

## ZÁZNAM Z DISKUSE O PARAMETRICKÝCH PROGRAMECH.

V únoru 1982 se v Ostravě uskutečnila diskuse k problematice parametrických programů (v dalším textu zkráceně PP). Diskuse se zúčastnili: ing. V. Čimbura, RNDr. P. Jiříček, J. Josíško, p.m., ing. J. Klečka, ing. E. Kopčil, ing. K. Metzl, ing. V. Orlik, RNDr. J. Peška, ing. Z. Rusín a ing. D. Streit. Následující text představuje koláž z myšlenek vyslovených v diskusi. Jednotlivé odstavce jsou uváděny bez jmen jejich autorů a celý text je tedy kolektivním příspěvkem. Jednotlivé diskusní příspěvky jsou odděleny znaky -----.

### 1. Poznámky k pojmu parametrický program.

Pojem PP se v ČSN 369001 Názvosloví číslicových a analogových počítačů nenachází. Měli bychom si aspoň připomenout termíny, které nějak a pojmem PP souvisejí.

Parametr - Proměnná, jejíž hodnota zůstává v daném systému či procesu pevná, která se však může měnit vně systému či procesu.

Program - Úplná logicky skloubená posloupnost instrukcí a dat pro řešení dané úlohy.

Obecný program - Program určený k zpracování široké oblasti určitých úkolů uvnitř daného typu použití.

Generátor - Program k vytváření jiných programů určitého typu (např. třídících programů, výstupních programů a pod.)

-----

Odvážil bych se položit takovou první pracovní definici, co je to PP. PP je takový program jehož základní funkce, t.zn. vstup dat, transformace a výpočet lze upřesňovat a doplňovat na základě více či méně jednoduchého jazyka parametrických příkazů, a to tak, že tyto parametrické příkazy se zpracovávají před vlastním zpracováním dat.

-----

Myslím, že "se zpracovávají před" v té definici je samozřejmé. Nemohou se práce zpracovávat za procesy, které mají ovlivňovat. Snad by tam mohlo být, že parametry netvoří samy o sobě program, že tedy v sobě mají jakési logické mezery, které doplní

až ten parametrický program.

PP je účelový a funkční a to ho odlišuje od jiných typů programů, např. kompilátorů. Na druhé straně má určitou obecnost, která ho zase odlišuje od programů řízených daty, protože každý program má v sobě určité větve a podle dat se větví jeho průchod. Dále, při tvorbě PP nevidím žádný rozdíl mezi interpretátorem a generátorem. Ten rozdíl je až při zpracování.

Interpretační program v první fázi vytváří jakýsi vnitřní kód těch parametrů. Jenom vyjimečně se parametry zpracovávají ve formě, v jaké jsou napsány. Při generování jakýmsi způsobem doplňujeme algoritmus, kdežto při interpretaci máme předem naprogramovány všechny možné větve a vnitřní kód parametrů řídí přepínače mezi jednotlivými větvemi existujícího programu. V tom je z hlediska časového dopadu hlavní rozdíl. Neustálé vyhodnocování přepínačů více nebo méně zpomaluje ten režim proti generačnímu.

Generátorem si mohu obvykle problém řešit více na míru. Místo parametrické konfekce interpretátora dostávám apartní obleček šitý na míru.

Ovšem pokud generujeme samostatný modul, zase tu přibude jeden krok. Nejdříve musíme generovat a v dalším kroku teprve můžeme vygenerovaný modul použít. Při jednorázovém použití je to zbytečné. A když tam předem připravíte nějaké funkce, tak to bude zase konfekce, která nemusí vždycky sedět.

Má to vůbec pro naši diskusi smysl, abychom přesně definovali, co je PP? Vždycky si v definici dokážeme najít nějaké hnidy. Instinktivně to chápeme asi tak, že programy pro jejichž činnost mají parametry zásadní význam, jsou parametrické. Ty programy, do nichž sice vstupují parametry, ale jejichž vliv není podstatný pro činnost programu, nebudeme mezi PP počítat. Myslím, že i když to není přesné, pro naše účely to stačí a my bychom se mohli věno-

vat důležitějším věcem. Parametrické programy se psaly a budou psát a my bychom měli přispět k tomu, aby se psalo více programů lepších a méně špatných. Při psaní parametrických programů existuje celá řada konkrétních problémů a o těch bychom měli mluvit.

## 2. Poznámky k návrhu a konstrukci parametrických programů.

Parametrický program by neměl být předváděcím kouskem programátora. Tím, že udělal krok v obecnosti, se chce předvést. V tvorbě PP by měl být systém vycházející z daného výpočetního střediska a z toho by měla vyjít filozofie jeho tvorby. Hlavně z té strany k uživateli. PP by se měl dělat pro určité prostředí.

Při tvorbě parametrických programů je nutno uplatňovat systémový přístup. Je třeba se podívat na systém zpracování dat, dekomponovat ho na jednotlivé části, rozdělit na jednotlivé funkce a ty postupně parametrizovat.

Chystám-li se napsat PP, rozhodují se, jaké vlastnosti by měl mít. Kdybych chtěl realizovat všechny funkce, které někoho napadly, nikdy bych tu věc nedodělal. Musím tedy přijmout nějaká omezení. Jak se rozhodovat o těch omezeních? Přijal jsem průběhem doby jisté zásady:

- a. raději méně funkcí, ale ty přijaté ať fungují pokud možno ve všech vzájemných kombinacích
- b. raději zavrhnou některé další funkce, ale pokud možno se vyhýbám kvantitativním omezením typu: "maximální hloubka závorekovaní je 5", "počet sloupců až 15" a pod.

Obě tato hlediska vycházejí ze snahy o větší celistvost a jednoduchost z hlediska uživatele.

Pokud člověk musí něco omezovat, měl by ta omezení posunout do nereálných hodnot. K tomu se dá dospět použitím vhodné programátorské techniky, např. místo pevného pole užít dynamický seznam. Když ale omezení jsou, měla by být popsána a měla by mít svou logiku. Logiku z hlediska uživatele a ne logiku plynoucí z technických potíží toho, kdo program dělal.

Když se parametrický systém vytváří, je dobré nechat uživateli v různých bodech možnost vstupu do systému. V těch bodech si může

uživatel připojit vlastní napsanou proceduru. Takový systém je potom životaschopný.

PP je nutno vidět v kontextu s okolím. Důležitou složkou je jazyk, tedy parametry řídící program. A tedy je třeba mít nejvíc na zřeteli, komu je program určen. Je-li pro programátory, analytiky, či dokonce pro náhodné uživatele. V každém případě by měl být jazyk PP samovysvětlující. To ale znamená něco jiného pro analytika a pro programátora.

Řeklo by se, že programátor bude vždy schopen rozumět parametrům, kterým rozumí laik, neprogramátor. Ale není to pravda. Programátor v tom chce mít vždy svoji logiku, aby si mohl představit, jak to funguje. Logika, kterou shledává laik jako samozřejmou, připadá programátorovi jako obkurní, protože odporuje všem jeho zkušenostem, jak mají vypadat parametry, které fungují.

Ve způsobu zápisu parametrů existují extrémy. Jsou PP, kde vstupní parametry tvoří řada čísel oddělených čárkami. Každé číslo, každá čárka něco znamená. Běda, když vám z toho něco vypadne. Takové parametry je umění napsat, ale je to úsporné. Druhým extrémem je psaní dlouhých klíčových slov s velmi přísným anglickým pravopisem a s důrazem na každé písmenko. Člověk toho napíše hodně, ale efekt neodpovídá vynaložené námaze. Zápis parametrů tak, aby byl stručný a zároveň srozumitelný uživateli, to už je panečku umění. A důležitost tohoto umění stoupne při používání interaktivních metod. Při dialogu bude srozumitelnost základní věcí. Ta bude ve značné míře určovat, zda se bude něco užívat nebo ne.

### 3. Užívání parametrických programů.

Takové parametrické programy dovedou pěkně zaplácat počítač.

Je takové někdy dost hanlivé pro programátora, že používá parametrické programy.

U nás máme tři nebo čtyři programy, které spotřebovávají asi 90 % všeho rutinního času počítače a to jsou shodou okolností

právě parametrické programy, a to proto, že na jejich základě je prostě postavena většina rutin. Jen co není rozumně řešitelné v rámci těchto rutin, na to se píší speciální programy.

Programátorské osobnosti odmítají parametrické programy, protože ony to napíší líp.

U nás je to uděláno tak, že programátoři musí. Vedoucí si to kontroluje. Oni nejdříve reptají a pak si zvyknou. Nejhorší je ten začátek, když musí nastudovat manuál a vzdát se svých zvyků nebo i zlozvyků. Když se to naučí a má na vybranou, pak určitě použije parametrický systém.

Pro nás banky dat nepřipadají v úvahu, ale náš parametrický systém nám umožnil dosáhnout jakési zvýšené integrity dat, která by samostatným programováním byla nedosažitelná.

Na jednom školení IBM se zdůrazňoval diferencovaný přístup k programům. Uvádělo se tam, že rozбором satížení počítače zjistíme, že ze 100 % programů existuje asi jen 2,5 % programů, které spotřebovávají 50 % strojového času. Na druhé straně je velká skupina programů, říkají, že jich je 50 % a ty spotřebovávají jen 2 % strojového času. A na základě těchto údajů zdůrazňovali diferencovaný přístup. U té malé skupinky programů s velkou spotřebou strojového času je hlavní hledisko hospodárnost, to znamená, co nejvíce optimalizovat a takové programy dělat třeba i v assembleru. U té druhé skupiny, která se na spotřebě strojového času podílí jen nepatrně, tam je místo pro parametrické programy. Tam hospodařit spíše s prací programátora než se strojovým časem. To bylo jedno hledisko, které uváděli. A druhé neméně důležité hledisko je životnost programů. Je celá řada programů, které jsou vyloženě jednorázového charakteru. Naproti tomu je určitá, ale mnohem menší skupina programů, které mají přímo kořičí život. Ty přežívají 5 i 10 let a často přecházejí při inovaci výpočetní techniky v téměř nezměněné podobě na další instalaci. A zase se má k tvorbě programů přistupovat diferencovaně. Programy s dlouhým životem musí být dělány mnohem pečlivěji, dobře dokumentované a u těch jednorázových je typické použití PP.

Pokud se týká efektivnosti, tam jsou rozdíly. Ale každý parametrický program jde udělat efektivně a nezáleží, je-li to interpretátor nebo generátor. Záleží na použitých jazycích a technikách.

Nemám pocit, že by používáním parametrických programů utrpěla efektivita zpracování. PP bývá dílem zkušeného programátora, na jeho zhotovení měl dost času, netlačil ho termín. Může se stát, že použitím nějakého špičkového jednoúčelového programu dosáhneme lepší efektivity, ale ve srovnání s vytvořením běžného programátora snese parametrický program určité srovnání.

Řada parametrických programů se tváří univerzálně, hovoříme-li o nich obecně, ale v konkrétní situaci jsou toporné.

Já jako uživatel se dívám na parametrické programy z trochu jiného pohledu. Chci, aby to bylo co nejjednodušší a udělalo mi to práci, kterou žádám a abych se v tom vyznal. Tedy by mohly hodně udělat dobré popisy, odpovídající funkce a spolehlivost. Program by měl fungovat pořád stejně. Mne jako uživatele nezajímá, co se u vás děje. Změňte si program, změňte si operační systém, ale jednou napsaný parametrický program musí stále fungovat. Pokud se týká popisu, ten by měl obsahovat:

1. jednoduchý popis včetně omezujících podmínek
2. příklady dobrého i špatného použití
3. diagnostiku a jak hledat chybu
4. návody na optimalizaci z hlediska zjednodušení a zrychlení instrukcí; časy, které jednotlivé výroky potřebují.

Dobrý PP, i když je složitý, by měl mít tu vlastnost, že začátečník jej může použít nějak primitivně. Jak rostou jeho nároky a zkušenosti, může pak jít do hloubky.

Výhodou parametrických programů je, že je jsou schopni přímo používat systémoví analytici bez zprostředkování programátorů. Zde udělaly parametrické programy spolu s rozhodovacími tabulkami mnoho.

Moje zkušenosti z vědecko-technických výpočtů (VTV) jsou takové, že pokud uživatel nemá možnost sám používat nějaké parametrické programy, má neúměrné nároky na práci programátorů.

Parametrické programy u VTV jsou většinou směřovány na uživatele. To souvisí s tím, že u VTV bývá organizace chodů zpracování jiná než u hromadných dat. U hromadných dat by měl být autorem návrhu PP analytik, a to se zatím neděje. Ty koncepce, pokud jsou, bývají ryze slovní, mnohdy zmatečné, plete se popis algoritmu s popisem dat. Tyhle věci většinou nejde zachraňovat, když je středisko zavedené. Nejlepším místem pro radikální změny je instalace nového počítače. Jestliže si to neuvědomíme, pak je zle, prosazují se staré přístupy a hlediska.

Je jasné, že pro různé uživatele nevyhovuje stejně dobře tentýž jazyk. Ideální řešení by mohlo být vytvořit jazykovou nadstavbu, která by byla schopna každému uživateli poskytnout na míru šitý jazyk.

Existují různé profese s podobnými problémy, ale různými jazyky, takže namají šanci se spolu domluvit. Tady by ta možnost měnit jazyk byla velmi prospěšná. A ještě k VTV. Tam mají autoři programů tendenci zanedbávat vstup a výstup. Prostě ta data do počítače nějak narvou a soustředí se na své transformace. Když jim pro vstup a výstup poskytnem parametrický program, zlepší se štábní kultura, ušetří se práce a zlepší aplikovatelnost. Chtěl bych ještě poukázat na přenesenost PP. Přecházíme-li na jiný počítač, nemusíme převádět řadu speciálních programů. Ten počet je nižší.

Řada z nás, jak tu sedíme, jsme autoři parametrických programů pro svoje výpočetní středisko a okolí. Myslím, že je správné, když se organizační hlediska, zvyklosti a potřeby shrnou do algoritmu a naprogramuje se to do PP. Myslím, že nemusíme usilovat dělat programy pro všechna střediska. Na to je RPG, CULPRIT a podobně. S předáváním programů cizím organizacím je stejně tolik potíží, že to, co se povedlo předat, se dá spočítat na prstech.

#### 4. Předávání parametrických programů.

Předávání programů je složitá transakce mezi čtyřmi partnery. Jsou zde dvě strany - předávající a přijímající a u obou je mnoho ještě rozlišit dvě složky. Jednu tvoří programátoři a druhou hospodářské vedení. Podívejme se na otázku předávání programů z hlediska jednotlivých partnerů.

Programátoři předávající strany by měli mít radost z toho, že vytvořili úspěšné dílo o které je zájem a může být použito i v jiných organizacích. Obecně a tudíž aložitější programy však zpravidla píší již zkušenější, světa znalí programátoři, kteří jsou si plně vědomi i negativních důsledků. Znají heslo: "s každým novým uživatelem, nové potíže" a vědí, že se na ně bude neustále někdo obracet a otravovat je. Pokud je program náhodou řešením tematického úkolu nebo zlepšovacím návrhem, je zde sice určitý stimul naděje na hmotnou odměnu, ale ani ten nebývá příliš silný. V nejhorším případě 200 Kčs rozděleno na čtyři autory nezní nijak lákavě.

Hospodářské vedení předávající organizace nemá na předávání žádný zájem. Ze zákona má sice právo fakturovat poměrnou část nákladů vynaložených na vývoj programu, ale to mnoho neznamená. Při vysokých cenách počítačů, velkých odpisech a fakturacích za strojový čas jsou částky získané prodejem programů řádově nižší a tudíž nezajímavé. Navíc si vedoucí pracovníci uvědomují, že předávání zdržuje jejich nejlepší programátory a odvádí je od užitečné práce pro vlastní organizaci.

Programátorům přijímající strany by měl předávaný program ušetřit práci a dát jim okamžitě do ruky nástroj, který by jinak museli vyvíjet měsíce. Je to ale cizí těleso ve vlastní software a jako k takovému k němu přistupují s nedůvěrou. Ze zkušenosti poznali, že to budou oni, na které se budou vlastní lidé obracet s dotazy, a co je horší, s žádostmi na případné doplnění nebo změny. Nelze podceňovat i určitý pocit obavy z obvinění, proč přebírají cizí programy a proč raději neudělají něco pořádného sami. Kladným podnětem by mohla být až 60 % odměna za realizaci v případě, že se jedná o zlepšovacím návrh. Těto možnosti, dané zákonem, se však využívá velice sřídka.

Největší zájem na výměně programů by mělo mít hospodářské vedení přijímající organizace, protože ta má z celé akce největší



prospěch. Bohužel hospodářské vedení si přínosy většinou nedovede představit, ale velmi dobře ví, že i když částka za koupi programu je směšně nízká, tak nebyla včas zařazena do plánu a budou s ní potíže. Výplata odměny či podílu za realizaci, pokud jde o zlepšovací návrh, je ještě obtížnější, protože ta jde z jiných, a to ostře sledovaných fondů.

Shrneme-li si vše co jsme dosud uvedli, dojdeme k závěru, že je až s podivem, že se nějaké programy vůbec předávají. Děje se tak jen díky osobním kontaktům jak mezi programátory tak i mezi vedoucími pracovníky na různých seminářích a školeních. Přesto si myslím, že výměna programů je věc velice užitečná a zasloužila by si větší pozornost i z vyšších míst.

-----

Jakmile se někdo dostane na čelo nebo chvost, pak přenositelnost jeho produktů je vždycky malá. Ten na chvostu je na tom relativně líp, protože má z čeho vybírat.

-----

Často autoři přehnaně inserují jednoduchost FP. Přitom vždy jsou nějaké potíže a to uživatele znechucuje.

-----

Spíše lze získat produkt, než garanci. Garantovat vám nebude nikdo nic.

-----

Pokud nějaký systém více umí, musí o tom být elaborát na 50 stranách a když to někomu pošlete, nikdo to nebude číst a lámat si s tím hlavu.

-----

Je-li PP určen programátorům, pak hlavní brzdou rozšíření je ješitnost programátorů. Přestě si to napíší sám a lepší, protože to bude moje. A má-li být uživatelem neprogramátor, je situace přímo tragická, protože ten uživatel, který by s tím měl přijít do styku, se vůbec nedozví, že něco takového existuje. Náš systém parametrických programů se užívá tam, kde se k němu dostaly jiné kruhy než programátoři a byly dost silné, aby si to prosadily. Málokdy se stane, že zvítězí druhý aspekt programátorské povahy a to je lenost.