

Ing. Stanislav B A L Á Z

Ústav systémového inžinýrství Bratislava

Generátor normového programovania NPG

1. Úvod

Projektovanie ASR v súčasnosti nie je už tak závislé na technickom a programovom vybavení počítačov ako na nedostatku kvalifikovaných programátorov. Negatívnu úlohu tu vytvára i tá skutočnosť, že na školeniach programátorov sa kladie dôraz na výuku programovacích jazykov, počítačových systémov, avšak menej na logiku a metodiku programovania, ktoré umožňujú racionalizáciu programovania.

Programovanie je príťažlivá práca, pre vykonávanie ktorej sú nutné viaceré predpoklady. Považuje sa za veľmi individuálnu, tvorivú činnosť. Treba si uvedomiť, že tvorivou prácou je len čiastočne. Keď analyzujeme jednotlivé programy z oblasti hromadného spracovania dát, vidíme, že sa v nich opakujú niektoré funkcie /napr. definovanie bodnôt, otváranie súborov, prenosy viet z vonkajších pamätových médií do vnútornej pamäti a naopak, riadi sa príchod spracovania viet, vykonávajú sa výpočty na vetách, súbory sa aktualizujú a pod./. Tieto funkcie sa opakujú vo väčšine programov a niektoré z nich znamenajú pre tvorivého programátora zbytočné zaťaženie rutinnou prácou. Každý takýto program možno rozdeliť na časť riadenia postupu programu a časť vlastného spracovania. Tento poznatok spolu

spolu so systematickou pre stavbu a logiku programu tvorí podstatu normovaného programovania.

Metóda normovaného programovania stanovuje pravidlá a predpisy pre stavbu a logiku priebehu programov, v ktorých sa spracovávajú vety a skupiny viet aspoň jedného vetupného sekvenčného súboru. Je nezávislá vzhľadom na typ programovacieho jazyka i počítača. Medzi základné prínosy aplikácie metódy normovaného programovania sa zaraďuje zjednotenie programovej dokumentácie a zvýšenie prehľadnosti, čím sa uľahčuje údržba programov a ich vymeniteľnosť medzi užívateľmi. Ďalej prínosom je rast produktivity programátorskej práce, urýchlenie testovania programov, úspora strojového času a i.

Na metódu normovaného programovania logicky nadväzuje automatizácia tvorby riadiacej časti programov pomocou podporného programového nástroja - Generátora normovaného programovania NPG. Použitím generátora NPG sa programátor odbremenjuje od rutinnej práce a ostáva mu riešiť iba ťažisko vlastnej problematiky - zostavenie problémovo-orientovaných častí programu. Takto sa dosiahne plný efekt prínosu metódy normovaného programovania.

V nasledovnej tabuľke sú zhrnuté požiadavky v praxi najrozšírenejšieho generátora NPG na počítačovú techniku, pre ktorú ho je možné využiť.

Výrobca/počítač		JSEP	IBM 360 a IBM 370	Siemens 4004	
Model počítača		od EC 1030	od 360/25 všetky	od 35	
Operačný systém		DOS OS	DOS OS	PBS, BSV	
Obsadenie vnútornej pamäti počas generovania	Verzia NPG	ASS	Žiadna potreba - makrotechnika		
		COB	28 KB	28 KB	32 KB
		PL/1	42 KB	42 KB	-
Potreba periférnych jednotiek		1 disková jednotka, 1 tlačiareň a 1 snímač diernych štítkov			

Verziu DOS/COBOL tohoto generátora spolu s metódou normovaného programovania ÚSIP zakúpil od ADV/ORGA v licencií pre ČSSR. Ďalšie časti článku sú orientované na výklad slovenskej verzie generátora NPG, DOS/COBOL.

2. Funkcia generátora NPG

Generátor normovaného programovania /NPG/ zabezpečuje automatizáciu programátorskej práce pri stavbe riadiacich častí priebehu programu. Je to v podstate súbor makropříkazov, ktorý vychádza z logiky normovaného programovania. Generátor umožňuje generovanie rozhodujúcej riadiacej zložky programu, pričom vytvára tie časti programu, pri tvorbe ktorých je zabezpečený výhodný pomer medzi zadáním a výstupom generátora. Generátor umožňuje generovať vo veľkom počte prípadov potrebné riadiace časti programov pomocou troch základných makropříkazov. Parametrami týchto makropříkazov sa popisuje počet vstupných sekvenčne čítaných súborov a pole skupinového kľúča, ktoré je potrebné pre dané spracovanie. Takto vygenerovaný program obsahuje všetky príkazy nevyhnutné pre riadenie programu, pričom logika vychádza zo schémy normovaného programovania. Vygenerované príkazy /skupiny príkazov/ si nevyžadujú ani vo výnimočných prípadoch dodatočnú modifikáciu, prípadne zmenu alebo úpravy.

V generovanom programe prebehnú tieto hlavné bloky:

- úvodný
- vstupný
- riadenie výberu vety
- spracovanie zmeny skupín
- spracovanie jednotlivých viet
- podprogramy.

Časti programu orientované na riešenie vlastnej úlohy sa organizujú ako výstupy, resp. moduly alebo podprogramy generovaného programu. Pre individuálne programovanie sú dané tieto možnosti:

- úvodný blok
- vytvorenie poľa skupinového kľúča
- špecifické ošetrenie chybného zoradenia súboru
- špecifické ošetrenie prvého a posledného priebehu programu
- spracovanie znaky skupinového kľúča
- spracovanie jednotlivých viet
- rôzne ďalšie podprogramy /kontrolné, výpočtové, tlačové, atď./.

Tieto moduly a podprogramy vyhotovuje programátor samostatne a nezávisle na riadení priebehu programu. Program vytvorený pomocou generátora prenecháva programátorovi iba bezpečnostne nutné naprogramovanie jednotlivých častí nezaberaajúcich sa riadením priebehu programu. Spolupráca generátora NPC a programátora pri tvorbe programu je znázornená na obrázku 8. 1.

3. Tvorba zdrojových programov sa použitia generátora NPG

Generátor normovaného programovania vytvára riadiace časti programov pomocou makropriказov modifikovaných príslušnými parametrami konkrétnej úlohy. Základom štruktúry generátora je diagram riadenia priebehu programu podľa schémy normovaného programovania. Tento diagram vychádza z členenia programu na bloky, ktoré predstavujú logicky a funkčne súvisiace časti programu.

Úvodný blok programu obsahuje všetky činnosti nevyhnutné pre prípravu vlastného programu. Je to napr. otváranie súborov, čítanie predpriebehových DŠ, spracovanie riadiacich informácií, tvorba tabuliek, atď. Keďže tieto činnosti sú úzko späté so zadanou konkrétnou úlohou, nemá význam ich automatizovať. Zároveň a účelom generátora NPG nie je tvorba častí programov v pomere 1:1 ako by to bolo napr. v prípade príkazu OPEN. Z týchto príčin sa generátor NPG v tejto oblasti nepoužíva.

Vo vstupnom bloku preberá generátor NPG celkové riadenie. Pre každý sekvenčný vstupný súbor /max. 99/ zabezpečuje:

- riadenie prvého priebehu
- výber modulu pre vstup nasledujúcej vety
- kontrolu vzostupného zoriedenia súboru
- napočítavanie počtu viet súborov
- priebeh pre podmienku EOF.

Programátor kóduje iba príkazy čítania /príkaz READ podobne ako OPEN/ a zostavuje pole skupinového kľúča prenesením potrebných kľúčov pre daný program do neutrálneho riadiaceho pola /NCRU/.

V bloku riadenia výberu vety, kde sa rieši problém výberu viet pre nasledujúce spracovanie a modifikácia vetvenia priebehu programu na moduly spracovania jednotlivých viet, sa NPG plne využíva pre vygenerovanie všetkých potrebných príkazov.

Blok spracovania zmeny skupín vyžaduje sled špecifických

činností, ktoré sú úzko späté s danou úlohou. Blok možno rozdeliť na riadenie priebehu pri výskyte zmien jednotlivých kľúčov a na vlastné riešenie úlohy v jednotlivých moduloch spracovania zmien kľúčov. Je zrejmé, že programátor musí sám zostaviť jednotlivé moduly zabezpečujúce spracovanie zmeny príslušného kľúča. Ovšem problém riadenia priebehu zmien, t.j. kam sa má program vetviť v prípade zmeny toho-ktorého kľúča alebo pri prvom a poslednom priebehu, sa plne automatizuje použitím generátora NPG.

Automatizovať blok spracovania jednotlivých viet i blok podprogramov je s podobných príčin ako u bloku úvodného neúčelné.

Generátor normovaného programovania je naprogramovaný v jazyku assembler. Nevyžaduje si žiaden zvláštny riadiaci programovací jazyk, takže jeho volanie formou makropříkazov nie je oddelené od častí vstahujúcich sa na riešenie daného problému, ale sa vykonáva podľa potreby priebežne a kódovaním na normálnom programovacom formulári. Vyvolanie generátora NPG sa vykonáva príslušnými makropříkazmi vo WORKING STORAGE SECTION, v PROCEDURE DIVISION a vo zvláštnych prípadoch v LINKAGE SECTION. Generátor sa aktivizuje, ak sa v siedmom stĺpci príslušného riadku programovacieho formulára nachádza abecedný znak "N", pokračovacie riadky musia obsahovať znak "P". Vo fáze generovania, ktorá predchádza fáze komplikácie, sa zdrojový program v jazyku COBOL rozšíri o príkazy vytvorené generátorom na základe parametrov zadávaných v jednotlivých makropříkazoch. Definície riadiacich tabuliek sa vytvorí vo WORKING STORAGE SECTION alebo v LINKAGE SECTION. Riadiaca časť programu je časťou PROCEDURE DIVISION, pričom sa vygeneruje hneď na tým istom mieste programu, na ktorom bol uvedený makropříkaz generátora. Konečným výstupom generátora NPG je úplný zdrojový program v jazyku COBOL AFS alebo D COBOL a protokol z generovania.

4. Makropriказы generátora NPG

Ako bolo už uvedené, generátor NPG pracuje na základe parametrov zadaných v jednotlivých makropriказoch. V normálnych prípadoch dostačujú pre riešenie daných úloh tri základné parametre:

1. NPGANZ = d

kde d značí počet sekvenčne čítaných vstupných súborov. Generátor umožňuje čítať maximálne 99 súborov. Zadaný počet súborov slúži generátoru i pre definovanie riadiacej tabuľky, v ktorej každý súbor zaberá jednu časť. Generátor predpokladá v programe existenciu modulov B21 až B2d a taktiež modulov B51 až B5d.

Príklad:

spracovanie troch sekvenčných vstupných súborov:

NPGANZ = 3

Potrebné moduly: B21001, B51001 - vstup a spracovanie viet
prvého súboru
B22001, B52001 - vstup a spracovanie viet
druhého súboru
B23001, B53001 - vstup a spracovanie viet
tretieho súboru

2. NPGGRU = /D1, D2, D3 ... až Dn/

kde parametre D1 ... Dn určujú dĺžku jednotlivých kľúčov v slabikách. Týmto parametrami sa generátoru zadáva počet a dĺžka jednotlivých kľúčov potrebných pre spracovanie. Maximálna dĺžka jedného kľúča môže byť 99 slabík, pričom počet kľúčov môže byť maximálne 45 /4455 slabík/. Generátor použije tieto parametre i pre tvorbu riadiacej tabuľky. Odpovedajúce počtu kľúčov generátor predpokladá existenciu takého istého počtu modulov spracovania smien skupinového kľúča /B41 až B4n/.

Príklad:

spracovanie podľa dvoch kľúčov /číslo dielne a pracoviska/, pričom číslo dielne je štvormiestne a číslo pracoviska

dvojmiestne:

HPGGRU = /4, 2/

Potrebné moduly: B41001 - spracovanie smeny čísla pracoviska
B42001 - spracovanie smeny čísla dieťa
B49001 - záver spracovania programu.

3. HPG = abecedný znak

Kde abecedný znak môže byť jeden z 26 znakov abecedy A až Z. Týmto parametrom sa zabezpečí vyvolanie riadiacej časti programu odpovedajúcej príslušnému abecednému znaku, ktorá pozostáva z vygenerovaných príkazov v jazyku COBOL. Vygenerovanú riadiacu časť programu generátor vloží na miesto jej volania v PROCEDURE DIVISION. V rámci jedného programu je možné použiť niekoľko riadiacich častí /max. 26/, pričom v príslušných častiach spracovania musí abecedný znak označenia jednotlivých názvov paragrafov súhlasiť so zvoleným znakom v parametri HPG.

Príklad:

názvy paragrafov programu budú začínať abecedným znakom C:
HPG = C

Potrebné moduly:	C21001	C41001	C51001	
	C22001	C42001	.	
	.	C43001	.	
	.	.	.	
	.	.	.	
	.	.	.	
	C28001	C49001	C58001	atď.

K dispozícii sú i ďalšie makropriказы, ktoré zabezpečujú riešenie zvláštnych prípadov:

4. HPGPRI = /p1, p2, p3, ... pd/

kde parametre p1 až pd sú číselné hodnoty priorít jednotlivých vstupných sekvencne čítaných súborov.

Parametre umožňujú programátorovi zadať /zmeniť/ prioritu jednotlivých súborov potrebnú pre prípad, že polia skupinových kľúčov viacerých súborov majú rovnakú hodnotu. Priorita v takomto prípade určí, z ktorého súboru sa

vyberie veta pre nasledovné spracovanie. Normálnym, sekvenčne čítaným vstupným súborom sa priradujú hodnoty priorít v rozsahu 10-89. Sekvenčným UPDATE súborom sú určené hodnoty 0-9. Najvyššiu prioritu má súbor, ktorému prináleží najmenšia hodnota parametra. V prípade, že je začený i parameter v NPGANZ, musí začený počet sekvenčne čítaných súborov súhlasiť s počtom jednotlivých parametrov uvedených v NPGPRI. Makroprikazom NPGPRI možno makroprikaz NPGANZ nahradiť.

Príklad:

zmena priority vstupných súborov:

NPGPRI = /10, 13, 11/

Poradie priorít vstupných súborov:

pôvodne	teraz
B21001	B21001
B22001	B23001
B23001	B22001

Príklad:

riadenie sekvenčného UPDATE:

NPGPRI = /8, 11, 15/

to znamená, že sekvenčný UPDATE súbor vstupuje v prvom vstupnom module.

5. NPGBLK = abecedný znak

kde abecedný znak môže byť jeden z 26 znakov abecedy A až Z. Týmto makroprikazom možno doziahnúť vygenerovanie viacerých paralelných riadiacich tabuliek. Z toho dôvodu je zápis makroprikazu dovolený vo WORKING STORAGE SECTION a LINKAGE SECTION. V prípade, že makroprikaz NPGBLK nie je uvedený, predpokladá sa NPGBLK = B. Každému makroprikazu NPGBLK musí byť priradený v PROCEDURE DIVISION makroprikaz NPG.

Príklad:

návvy paragrafov budú začínať abecedným znakom D:

NPGBLK = D

v PROCEDURE DIVISION sa musí vyskytnúť NPG = D.

6. NPGFLR = názov paragrafu

Makroprikaz umožňuje vetvenie programu do podprogramu s danou adresou v prípade vyskytnu chyby vo vzostupnom sorení súboru.

Chyba nastane, ak pole skupinového kľúča nasledujúcej vety /NGRU/ je menšie ako pole skupinového kľúča predchádzajúcej vety /AGRU/ alebo ak sa neuskutočnil prenos skupiny kľúčov do NGRU. Z vygenerovanej tabuľky NPG sa dá zistiť, ktorý súbor a veta v ňom zapríčinila chybu. V prípade, že makroprikaz nie je zadany a chyba nastane, program sa vetví na modul B20001.

Príklady:

1. Ošetrenie chyby uskutočniť v podprograme zadaným osobitným EKIT-om.
NPGFLR = B60999
2. Ošetrenie chyby uskutočniť v podprograme /module/ B20001.
Makroprikaz nie je potrebné zadať
3. V prípade chyby program ukončiť.
NPGFLR = B49001
4. Kontrolu zoradenia neuskutočňovať.
NPGFLR = NPG

7. NPGPRF = názov paragrafu

Program sa vetví na zadaný podprogram vždy po priebehu riadenia výberu vety. V poli NGRU sa už nachádza skupinový kľúč vety pripravenej na spracovanie, takže túto vetu alebo i skupinový kľúč NGRU je možné ešte pred vstupom do štvrtého bloku osobitne ošetriť, prípadne modifikovať.

Príklady:

1. Spoločný EKIT pre ošetrenie viet všetkých súborov.
NPGPRF = B30001
2. Špecifické EKIT-y pre osobitné ošetrenie viet jednotlivých súborov.
NPGPRF = YES
V prípade, že budú zadané štyri vstupné súbory a tento makroprikaz, generátor bude predpokladať v programe tieto EKIT-y: B 31001, B32001, B33001, B34001.

8. NPGHEU = názov paragrafu

Makroprikaz umožňuje vetvenie na zadanú časť programu pri prvom priebehu programu a vo všetkých prípadoch, ak nastala zmena hociktorého stupňa kľúča so skupinového kľúča a programátor

nastavil prepínač NPG-NEU /definuje generátor/ na hodnotu rozdielnu od nuly. Po návrate z tohoto EXIT-u generátor opäť nastaví prepínač na hodnotu NPG-NEU=0. Pri poslednom priebehu sa program na túto adresu nevetví.

Príklady:

1. Spoločný EXIT pre prípravu všetkých skupín

NPGNEU = B40502

2. Špecifické EXIT-y pre osobitnú prípravu každej skupiny.

NPGNEU = YES

V tomto prípade sa prepínač NPG-NEU nepoužíva. Generátor predpokladá podprogramy /moduly/:

B41501 - príprava spracovania skupiny najnižšieho stupňa

B42501 - príprava spracovania skupiny vyššieho stupňa

.

.

.

B4n501 - príprava spracovania skupiny najvyššieho stupňa.

9. NPGOPH = abecedný znak

kde abecedný znak je jeden z 26 znakov abecedy A až Z. Počiatočný stav hodnôt v riadiacích tabuľkách 1 vlastnej riadiacej časti sa prvým a ďalšími priebehmi programu poznajú. Týmto makroprikazom je možné v rámci programu znovu nastaviť počiatočný stav hodnôt podľa potreby 1 viac razy.

Príklad:

nastavenie počiatočných hodnôt vygenerovanej riadiacej tabuľky pre riadenie B:

NPGOPH = B

10. NPGIND = abecedný znak

kde abecedný znak je jeden z 26 znakov abecedy A až Z. Makroprikaz dáva možnosť vyberať údaje priamo z riadiacej tabuľky NPG, ktorú vygeneruje generátor. Pred aktivovaním makroprikazu sa musí číslo hľadaného súboru praniešť do nezhasnutého poľa AGRU-DATEI-ER. Po vykonaní NPGIND sa nachádza číslo poľa tabuľky NPG-TABELLE príslušiace danému súboru v poli NPG-INDEX. Indexovaním častí tabuľky NPG-TABELLE týmto indexom

je možné sprístupniť údaje z tejto tabuľky pre každý súbor. Tento makroprikaz je možné použiť v rámci programu viac razy, pre výber rôznych častí z riadiacej tabuľky NPG-TABELLE, vytvorenej generátorom NPG.

Príklad:

treba vytlačiť počet viet tretieho súboru:

```
MOVE 3 TO AGRU-DATEI-NR.
```

```
NPGIND = B
```

```
MOVE NPG-ANZAHL /NPG-INDEX/ TO RIADOK.
```

```
WRITE RIADOK.
```

5. Generovanie a preklad programov

Vstupom pre generátor NPG je program v jazyku COBOL, ktorý môže obsahovať volanie generátora formou riadiacich parametrov v INPUT-OUTPUT SECTION, FILE SECTION, WORKING-STORAGE SECTION alebo LINKAGE SECTION a v PROCEDURE DIVISION.

Generátor NPG verzia COBOL nahradzuje jednotlivé makropriказы generátora generovanými príkazmi /definíciami/. Generovanie sa vykonáva podľa kľúčových slov zadaných v makroprikazoch podľa parametrov a údajov v UPSI-slabike. Výstupom chodu generovania je užívateľský program s makroprikazmi generátora a ich príslušným riešením generátorom. Takýto program je možné použiť pre generovanie i viac razy. Pred novým generovaním môže užívateľ zmeniť hociktoré parametre.

Slabikou UPSI sa určí, či má byť vytlačený protokol s celým COBOL-programom spolu s vygenerovanými príkazmi. V prípade, že sa UPSI-slabikou tlač potlačí, tak sa so SYSLST vytlačia iba makropriказы generátora a chyby.

V protokole sa tlačia všetky užívateľské príказы v jazyku COBOL ako aj všetky generované príказы. Vygenerované príказы sú označené na 81. mieste riadku zostavy znakom "+", príказы volania generátora znakmi "N" alebo "P".

K očíslovaniu príkazov vygenerovaných generátorom sa používa číslo predchádzajúceho príkazu v jazyku COBOL, alebo ak bol bezprostredne pred príkazom vyvolania generátora zakódovaný iný príkaz jeho volania, tak sa použije číslo posledného vygenerovania riadku.

Bezprostredne pred prvým paragrafom modulu, ktorý musí vyhovovať metóde normovaného programovania, sa generuje nadpis modulu. Za počiatočný paragraf sa považuje paragraf, ktorý končí trojicou '001'. Ak bol zadán parameter NPGNEU = YES, tak v tomto prípade sa považujú za počiatočné paragrafy v štvrtom bloku i tie, ktoré končia trojicou '501'.

Vygenerovaný program, ktorý generátor ukladá na magnetický disk /príp. iné médium/, je v bežnom prípade okamžitým vstupom kompilátora COBOL /obr. č. 2/. V prípade, že v priebehu generovania sa zistili chyby, tak generátor ukončí po spracovaní a skontrolovaní všetkých príkazov v jazyku COBOL, parametrických príkazov NPG a uzavretí súborov prácu oznamom CANCEL. Ak v tom istom chode bol zaradený i preklad z jazyka COBOL, tak sa neuskutoční.

6. Záver

Osvojenie si jednotnej a efektívnej tvorby programov metódou normovaného programovania i s využitím generátora normovaného programovania NPG je jednoduché a rýchle. Postačuje trojdňové školenie na to, aby si i programátor - začiatočník dokázal osvojiť základné princípy tejto metódy a generátora NPG a použiť ich vo svojej programátorskej práci. Pracovníci Ústavu systémového inžinierstva priemyslu na Slovensku zabezpečujú podľa požiadaviek školenia normovaného programovania. Ako učebné pomôcky ku školeniu dostanú účastníci tieto príručky:

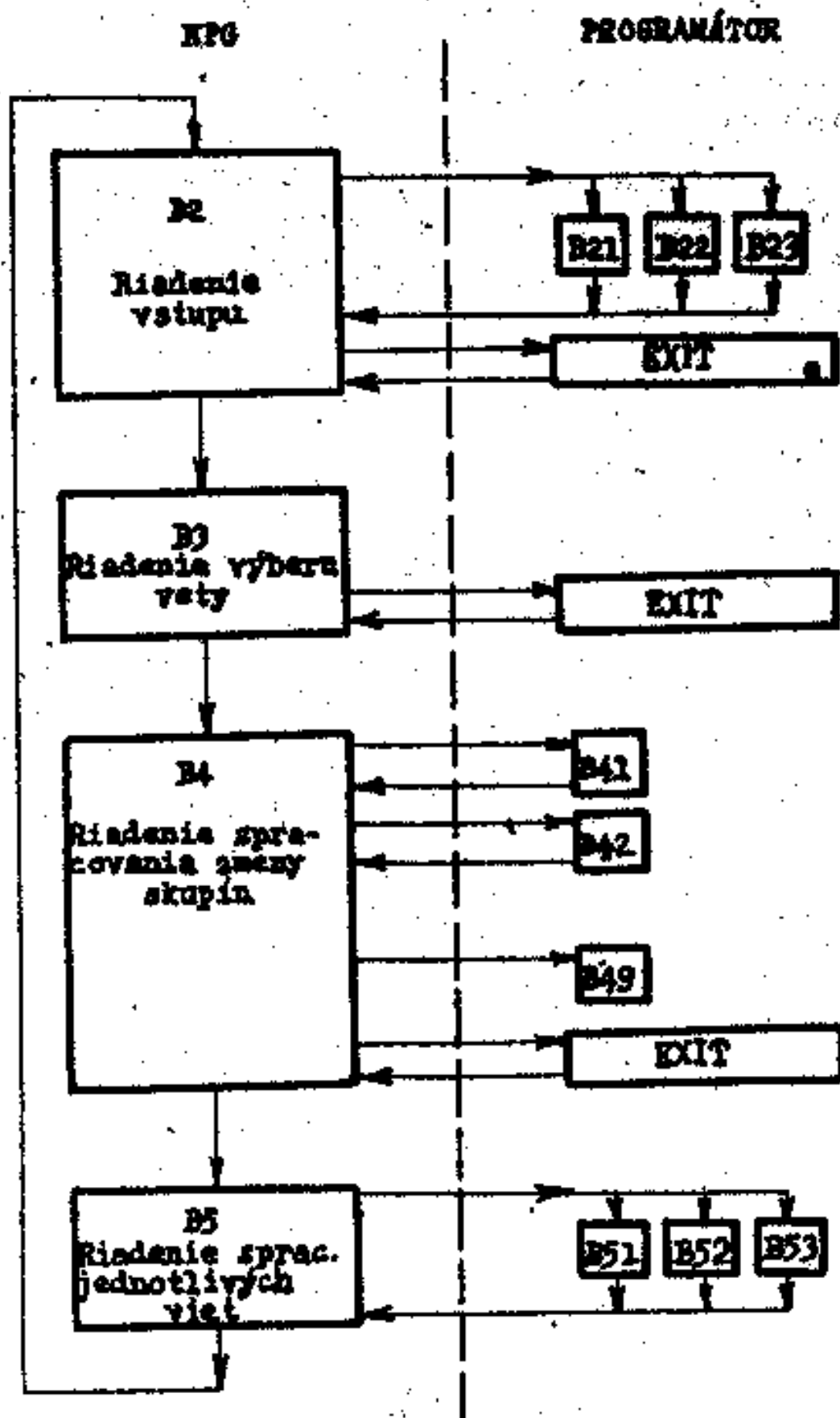
- a/ Metodika normovaného programovania
- b/ Metodika normovaného programovania /príkladová časť/
- c/ Generátor normovaného programovania /stručný popis/.

Organizácie, ktoré majú záujem získať generátor normove-
ného programovania EPG, si ho môžu zakúpiť po vyškolení svo-
jich pracovníkov a používať ho pre svoje potreby naobmedzene.
Podrobný návod ako používať generátor EPG je uvedený v Uži-
vateľskej príručke generátora EPG, ktorú užívateľ dostane pri
zakúpení generátora.

Podrobnejšie informácie, organizačné a iné podmienky tý-
kajúce sa školení, metódy EPG, generátora EPG, poskytuje odbor
realizácie výsledkov výskumu na úseku inžinierskych služieb
ústavu ÚSIP v Bratislave, Kráľovská 30.

Obr. č. 1.

Grafické znázornenie práce generátora KPG a programátora



Obr. 8. 2.

Príbeh generovania a prekladu programov

