OF Y93.  

1 LINE PLUS 3.  

3 Y0336.  

4 Y0337.  

7 Y0660.  

13 Y0993.  

19 E0991.  

33 Y(18)9,00-1.  

49 E0991.  


```

* SOUCTY ZA Y

1D APLIKACNI BLOK (DEKLARACIJA)
PRIVLEK BLOK (PARAMETRI)

P1
PIC X(0) VALUE '3X5000000'
PIC X(7) VALUE '00000000'
PIC X(7) VALUE '1370012'
Y1
PIC X(15) VALUE 'CHYRNA DATA'
Y2
PIC X(15) VALUE 'NESPREAVNY KOD'

1D APLIKACNI BLOK (DEKLARACIJA)
POZNAVATI DOPIPOVANI ZE SLO

P
E981 PIC S(11) V90 COMP-3
E981 PIC Z(10) 9,99-
PIC X
E981 PIC Z(10) 9,99-

SLOPORY (VETV, KLICE)

A-2 PIC X(6) VALUE 'S01741'
B-2 PIC X(30) VALUE 'PRIKLAD REKAPITULACE'
C-2 PIC X(9) VALUE 'VIZ DATE'
A-3 PIC X(6) VALUE 'S01742'
B-3 PIC X(7) VALUE 'PRIKLAD POL. ROZPISU'
C-3 PIC X(9) VALUE 'VIZ DATE'

LINKAGE.

1D APLIKACNI BLOK (DEKLARACIJA)

COMMUNICATION.

1D APLIKACNI BLOK (DEKLARACIJA)

REPORT.

1D APLIKACNI BLOK (DEKLARACIJA)

RD S01742 CONTROL Y903
PAGE 56 FIRST DETAIL 7.

1 L-3 TYPE PH LINE 2
COL 1 PIC X(15) VALUE 'INGS TAV BRNO -'
COL 14 PIC X(6) SOURCE A-3
COL 23 PIC X(3) SOURCE B-3
COL 54 PIC X(7) VALUE 'STRANKA :'
COL 64 PIC Z(9) SOURCE PAGE-COUNTER.
2 LINE 3
COL 1 PIC X(11) VALUE 'PLATNOST :'
COL 12 PIC X(8) SOURCE C-3
COL 57 PIC X(7) VALUE 'DATE :'
COL 64 PIC X(8) SOURCE CURRENT-DATE.
LINE 5
COL 003 PIC X(15) VALUE 'SU AU CI TP MM'.
COL 009 PIC X(02) VALUE 'PK'.
COL 045 PIC X(02) VALUE 'RM'.
COL 061 PIC X(02) VALUE 'MM'.
COL 077 PIC X(02) VALUE 'PD'.
COL 093 PIC X(02) VALUE 'RD'.
COL 109 PIC X(02) VALUE 'ND'.
COL 118 PIC X(02) VALUE 'ZA'.

1 R 31 TYPE PH
LINE PLUS 1
COL 002 PIC 999
COL 006 PIC 999
COL 010 PIC 999
COL 013 PIC 999
COL 014 PIC 999
COL 015 PIC Z(1) 9,99-
COL 035 PIC Z(10) 9,99-
COL 051 PIC Z(10) 9,99-
COL 067 PIC Z(1) 9,99-
COL 083 PIC Z(1) 9,99-
COL 099 PIC Z(1) 9,99-
COL 118 PIC 999

SOURCE E974
SOURCE E975
SOURCE Y936
SOURCE Y937
SOURCE Y960
SOURCE E981
SOURCE E984
SOURCE E983
SOURCE E985
SOURCE E983
SOURCE E986
SOURCE Y903

8. BEZNY RAD

S 31 TYPE PH Y903 NEXT GROUP
RD S01741 CONTROL Y903 E974
PAGE 56 FIRST DETAIL 7.


```

1 L-2 TYPE PH LINE 2
COL 16 PTC X(15) VALUE INGSTAY PRMO -
COL 27 PTC X(8) SOURCE A-2
COL 54 PTC X(30) SOURCE R-2
COL 64 PTC X(10) VALUE *STRANKA :
COL 74 PTC ZZ99D SOURCE PAGE-COUNTER.

2 LINE
COL 11 PTC X(17) VALUE *PLATNOST :
COL 27 PTC X(8) SOURCE C-2
COL 54 PTC X(7) VALUE *DATE :
COL 74 PTC X(8) SOURCE CURRENT-DATE.

LINE 3
COL 11 PTC X(09) VALUE *CI TP *
COL 27 PTC X(02) VALUE *P*
COL 54 PTC X(02) VALUE *SM*
COL 74 PTC X(02) VALUE *MM*
COL 11 PTC X(02) VALUE *PD*
COL 27 PTC X(02) VALUE *PD*
COL 54 PTC X(02) VALUE *MB*
COL 74 PTC X(02) VALUE *SC*
COL 11 PTC X(02) VALUE *ZA*

```

05045

```

1 S21 TYPE PLU
LINE 000000
COL 11 PTC 09 SOURCE Y976.
COL 27 PTC 99 SOURCE Y977.
COL 54 PTC 99 SOURCE Y978.
COL 74 PTC 99 SOURCE Y979.
COL 11 PTC Z(10) 9,99- SUM E081.
COL 27 PTC Z(10) 9,99- SUM E082.
COL 54 PTC Z(10) 9,99- SUM E083.
COL 74 PTC Z(10) 9,99- SUM E084.
COL 11 PTC Z(10) 9,99- SUM E085.
COL 27 PTC Z(10) 9,99- SUM E086.

```

* SOUCTY 24

05045
05045

```

1 S22 TYPE PLU
LINE 000000
COL 11 PTC 09 SOURCE Y936.
COL 27 PTC 99 SOURCE Y937.
COL 54 PTC 99 SOURCE Y940.
COL 74 PTC 99 SOURCE Y943.
COL 11 PTC Z(10) 9,99- SUM E081.
COL 27 PTC Z(10) 9,99- SUM E082.
COL 54 PTC Z(10) 9,99- SUM E083.
COL 74 PTC Z(10) 9,99- SUM E084.
COL 11 PTC Z(10) 9,99- SUM E085.
COL 27 PTC Z(10) 9,99- SUM E086.
COL 11 PTC 0(5) SOURCE F976.
COL 27 PTC 99 SOURCE Y903.

```

& SOUCTY 24

05045
05045

```

1 P23 TYPE FE.
SOURCE F981.
SOURCE F982.
SOURCE F983.
SOURCE F984.
SOURCE F985.
SOURCE F986.

```

& PDMOCNY 8

05045

```

1 P29 TYPE FE.
COL 001 FIC 09 SOURCE Y935 (5) OF *0075.
COL 012 FIC 9(8) SOURCE KLICE.

```

DALSÍ MOZNOSTI V RAMCI 98

05045

PROCEDURE.

```

* MOVE 0 TO P-I.

```

D. APLIKACNI BLOK (DECLAR

```

* ACCEPT K-I FROM SYS IPT.
ACCEPT E-I FROM SYS IPT.
MOVE TC C-3.
OPEN INPUT S-1 NO BEWING.
OPEN OUTPUT S-2.
OPEN OUTPUT S-3.

```

SOURCNY (OPEN)

```

PERFORM F-P1.
PERFORM F-PY.
IF H-1 = 1
  PERFORM H-Z.

```

```

H-X PERFORM X-DIVERZE
  UNTIL E-I = 1.

```

A
A
A
P
P
R
R
A
A
A
A
A
A

H-W.	PERFORM F-W.		A
H-Z.	PERFORM F-PZ.		A
*		<u>SOUBORY (CLOSE)</u>	A
	CLOSE S-1.		R
	CLOSE S-2.		B
	CLOSE S-3.		E
	STOP PUN.		A
K-ONVERZE.			A
*		<u>SOUBORY (READ, END MOVE 1 T</u>	A
	READ S-1.		P
	END MOVE 1 TO E-F.		B
	IF E-I NOT = 1		A
	ADD 1 TO P-J		A
	PERFORM P-X.		A
P-V.		<u>V APLIKACNI BLOK (ZAHAJENZE)</u>	A
*			A
P-W.		<u>W APLIYACNI BLOK (UKONCENZE)</u>	A
*			A
P-X.		<u>X APLIKACNI BLOK (ZPRAC. TY)</u>	A
*		<u>TRANSFORMACE KODU TESLA -</u>	A
	CALL 'IC0001' USING		H05045
	Y961 E974 E975 Y936 Y937 Y903 E976 E977 Y960 E978.		H05045
	CALL 'IC0003' USING		H05045
	E931 E982 E983 E984 E985 E986.		H05045
*		<u>TVORBA SESTAV S OMEZENIM V</u>	H05045
	IF Y954 = '1' AND Y903 < 3		H05045
	GENERATE R31. ELSE		H05045
	IF Y961 = '3' AND Y903 < 3		H05045
	GENERATE S01741.		H05045
P-P1.		<u>SOUBORY (INITIATE)</u>	A
*			A
	INITIATE S01741.		B
	INITIATE S01742.		B
P-P2.		<u>SOUBORY (TERMINATE)</u>	A
*			A
	TERMINATE S01741.		B
	TERMINATE S01742.		B

Příklad typového textu

```

SB13C *          O INDEXOVY DISKOVY SOUBOR
SB13AN          SELECT S#X ASSIGN SYS1#Y-1 ACCESS SEQUENTIAL
SB13BN          RECORD KEY R#X.
SB13AF FD       S#X.
SB13BY FD       S#X RECORDING V.
SB13CN 1        O#X.
SB13DA 2        PIC X(X012).
SB13EM 2        R#X PIC X(X8).
SB13FB 2        PIC X(X015).
SB13GM          COPY X00010.
SB13AN 1        ENX PIC 9.
SP13AN          OPEN OUTPUT S#X.
SR13AN          CLOSE S#X.
SY13AN W#X.
SY13BN          MOVE 0 TO ENX.
SY13CN          WRITE O#X
SY13DN          INVALID MOVE 1 TO ENX.
PA13AN //       RUN E00150,P=X00001
PA13BN //       RUN E00075,NAME='X00001',PL=0,SYS021=SYS022
PB13AN //       ASSIGN SYS1#Y,SYS022
PB13BN //       DLBL SYS1#Y,'X00001',01,1SC,CYLOFL=4,AUT=3,MDS,CTLG
PB13CN //       EXTENT SYS1#Y,,4,,X14
PB13DN //       EXTENT SYS1#Y,,1,,X004

```