

Ing. Jaroslav VACULÍN,
Tesla n.p. Valašské Meziříčí

STATGEN - SYSTÉM PRO STATISTICKÉ VYHODNOCOVÁNÍ SOUBORŮ DAT

Úvod.

V tomto příspěvku bude krátce popsán systém STATGEN určený pro statistické vyhodnocování souborů dat, který byl realizován na počítači GE425, instalovaném v n.p. Tesla Rožnov. Celková koncepce systému byla zvolena na základě několikaleťtých zkušeností s řešením úloh tohoto typu. Uživatel se obvykle omezuje na využívání jednodušších statistických metod aplikovaných na vhodně definované podsoubory, které vznikají výběrem z úplného souboru dat při splnění zadaných podmínek respektive po provedení vhodných aritmetických operací s jednotlivými položkami v souboru dat. Má-li být tedy systém dostatečně obecný, musí kromě toho, že uživateli poskytuje dostatečný výběr vhodných statistických metod, umožňovat uživateli provádět logické a aritmetické operace s daty a zadávat podmínky.

Koncepce, která byla zvolena pro realizaci systému STATGEN dává uživateli velmi bohaté možnosti manipulace s daty, zadávání podmínek, provádění různých nestandardních zpracování i rozšiřování systému o další statistické metody a při tom realizace systému je poměrně jednoduchá. Systém není realizován jako generátor v pravém slova smyslu, z hlediska uživatele se však jeví jako generátor s velmi bohatými možnostmi.

Konkrétní realizace systému STATGEN je sice dosti závislá na typu počítače, základní myšlenky jsou však použitelné obecněji a to nejen v úlohách statistického zpracování dat.

V tomto příspěvku se budu snažit popsat systém STATGEN z hlediska obecněji platných principů jeho realizace. Nebude uváděn popis jednotlivých metod v systému realizovaných. Pro objasnění bude uvedeno několik příkladů ilustrujících použití systému z hlediska uživatele.

Všeobecný popis systému STATGEN.

Systém STATGEN předpokládá pevný algoritmus zpracování orientovaný na statistické zpracování dat. V dalším textu jej budeme označovat jako systémový algoritmus. Činnost systémového algoritmu je realizována pomocí souboru podprogramů a funkcí, které budeme dále nazývat organizačními podprogramy. Uživatel se při zadávání vyhodnocení nemusí o algoritmus zpracování vůbec starat, stačí, když budou zachována určitá jednoduchá pravidla pro zadání vyhodnocení. Organizační podprogramy jsou psány v assembleru, poněvadž v řadě případů provádějí manipulaci s instrukcemi, které na úrovni vyšších programovacích jazyků by bylo možné provádět.

Vlastní statistické vyhodnocování dat je zajišťováno pomocí řady různých podprogramů, přičemž pro jednotlivé statistické metody jsou vypracovány samostatné podprogramy a to výhradně v jazyku FORTRAN. Dále budeme tyto podprogramy označovat jako výpočetní. Soubor výpočetních podprogramů je možno podle potřeb uživatele dále rozšiřovat bez nároku na detailní znalost systému.

Organizační a výpočetní podprogramy systému STATGEN budeme nazývat systémovými podprogramy. Všechny systémové podprogramy jsou spolu úzce svázány a mimo systém STATGEN nejsou použitelné.

Zadání statistického vyhodnocení souboru dat pomocí systému STATGEN zapisuje uživatel jako hlavní program v programovacím jazyku FORTRAN. Pokud uživatel využívá pouze standardních možností systému STATGEN, zapíše se celé zadání jako posloupnost volání vhodných systémových podprogramů

s příslušnými parametry. V případě potřeby má však uživatel možnost použít úseky psané v programovacím jazyku FORTRAN, ve kterých je možno používat prakticky všech příkazů jazyka FORTRAN. Tím je zajištěna velká pružnost systému STATGEN, což se potvrdilo při řešení řady praktických úloh.

System STATGEN pracuje výhradně s proměnnými typu REAL. Pouze některé parametry systémových podprogramů musí být typu INTEGER nebo LOGICAL.

Systemový algoritmus.

Systemový algoritmus předpokládá jeden vstupní soubor, který obsahuje data určená k vyhodnocení. Nositelem souboru může být buď magnetická páska nebo 80 sloupcové děrné štítky. Předpokládá se, že každý záznam na vstupním souboru se vztahuje k jednomu pozorování jevu, který chceme vyhodnotit a že každý záznam má stejnou váhu. Lze však řešit i ty případy, kdy tyto podmínky nejsou splněny. Vstupní soubor nemusí být seříděn. V případě potřeby má možnost uživatel číst jeden případně i více vedlejších vstupních souborů, přičemž jejich záznamy může dle vhodných klíčů připojovat k odpovídajícím záznamům hlavního vstupního souboru /v tomto případě je ovšem seříděnost souborů dle stejných klíčů nezbytností/. Funkce systémového algoritmu je však vždy řízena načtením záznamu z hlavního vstupního souboru.

Čtení hlavního vstupního souboru zajišťuje příkaz

CALL SOUBMP (a,b,c,d)

pro soubor na magnetické pásce nebo příkaz

CALL SOUBDS (e,f)

pro soubor na 80 sloupcových děrných štítcích. Oba uvedené podprogramy patří do skupiny organizačních podprogramů. Parametry a, b, c, e popisují záznam, parametry d, f, jsou taxty délky maximálně 120 znaků, které se tisknou v záhlaví každé stránky na výstupní sestavě. V zadání STATGEN musí být uveden právě jednou buď příkaz pro čtení souboru z mag. pásky nebo pro čtení souboru z děrných štítků. Při prvním vstupu do podprogramu se provede kontrola parametrů a v případě, že není zjiš-

těna chyba, provede se rezervace potřebné oblasti pro uložení načteného záznamu a pro práci se souborem /viz dále/, otevření vstupního souboru a načtení prvního záznamu. Při každém následujícím vstupu je pak načten další záznam do rezervované oblasti. Jakmile je záznam načten, řízení je předáno na příkaz následující ze příkazem pro čtení vstupního souboru v zadání STATGEN. Následující příkazy zajišťují zpracování načteného záznamu. Zpracování záznamu je ukončeno příkazem

CALL STOP

který musí být v zadání STATGEN uveden právě jednou. Tento příkaz zajistí návrat do podprogramu čtení vstupního souboru pro načtení dalšího záznamu a v případě zjištění konce souboru zajistí tisk výsledků zpracování a předání řízení na příkaz následující ze CALL STOP v zadání STATGEN. Standardním výstupním souborem systému je tiskárna. V případě potřeby je však možno použít vedlejší výstupní soubory na magnetické páse.

Před příkazem čtení vstupního souboru lze v zadání STATGEN provést otevření vedlejšího vstupního nebo výstupního souboru, definovat počáteční hodnoty pomocných proměnných a pod. Tuto část zadání STATGEN budeme nazývat úsekem předběžného zpracování. Nemá v něm být použit žádný systémový podprogram, jinak však je možno použít prakticky všech příkazů jazyka FORTRAN. Úsek předběžného zpracování se provádí jen jednou před otevřením hlavního vstupního souboru. Nemá být v zadání vůbec použit.

Část zadání STATGEN mezi příkazem pro čtení vstupního souboru a příkazem CALL STOP popisuje vlastní zpracování záznamu. Budeme je nazývat úsekem zpracování záznamu. Provádí se opakovaně pro každý záznam načtený z hlavního vstupního souboru. Statistické zpracování je charakteristické tím, že během zpracování vstupního souboru dat se v paměti ukládají jen mezivýsledky / např. $\sum_{i=1}^n x_i$, $\sum_{i=1}^n x_i^2$ a počet záznamů n při výpočtu střední hodnoty a směrodatné odchylky / a konečný výsledek lze vypočítat a vytisknout až po zpracování celého vstupního souboru /např. střední hodnota $\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ /. Z uvedeného je zřejmé, že každý podprogram, realizující statistické

ké zpracování dat bude vyžadovat ke své činnosti určité místo v operační paměti. Systém STATGEN je vyřešen tak, že rezervace potřebného místa v paměti se děje automaticky na základě zadaných parametrů, uživatel se o rezervaci nemusí vůbec starat. Systém STATGEN rovněž automaticky zajišťuje tisk výsledků po uzavření vstupního souboru. Přitom je zajištěno, že každý podprogram realizující statistické zpracování je možno v zadání STATGEN vyvolávat vícenásobně na různých místech zadání. Veškerá tato činnost je zajištěna organizačními podprogramy, se kterými uživatel systému nemusí vůbec pracovat. Úsek zpracování záznamu musí být v zadání STATGEN uveden povinně. Podprogramy, které realizují statistické zpracování souboru dat patří do skupiny výpočetních podprogramů a nesmí být mimo úsek zpracování záznamu použity. Proto je budeme označovat jako výpočetní podprogramy zpracování záznamu.

Za příkaz CALL STOP je řízení předáno po uzavření vstupního souboru a vytištění výsledků zpracování realizovaného výpočetními podprogramy zpracování záznamu. Tuto část zadání STATGEN budeme označovat úsek dodatečného zpracování. Proveďte se jen jednou. Musí být ukončen příkazem END. V úseku dodatečného zpracování lze např. uzavřít vedlejší vstupní nebo výstupní soubory a dále lze provést další zpracování dat, která byla uložena do paměti v úseku zpracování záznamu /např. testování, zda rozdíl mezi četnostmi ve dvou podsouborech je statisticky významný a pod./. K tomuto zpracování je v systému STATGEN k dispozici několik výpočetních podprogramů, které jsou použitelné pouze v tomto úseku, a proto je budeme označovat výpočetní podprogramy dodatečného zpracování. Úsek dodatečného zpracování není povinný.

Zadání vyhodnocení pomocí systému STATGEN z hlediska uživatele lze znázornit následujícím schématem:

seznam monitorových příkazů pro vyvolání překladače

[úsek předběžného zpracování]

CALL SCUBMP (a,b,c,d) nebo CALL SCUBDS (e,f)

úsek zpracování záznamu

CALL STOP

[úsek dodatečného zpracování]

END

monitorové příkazy pro provedení úlohy

V jednotlivých úsecích lze kromě vyvolávání výpočetních podprogramů použít prakticky všech příkazů jazyka FORTRAN, je však nezbytně nutno dodržet zásadu, že řízení nesmí být předáno z jednoho úseku do jiného úseku, poněvadž by tím byla porušena činnost systémového algoritmu, což by vedlo k nepředvídatelným výsledkům. Popsaná činnost systémového algoritmu je popsána blokovým schématem na obr.1.

Organizace operační paměti.

Na počátku operační paměti přidělené zpracovávané úloze je uložen hlavní program, který vznikne kompilací sadání STATGEN. Za ním jsou zaváděcími programem uloženy všechny potřebné podprogramy. Zbývající část paměti je při běhu programu přidělována jednotlivým systémovým podprogramům dle jejich požadavků. Aby v případě potřeby byla dostupná uživateli, je definována jako oblast COMMON bez označení. Zcela na počátku této oblasti je podprogramem čtení vstupního souboru rezervována oblast pro uložení načteného záznamu, za ní pak následuje tabulka souboru, seznam řídicích slov pro přenos dat a 2 vyrovnávací oblasti pro přenos dat z periferního zařízení do operační paměti. Délka této oblasti závisí na typu vstupního souboru, minimálně je ale 1000 slov, poněvadž se počítá s možností využití této oblasti k jiným účelům v době, kdy vstupní soubor dat je uzavřen /např. jako pracovní oblast pro ukládání mezivýsledků při tisku výstupní sestavy/. Další část paměti je pak přidělována jednotlivým výpočetním podprogramům dle jejich požadavků. Od konce paměti jsou generovány při běhu některé části programu /např. vyvolávací posloupnosti pro tisk - viz dále/. Rozhraní mezi těmito dvěma oblastmi není pevně stanoveno. V okamžiku, kdy by výpočetní program požadoval větší oblast v paměti než je k dispozici, je tento požadavek zamítnut a volání tohoto podprogramu je zcela vyloučeno ze

zpracování. Je však možno přijat ke zpracování jiné podprogramy, které mají menší požadavky na paměť. Přidělování operační paměti řídí organizační podprogram RET zcela automaticky bez vědomí uživatele. Činnost tohoto podprogramu bude podrobněji popsána dále. Rozčlenění operační paměti je schematicky znázorněno na obr. 2.

Přístup k záznamu z hlavního vstupního souboru.

Podprogram čtení vstupního souboru uloží každý načtený záznam do rezervované oblasti v přesně takovém tvaru, v jakém je zapsán na vstupním mediu. Uživatel by sice se záznamem mohl pracovat přímo přes oblast COMMON, to však ve většině případů nebude vhodné, neboť STATGEN pracuje s proměnnými typu REAL, kdežto nejčastější forma zápisu na externím mediu je znaková. Proto jsou mezi systémové podprogramy začleněny funkce R a RR, které umožňují výběr kterékoli položky ze záznamu i potřebnou konverzi zobrazení. Obě uvedené funkce jsou libovolněkrát použitelné v zadání STATGEN a volají se příkazy

R (a,b,c) nebo RR (a,b,c)

kde parametr a definuje typ konverze zobrazení /výběr položky bez konverze s doplněním vedoucích nul, výběr položky s konverzí znakového zobrazení na real nebo výběr položky s konverzí integer - real/, parametry b, c udávají adresu prvního a posledního znaku položky ve vstupním záznamu. Funkce R vyžaduje, aby parametry a, b, c byly konstanty. To umožňuje při prvním vyvolání funkce R zgenerovat optimální posloupnost instrukcí pro výběr případně konverzi položky, takže zpracování položky je velmi efektivní. Funkce RR toto omezení nemá, bude tedy vhodná i v případě práce s masivem, výběr položky je však pomalejší vzhledem k interpretačnímu způsobu zpracování.

Uživatel s jednotlivými položkami ze vstupního záznamu pracuje jako s funkcemi R nebo RR, přičemž tyto funkce může použít jako parametry výpočetních podprogramů, v aritmetických nebo příkazovacích příkazech, relačních testech a pod.

Výpočetní podprogramy zpracování záznamu.

Jak vyplývá z popisu systémového algoritmu, činnost každého výpočetního podprogramu zpracování záznamu probíhá ve třech fázích. Tomu také odpovídá rozdělení každého výpočetního podprogramu na tři samostatné podprogramy typu SUBROUTINE.

Fáze 1. Tato fáze se provede jen jednou při prvním vyvolání výpočetního podprogramu z určitého místa v zadání STATGEN. Subrutina realizující tuto fázi zpracování musí mít název, pod kterým je vyvoláván příslušný výpočetní podprogram uživatelem. Nejdříve je nutno provést formální kontrolu hodnot těch skutečných parametrů, které se zadávají jako konstanty, pak je nutno vypočítat požadavek na rezervaci paměti a vyvolat organizační podprogram RET, který zajistí všechny další činnosti:

```
CALL RET (a,b,c,d,e,f)
RETURN
```

Parametr a udává výsledek kontroly skutečných parametrů výpočetního podprogramu, parametr b obsahuje znakově uložený název výpočetního podprogramu /je použit při tisku indikace na výstupní sestavě/. Parametry c, d jsou názvy příslušných subrutin pro fázi 2,3 a musí být definovány jako EXTERNAL, čímž bude zajištěno, že na místě parametrů c, d se uloží adresy vstupů do těchto subrutin. Parametr e udává vypočtený požadavek na rezervaci pracovní oblasti pro výpočetní podprogram. Parametr f udává počet parametrů, které výpočetní podprogram předpokládá. Příkaz RETURN je v tomto případě použit jako další parametr podprogramu RET a umožňuje určit adresu v zadání STATGEN, odkud byl výpočetní podprogram uživatelem vyvolán. Podprogram RET nikdy nevrací řízení na následující příkaz. Poněvadž činnost příkazu RET je rozhodující pro práci systému STATGEN, bude dále podrobněji popsána.

1. Pomocí parametru f a následujícího příkazu RETURN je určena adresa v zadání STATGEN, odkud byl příslušný podprogram vyvolán. Tato adresa je spolu s názvem

podprogramu /parametr b/ vytištěna na tiskárně, aby v případě zjištění chyby bylo možno nalézt chybný příkaz v zadání.

2. Kontroluje se, zda počet skutečných parametrů výpočetního podprogramu je stejný jako požadovaný počet parametrů /parametr f/. V případě nesouhlasu přejde se na bod 9.
3. Jestliže indikátor chyby /parametr a/indikuje chybné hodnoty parametrů, přejde se na bod 9.
4. Kontroluje se, zda vstupní soubor dat je otevřen /t.j. zda podprogram je použit v úseku zpracování záznamu/. Není-li otevřen, přejde se na bod 9.
5. Kontroluje se, zda v paměti je dostatek místa na rezervaci pracovní oblasti pro podprogram /parametr e/. Není-li, přejde se na bod 9. V opačném případě se potřebná paměť zareservuje a vynuluje.
6. Na konci paměti /viz obr. 2/se generuje vyvolávací posloupnost pro tisk výsledků, která obsahuje:
 - a. Adresu, která byla určena dle bodu 1. Je nutná k tomu, aby tisk výsledků bylo možno provést v tom pořadí, ve kterém jsou výpočetní podprogramy zadány uživatelem /vzhledem k tomu, že část zadání STATGEN může být podmíněna a podmínka nemusí být při zpracování prvního záznamu splněna, jsou vyvolávací posloupnosti pro tisk generovány obecně v náhodném pořadí/.
 - b. Instrukci pro vyvolání subrutiny pro fázi 3 /tisk výsledků/, jejíž adresa je určena parametrem d podprogramu RET.
 - c. Seznam adres skutečných parametrů výpočetního podprogramu, shodný se seznamem ve vyvolávací posloupnosti podprogramu v zadání STATGEN, která byla sestavena kompilátorem FORTRANU.
 - d. Adresa pracovní oblasti přidělené výpočetnímu podprogramu dle bodu 5 a představující další parametr subrutiny pro fázi 3.

- e. Skokovou instrukci pro návrat do podprogramu STOP, který zajišťuje provádění tisku výsledků ve správném pořadí.
7. Vyvolávací posloupnost výpočetního podprogramu v zadání STATGEN, která byla sestavena kompilátorem FORTRANU, se upraví tak, aby při každém dalším vstupu se vyvolávala subrutina pro fázi 2:
 - a. Adresa subrutiny pro fázi 1 v instrukci skoku do podprogramu je nahrazena adresou subrutiny pro fázi 2, která je dána parametrem c podprogramu RET.
 - b. Adresa posledního skutečného parametru je nahrazena adresou pracovní oblasti přidělené dle bodu 5. Tato změna je možná, neboť jako poslední parametr každého výpočetního podprogramu se uvádí text, který se tiskne na výstupní sestavě během fáze 3, pro niž však byl uchován (viz bod 6.). Ve fázi dvě se nepoužívá.
 8. Řízení je předáno na adresu určenou dle bodu 1, takže se bezprostředně vyvolá zpracování fáze 2. To je nezbytné, neboť fáze 1 se provádí v době, kdy záznam ze vstupního souboru je načten a musí být tedy zpracován.
 9. V případě zjištění chyby se vypíše na výstupní sestavě indikace zjištěné chyby, celá vyvolávací posloupnost v zadání STATGEN generovaná překladačem FORTRAN je nahrazena instrukcemi "nedělej nic" a řízení je předáno na adresu určenou dle bodu 1 /viz bod 8/.

Fáze 2. Tato fáze se provádí opakovaně pro každý záznam načtený ze vstupního souboru. Subrutina provádějící zpracování v této fázi ukládá do pracovní oblasti přidělené ve fázi 1 data dle algoritmu daného statistickou metodou, kterou příslušný výpočetní podprogram realizuje. Subrutina pro fázi 2 je vyvolávána z původní vyvolávací posloupnosti ze zadání STATGEN s upraveným posledním parametrem /viz fázi 1/.

Fáze 3. Tato fáze se provádí pouze jednou po uzavření vstupního souboru. Vyvolání subrutiny pro fázi 3 zajišťuje organizační podprogram STOP, který postupně předává řízení na vyvo-

lávací posloupnosti generované ve fázi 1 na konci paměti, přičemž zajišťuje, že jsou prováděny ve stejném pořadí jako jsou příslušné výpočetní podprogramy uvedeny v sadání uživatele. Subrutina realizující fázi 3 vytiskne standardní záhlaví, hodnoty zadáných parametrů a text, který se zadává jako poslední parametr /délka textu může být 0 až 120 znaků/. Pak podle algoritmu daného statistickou metodou, kterou příslušný výpočetní podprogram realizuje se zpracují data uložená do pracovní oblasti ve fázi 2 a provede se tisk výsledků. Výpočty se provádějí tak, aby nedošlo k porušení dat uložených do pracovní oblasti, poněvadž se počítá s jejich využitím v úseku dodatečného zpracování. V případě potřeby je na počátku oblasti COMMON k dispozici pracovní oblast délky 1000 slov, která je použitelná pro ukládání mezivýsledků.

Pro ilustraci uvedeme popis jednoho z nejjednodušších výpočetních podprogramů v úseku zpracování záznamu - podprogramu TRDC. Podprogram se vyvolá instrukcí

CALL TRDC (a,b,c,d,e)

Podprogram počítá absolutní a relativní četnosti ve skupinách stejné šířky. Parametr a je proměnná, dle jejíž hodnoty se pozorování zařadí do příslušné skupiny. Parametry b, c, d jsou konstanty definující skupiny /dolní a horní hranice a šířka skupiny/. Pozorování k je zařazeno do i-té skupiny, jestliže platí $b + (i - 1) * d \leq a_k < c + i * d$

V případě $a_k < b$ nebo $a_k \geq c$ pozorování k není vůbec započítáno. Nejjednodušší způsob použití systému STATGEN pro vyhodnocení souboru dat /bez úseku předběžného a dodatečného zpracování a bez podmínek/ si ukážeme na následujícím příkladě.

Příklad 1. Na magnetické páse UONI /20 slov v záznamu, 20 záznamů v bloku/ jsou zapsány výsledky psychologických testů provedených s uční prvního až třetího ročníku. Páska je zapsána znakově, přičemž 1 záznam obsahuje výsledky všech testů pro jednoho učně. Ve znaku 1 v záznamu je uvedeno číslo ročníku /1,2,3/, ve znaku dvě je uvedeno pohlaví učně /1 - dívky, 2 - chlapci/, ve znacích 25,26 jsou uvedeny výsledky testu technického porozumění, které mohou nabývat hodnot 1 až 30.

Zajímají nás četnosti ve skupinách 1-5, 6-10 atd až 26-30. Řešení:

```
CALL SOUBMP (20,20,'UCNI','TECH.POROZUMENI!')
```

```
CALL TRDC (R(1,25,26) ,1.,31.,5.,'!')
```

```
CALL STOP
```

```
END
```

Podmínky v úseku zpracování záznamu.

Velmi častým požadavkem uživatele je provést vyhodnocení určitého podsouboru vybraného z úplného souboru dle zadáných podmínek. Systém STATGEN musí tedy umožňovat, aby některé části úseku zpracování záznamu se prováděly jen pro ty záznamy, u nichž je splněna zadaná podmínka. Podmínka se ovšem musí uplatnit pouze ve fázi 2. To je zajištěno organizačními podprogramy. Pouze v případě, kdy by podmínka nebyla splněna pro žádný záznam neproběhne ani fáze 3, což ovšem není na závadu, poněvadž v tomto případě výpočet nemá smysl.

Systém STATGEN dává velmi bohaté možnosti podmínek. Jednou možností je použití příkazů aritmetického nebo logického IF nebo větvení pomocí počítaného GOTO. Nevýhodou je, že v těchto případech není při tisku výstupní sestavy vytištěna žádná zpráva o podmínění úseku zadání, v řadě případů to však nemusí vadit, poněvadž toto označení může uživatel uvést v textu u jednotlivých výpočetních podprogramů. Druhou možností je použití organizačních podprogramů PODM a KPOD, pomocí kterých lze ohraničit počátek a konec podmíněného úseku. Při tisku zajišťují tyto podprogramy vytištění zprávy o počátku a konci podmíněného úseku na výstupní sestavě. Uvnitř každého podmíněného úseku mohou být vloženy další podmíněné úseky buď s použitím podprogramů PODM - KPOD nebo za použití podmínek na úrovni FORTRANU.

Podmíněný úsek začíná příkazem

```
CALL PODM (l,m,t)
```

a končí příkazem CALL KPOD (m)

kde parametr l je logická proměnná nebo logický výraz, který určuje, zda podmíněný úsek se provede nebo ne. Parametr m

je konstanta určující číslo podmíněného úseku a umožňující určit ukončení podmíněného úseku v případě vložení několika úseků do sebe. Parametr t je text, který se tiskne na výstupní sestavě. Podprogram PODM pracuje ve třech fázích obdobně jako výpočetní podprogramy úseku zpracování záznamu, činnost v těchto fázích je však poněkud odlišná.

Podprogram PODM ve fázi 1 postupně prohledává program, který generoval překladač PORTRANU na základě zadání uživatele a hledá odpovídající příkaz KPOD dle čísla podmínky. Je-li nalezen, je na místo parametru m uložena adresa odpovídajícího příkazu KPOD, v opačném případě je nastaven indikátor chyby. Další zpracování je pak provedeno podprogramem RET. Ve fázi 2 v případě, že podmínka není splněna, je podle adresy uložené v parametru m řízení předáno na příkaz následující za odpovídajícím příkazem KPOD, tedy příkazy v podmíněném úseku nejsou pro tento záznam provedeny. Jestliže záznam splňuje podmínku, je řízení předáno na příkaz následující za PODM, takže podmíněný úsek se provede. Ve fázi 3 je prováděn tisk zprávy o počátku podmíněného úseku a text uvedený jako parametr t.

Podprogram KPOD ve fázi 2 nezajišťuje žádnou činnost, řízení předává bezprostředně na následující příkaz v zadání STATGEN, ve fázi 3 pak provádí tisk zprávy o ukončení podmíněného úseku.

Příklad 2. U souboru UCNI dle příkladu 1 se má kromě výpočtu četnosti pro celý soubor vypočítat četnost pro následující podsoubory:

- a. celý 1. ročník
- b. dívky 1. ročník
- c. chlápci 1. ročník

Řešení: CALL SOUBMP (20,20, 'UCNI', 'TECH.POROZUMENI!')
TP = R(1,25,26)
POH = R(1,2,2)
CALL TRDC(TP,1.,31.,5., 'CELY SOUBOR!')
CALL PODM (R(1,1,1).EQ.1.,1, 'ROCNIK 1!')
IF (POH.EQ.1.) CALL TRDC (TP,1.,31.,5., 'DIVKY!')

```

IF (POH.EQ.2.) CALL TRDC (TP,1.,31.,5., 'CHLAPCI!')
CALL KPOD (1)
CALL STOP
END

```

Výpočetní podprogramy dodatečného zpracování.

Tyto systémové podprogramy jsou použitelné pouze v úseku dodatečného zpracování a zpracovávají data, která byla do paměti uložena v úseku zpracování záznamu. Příkladem použití může být testování shody skutečné četností a teoretickou četností, testování zda rozdíl mezi dvěma podsoubory je statisticky významný a pod. V systému STATGEN je několik podprogramů, které toto dodatečné zpracování realizují a uživatel má možnost v případě potřeby rozšířit systém o další výpočetní podprogramy dodatečného zpracování dle svých potřeb bez nutnosti podrobné znalosti systému. V úseku dodatečného zpracování se při každém volání provádí podprogram jen jednou.

Podprogramy úseku dodatečného zpracování rovněž nevyžadují ke své činnosti samostatné pracovní oblasti v paměti. V případě potřeby je k dispozici pracovní oblast na počátku oblasti COMMON, která je určena k práci se vstupním souborem a tudíž v úseku dodatečného zpracování je volná. Každému výpočetnímu podprogramu je však nutno zpřístupnit oblast v paměti, v níž má pracovat. Řešení tohoto problému bude blíže popsáno. Každý výpočetní podprogram zpracování záznamu, s jehož daty bude uživatel pracovat v úseku dodatečného zpracování je nutno opatřit číslem příkazu. Kompilátor FORTRAN GE-400 sice neumožňuje uvádět číslo příkazu jako parametr podprogramu, lze to však obejít použitím příkazu

```
ASSIGN m TO I
```

který zajistí uložení adresy příkazu s číslem m do proměnné I, kterou pak uvedeme jako parametr výpočetního podprogramu dodatečného zpracování.

Vlastní zpřístupnění oblasti m daty pak zajišťuje podprogram SUBTST, který patří do skupiny organizačních podprogramů. Podprogram SUBTST je nutno vyvolat na počátku výpo-

četního podprogramu dodatečného zpracování. Prohledává paměť od adresy určené příkazem ASSIGN až najde adresu přidělené pracovní oblasti, která byla do původní vyvolávací posloupnosti generované překladačem FORTRAN uložena ve fázi 1. Je nutno si uvědomit, že v obecném případě nebude vstup do výpočetního podprogramu zpracování záznamu na adrese určené příkazem ASSIGN, ale na adrese vyšší, neboť nejdříve se musí provést vyčíslení hodnot skutečných parametrů podprogramu, pokud jako parametry jsou použity funkce nebo aritmetické či logické výrazy. Podprogram SUBTST je vyřešen tak, že umožňuje zpřístupnit současně několik oblastí. Jednotlivé položky přidělené oblastí jsou pak dostupné jako funkční hodnota funkce SUB, což je vedlejší vstup do podprogramu SUBTST.

Pro ilustraci uvedeme popis jednoho z nejjednodušších výpočetních podprogramů dodatečného zpracování - podprogramu TSTTC, který provádí χ^2 test a Kolmogorov-Smirnovův test dobré shody skutečné četnosti s četností teoretickou. Skutečné četnosti se v úseku zpracování záznamu určí pomocí podprogramu TRDC, kterému uživatel přidělí číslo m . Srovnání s teoretickou četností se v úseku dodatečného zpracování provede pomocí příkazů

```
ASSIGN m TO I
```

```
CALL TSTTC (I, c1, c2, c3, ..., cn, t)
```

kde $c_1, c_2, c_3, \dots, c_n$ jsou konstanty udávající teoretické četnosti ve skupinách a t je text, který se vytiskne na výstupní sestavě. Musí být splněna podmínka $\sum_{i=1}^n c_i = 1$ a současně počet zadaných skupin u podprogramu TRDC musí být shodný s počtem zadaných teoretických četností. Není-li tato podmínka splněna, vytiskne se na výstupní sestavě chybová zpráva a zpracování není provedeno. V každém případě je však řízení předáno na příkaz následující v zadání STATGEN. Podprogram TSTTC pracuje s proměnným počtem parametrů. Ke zpracování proměnného počtu parametrů jsou určeny systémové podprogramy PARTST a KPARAM.

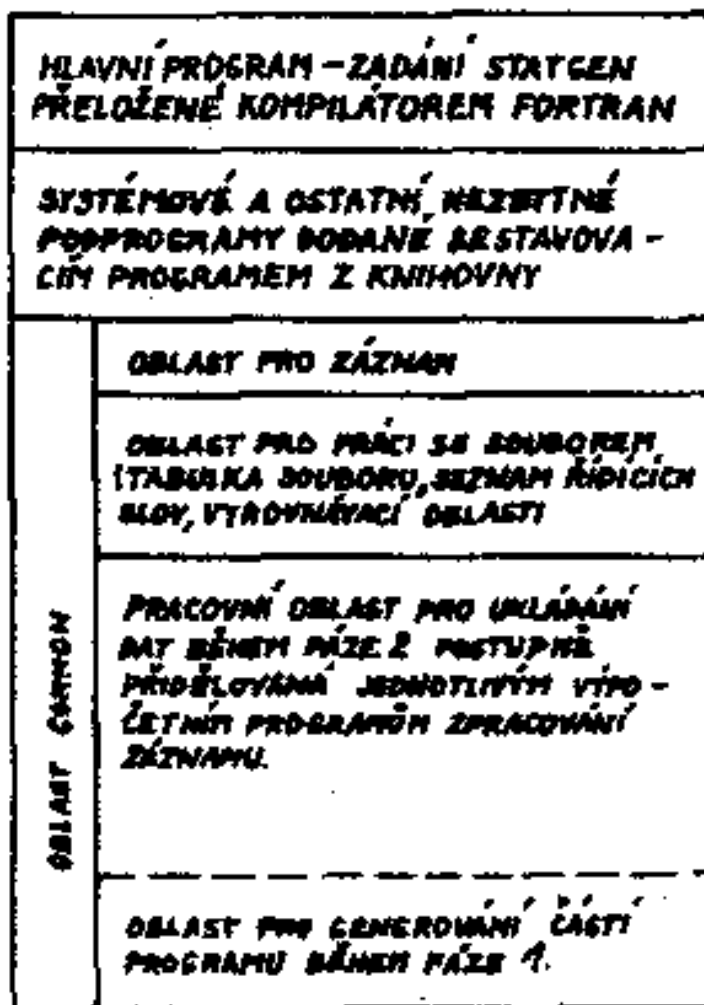
Příklad 1. Počet živě narozených dětí kolísá v jednotlivých měsících roku. V určitém okrese byl proveden průzkum narozených dětí a jedním z úkolů je prověřit, zda sezonní výkyvy

v okrese se statisticky významně liší od sezonních výkyvů v celostátním průměru, které jsou známy. Na magnetické páse POPULACE /25 záznamů v bloku, 15 slov v záznamu/ jsou shromážděny údaje o všech narozených dětech v okrese za několik posledních roků tak, že každý záznam obsahuje údaje vztahující se k jednomu dítěti. Měsíc narození dítěte je ve znacích 8,9.
Řešení: CALL SOUBMP (25,15,'POPULACE', 'PRUZKUM POPULACE!')

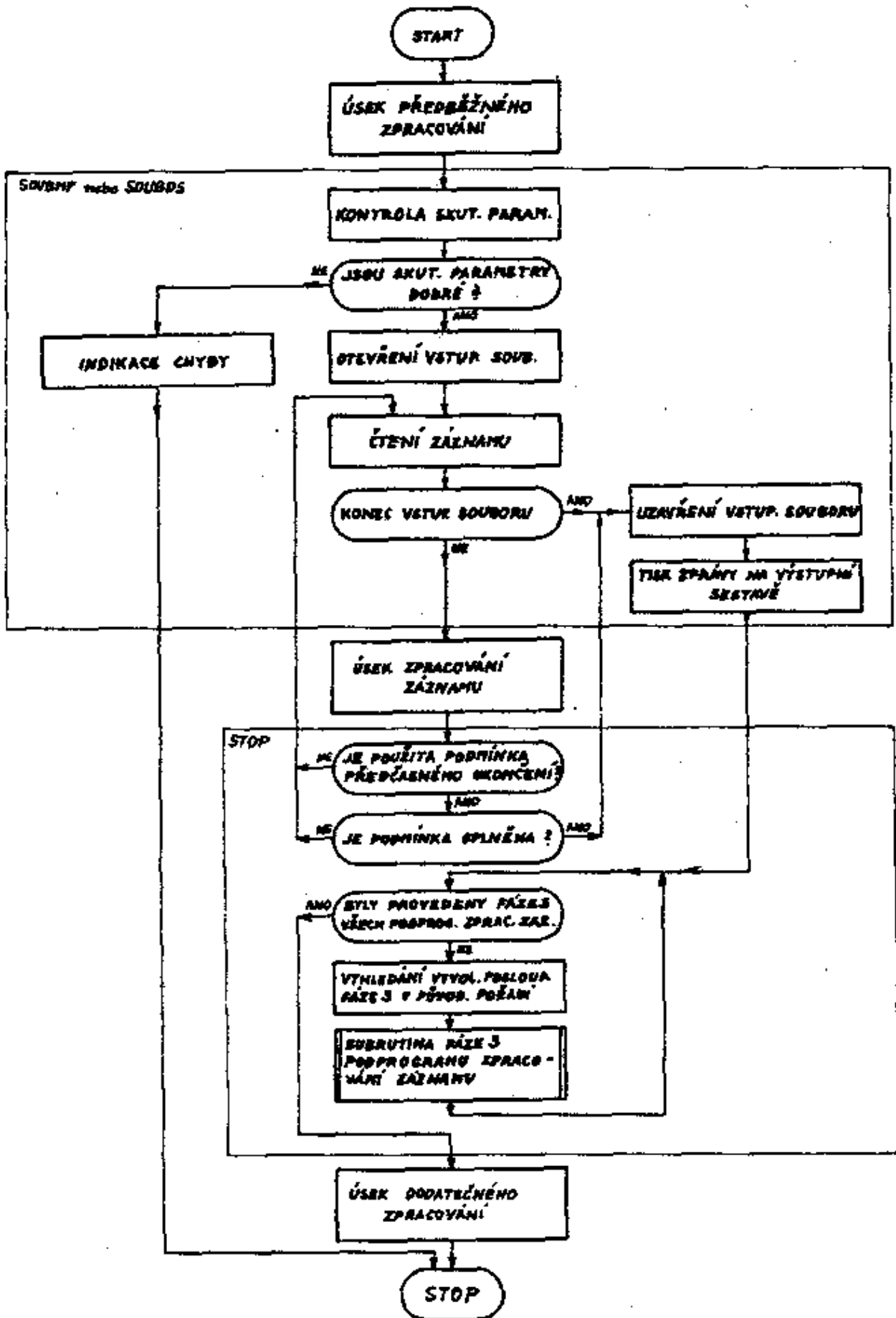
```
10 CALL TRDC (R(1,8,9),1.,13.,1., 'DETI NAROZENÉ V MES!')  
CALL STOP  
ASSIGN 10 TO I  
CALL TSTTC (I, .0839, .0791, .0902, .0915, .0864,  
- .0845, .0804, .0828, .0793, .0741, .0775, .0903,  
- 'SROVNANI S CELOSTATNIM PRUMEREM!')  
END
```

Závěr.

Smyslem tohoto příspěvku nebylo dát ucelený přehled o systému STATGEN. Řada možností zde nebyla vůbec uvedena, nebo byla uvedena bez bližšího vysvětlení. Hlavní snahou bylo ukázat možnost, jak lze poměrně jednoduchými prostředky vytvořit systém, který uživateli dává velmi bohaté možnosti.



OBRAZ. ORGANIZACE OPERAČNÍ PAMĚTI PŘEDLOŽENÉ OPER. SYSTÉMEM PRO PRÁCI STATGEN.



OBRAZ 1. HRUBÉ BLOKOVÉ SCHÉMA ČINNOSTI SYSTÉMOVÉHO ALGORITMU