

Ing. Petr Henzl

RNDr. Jiří Hřebíček

Ústav fyzikální metalurgie ČSAV Brno

PROGRAMOVÁNÍ VĚDECKOTECHNICKÝCH VÝPOČTŮ

Příspěvek je věnován některým nejdůležitějším oblastem činnosti výpočetního střediska, zabývajícího se vědeckotechnickými výpočty (dále VTV). Na příkladu výpočtového oddělení ÚFM ČSAV je ukázán způsob evidence publikovaných algoritmů, metodika týmové práce a programovací konvence při vytváření velkých programových systémů ve Fortranu a způsob jejich dokumentace.

Výpočetní střediska, která se zabývají především VTV, řeší zpravidla úlohy, jež mají obecný matematický charakter nebo danou úlohu lze převést na řešení známého matematického problému, pro který lze někdy použít publikovaný algoritmus. Pro úsporu práce jak programátorů, tak i výzkumných pracovníků je účelné častěji se vyskytující odborné dílčí úlohy formulovat jako samostatné algoritmy, publikovat, třídit podle oborů a tak je dávat k dispozici širší veřejnosti. Stejně důležité je shromažďování publikovaných algoritmů z oblasti VTV. Ve výpočtovém oddělení ÚFM ČSAV se důsledným evidováním a shromažďováním publikovaných algoritmů zabýváme od roku 1969. Účinnou pomocí v této činnosti bylo využití automatizovaného rešeršního, dokumentačního a informačního systému ARDIS. Vytvořený doku-

mentální fond byl nepřetržitě doplňován záznamy o algoritmech, publikovaných především v Comm. & f. M., Aplikace matematiky, Výběr informací z organ. a výp. techniky, Knihovna algoritmů ze sympozií "Algoritmy ve výpočtové technice", Computer Physics Communications, IBM System 360 Scientific Subroutine Package, knihovna MAG (National Algorithm Group). Dále jsou v něm uvedeny informace o algoritmech, standardně využívaných ve výpočtovém oddělení ÚFM ČSAV a některých dalších použitelných algoritmech. Evidence a shromažďování algoritmů se osvědčuje a v řadě případů bylo možné převzít a používat efektivní algoritmy, které uspořily jak programátorskou kapacitu, tak strojový čas.

Často se stává, že buď algoritmus nebo publikovaný program není k dispozici. Pak je nutné matematický problém algoritmizovat nebo znovu naprogramovat. Přitom se vyskytuje celá řada příbuzných problémů, které lze zahrnout do řešení obecnějšího problému. V takových případech bývá účelné a efektivní vytvořit univerzálnější matematický software, který v rámci řešení obecného problému umožňuje řešit jednodušší problémy.

Při vytváření takového matematického (nebo aplikovaného) software je důležité, aby programový systém byl portabilní, flexibilní, spolehlivý, kladl minimální nároky na paměť, byl dostatečně rychlý a hlavně aby jeho používání bylo co nejjednodušší. Je pochopitelné, že takovýto systém bude vytvářet programátorský tým, případně budou spolupracovat programátorské týmy z více výpočetních středisek. Příkladem takovýchto univerzálních programových systémů jsou systémy OPTIPACK [1], resp. PROKOP [2], vytvářené v rámci komplexních racionalizačních brigád ve spolupráci ÚFM ČSAV, SVT ČSAV, resp. ÚFM ČSAV, LPS Brno, ČKD Blansko a ŽDAS.

Při vytváření těchto systémů se osvědčila týmová práce. Celá koncepce jak systémů OPTIPACK, tak PROKOP vznikla za spolupráce vždy více (6 - 10) pracovníků. Práce týmu je organizována tak, že vedoucí týmu a jeho dva zástupci znají dokonale celý systém a navrhuji a vytvářejí jej vzájemně jak z hlediska algoritmizace matematického problému, tak z hlediska operačního systému počítače. Ostatní programátoři v rámci dané koncepce samostatně řeší dílčí úkoly a programují je. Vzhledem k neustálému vývoji systému je vždy určitá etapa "konzervována" (obvykle jednou ročně), jsou provedeny popisy všech uživatelských částí systému, revize opravených algoritmů a do užívání je dána další verze systému. Při řízení týmové práce je životně nutně zavést jednotnou metodiku programování pro všechny členy týmu a jednotné systémové prostředky pro komunikaci s počítačem. Uvedeme zde stručně některé naše zásady tvorby velkých programových systémů ve Fortranu, které se nám osvědčily.

Univerzální programový systém (dále jen SYSTEM) je navržen na těchto hlavních zásadách : Co největší portabilita na počítačích vybavených fortranským kompilátorem, co nejjednodušší používání ze strany uživatele a co největší jednotnost systémových konvencí, týkajících se formy a struktury programů SYSTEMu. Proto je SYSTEM vytvářen jako knihovna podprogramů, která se neustále doplňuje. Spojováním podprogramů se pak sestavují programy pro řešení různých typů dílčích úloh daného obecného problému. Přenos informací mezi podprogramy se děje přes parametry a systémové COMMON-bloky. Velké datové soubory se přenášejí přes vnější paměťová média (obvykle magnetické disky) s jednotným způsobem zápisu. Programy se píšou ve standardním Fortranu (podle normy USASI X3.9-1966).

V odůvodněných případech je možno se od normy odchýlit, zejména v zájmu vytvoření rychlejšího programu; odchylky je nutno uvést v úvodním komentáři a bezprostředně před odchylným příkazem. Některé programy jsou závislé na typu počítače či operačním systému, tato závislost se rovněž vyznačí v úvodních komentářích. Programy jsou psány podle zásad strukturovaného programování [3]. Ačkoliv Fortran neobsahuje řídicí struktury vhodné pro psaní programů podle zásad strukturovaného programování, tyto struktury se v SYSTEMu přijatelným způsobem simulují. Jedná se o struktury typu if-then, if-then-else, do-while, do-until a case, případně jiné. Dále je v SYSTEMu předepsaná formální úprava programů, tj. např. rozdělování příkazů, označení pokračovacích řádků, číslování návěští, užívání celočíselných proměnných u čísel zařízení vstupu a výstupu atd. Názvy programů provádějících podobné činnosti (vstupy, výstupy, atp.) začínají předepsanými skupinami znaků, které stanoví vedoucí týmu. Pro zvýšení srozumitelnosti programů se programy co nejlépe komentují. V SYSTEMu jsou používány následující typy komentářů: Úvodní, jímž začíná každá programová jednotka a který má přísně formalizovaný tvar, definující název podprogramu, operační systém, číslo verze a modifikace, zkratku autora a datum napsání dané verze a modifikace. Další řádky úvodního komentáře stručně popisují činnost daného programu. Popis parametrů, kterým vysvětluje význam formálních parametrů a u funkčních podprogramů též funkční hodnoty. Odchylky od normy, které upozorní na příkazy nezapsané v čistém Fortranu bezprostředně před jejím výskytem a vysvětlení, jež blíže vysvětlují činnost následující části programů nebo simulovanou řídicí strukturu.

Neméně důležitá jako vlastní programování je i dostatečná

dokumentace programů SYSTEMu. Pojem dostatečnosti dokumentace je relativní a závisí na tom, komu je dokumentace určena. Jiný obsah i rozsah dokumentace je třeba volit pro uživatele příslušného programu a jiný pro autory dělající opravy a změny programu.

Při vytváření programové dokumentace SYSTEMu se ve výpočtovém oddělení ÚFM ČSAV osvědčilo tvořit pouze uživatelskou dokumentaci, kterou mají k dispozici všichni uživatelé tohoto systému. Autorská dokumentace jako doplněk uživatelské je uložena přímo ve zdrojovém textu programů SYSTEMu. Tak má vždy ten, kdo ji potřebuje, požadovanou dokumentaci k dispozici.

Při vytvoření nové verze SYSTEMu, která vznikne jako výsledek provedení oprav a "konzervování" tohoto stavu, je nutno změnit také uživatelskou dokumentaci a to jak na autorském pracovišti, tak u externích organizací, kterým byl SYSTEM předán do užívání na jejich počítačích. Těmto organizacím jsou rovněž dodány opravené podprogramy SYSTEMu. Aktualizace uživatelských manuálů se provádí centrálně ve výpočtovém oddělení ÚPL ČSAV všem uživatelům, kteří jej k aktualizaci dodají. Odtud je vidět, že způsob vedení programové dokumentace je v ÚFM ČSAV jednoduchý a přehledný, což jsou základní předpoklady pro její úspěšné používání.

Literatura :

- [1] Kučera J., Hřebíček J., Kopeček I., Lukšan L.: OPTIPACK, uživatelský popis verze 2.0., ÚFM ČSAV Brno 1980.
- [2] Hřebíček J., Holuša L., Kopeček I., Kučera J., Čermák L.: PROKOP verze 0. Základní programové vybavení. ÚFM ČSAV Brno 1980.
- [3] Dahl O., Dijkstra E.W., Hoare C.A.R.: Structure programming, Academic Press, 1972.