

Redakční rada

VÝBĚR Z DISKUSE NA SEMINÁŘI 1977

Předkládáme Vám výběr některých myšlenek z diskuse na loňském semináři. Myslíme si, že výběr je v této podobě jakousi kolektivní mozkovou vyčišťující se k některým problémům, a že si tedy snad počtete, což Vám přejeme

Redakční rada.

* * *

Tak se rok z roku nešel a jestli jsme si loni mohli říci s klanem Jára Cimrmanem: "Tady ne, přátelé!", můžeme se letos zeptat znovu a připojit otázku: Tady tedy v těch nových, převratných a takle teoreticky opřevzaných programovacích technikách? Domníváme se, že řada referátů tento problém obhájela.

* * *

Modul musí být konstruován tak, aby byl blávdorný. Mám-li se spolehnout na tu jednoduchou sílu - stavební modul a postavím-li s ním určitý program, pak příslušný program můžu hledat jenom v obytných spojkách. Stavět takovou konstrukci však ohne důslednou analýzu. Má-li být analýza důkladná a důsledná, musí být exaktně dokumentována a koncepčně dobře vedená. Hlavní kvalita řešení programu leží tedy v oblasti analýzy. Myslím si, že doba došla k tomu, aby se specializoval jeden člověk, takový stavitel modulů, který by čerpal z analýzy nebo spolupracoval s ana-

lyzou, ale neprovoval by na ní. Spolupracoval by úzce s analytikem, navrhoval by programovací práce a veděl by jednotlivé moduly programátorům.

* * *

Při největší fluktuaci lidí a rozvoji techniky, rozvoji operačních systémů, ne k počítači dostávají lidé, kteří mají často pouze vedléjší ÚSV nebo ÚSO, kteří jsou ochotni chodit na směny. Operátoři se naučí mazat pásky, psát GAMBEL nebo DELETTÉ, ale služby a servisnosti, které vidí programátor, ty nevidí. Myslím si, že na světě by měl být aspoň jeden člověk, který aktivně ovládá programování, své systémy, kterého by programátoři mohli oslovnout "pane Operátore".

* * *

Nejsem zastáncem stávajícího ani usavřené provozu. Jsem v tomto směru spíše radikální. Některé situace, které se přihodily mi před čtrnácti dny, když jsem přišel k počítači a napečítal jsem tam osmáct lidí a já jsem byl devatenáctý, je extrémní.

* * *

Myslím, že bychom se neměli bát nějaké převíry chování na síti. Programátor, který připravuje rutinní provoz a musí jej tedy znát, se společně několikrát podívá a potom přestane na síti chodit. Postupně začne velice pečlivě svažovat, kdy se mu vyplatí strávit čas chůzí na síti a kdy ne.

* * *

Usavřený provoz je špatný, protože tam pracuje operátor. Tam, kde pracuje programátor, také operátor nechodí a nestravuje ho.

* * *

Pokud se programátor potřebuje k počítači dostat a má k tomu rozumný důvod, měla by se mu ta možnost nachat. Když se kolem počítače nachází 20 programátorů a každý k němu potřebuje jít dvakrát ročně, tak těch 40 čtvrt hodin je pro provoz nezbytná.

* * *

Jsem otřesen, když si uvědomím, že když tedy někdo začne
mluvit o zavěšení a stavěním provazu, že v plánu tolik lidí
o tom vzniknou jenom dvě představy. Představa nálu, kam ode-
jí třicet programátorů hloučkat se operátory, brát jim psof
stroj z ruky a občas si operovat sami. Myslím si, že to je
překonaná představa stavěním provazu. Třetí generace nám
otvírá dveře k možnosti dálkového zpracování a dálkového
ovládání počítače. Je třeba někde vybevit některou místnost
displejem a říci novým programátorům: "Tady máte, kluci,
skuste nám to skazit. Kuste si tedy dělat, se choďte."
A kluci si budou občas hrát, ladit si své programy, mohou
tam kompilovat a někde občas občas operátory. Tak jsem si
představoval stavěný systém a třetí generace.

* * *

Článek 1 přibližně strukturovaného programování se
shodují v tom, že programy by měly být jednoduché, přehledné
a čitelné. Avšak rozcházejí se v tom, jakou metodu pro de-
finici těchto cílů zvolit. Domnívám se, že žádná vlastně ne-
existuje jednoduchá, přehledná a čitelná definice jednodu-
chosti, přehlednosti a čitelnosti programu a navíc neexistu-
je ani definice těchto pojemů a je velmi obtížné nějak tyto
vlastnosti kvantifikovat.

* * *

Ať už se programovat bez GO TO. Někdo na tom je. Přibí-
je se tím rozum a spolek. že je dobré, když si člověk nějaký
ten program bez GO TO skusí. Kustil jsem si naproti dva
programy asi tak v rozsahu kolem třiceti na jeden. Celkem
člověk získá nový pohled na ty FOR a END. Ale pravda, tvrdí
přikázat programovat bez GO TO, to ne.

* * *

Každý z nás by pravděpodobně dohlávil naproti program,
kde GO TO vůbec nebývá a přitom to bude program naprosto
necromentální, nepřehledný a zcela špatný. Ale nakázat GO TO
je, jako když ministrům kultury někde literární
poukázat přehledně.

* * *

Nikdo se netvrdí, že GO TO je třeba odstranit úplně, prostě likvidovat. Můj dojem, který žna dlejší v odborné literatuře, že dnes už toto stanovisko nasertová nikdo.

* * *

Strukturování programování jako takové je velká věc, v níž nejde vůbec o to, jestli se má programovat s GO TO nebo bez GO TO, ale o to přiklonit se kladně krovni každému modulu nějakou jednotnostem ústředí, podle které si k danému vstupním parametrům v modulu vyrobí příslušné vstupní parametry.

* * *

Má vůbec ty články GO TO nebo bez GO TO připomínají ty různé spory, jestli je PL/I lepší než Cobol nebo Fortran. Ani tedy to není podstatná věc a je zbytečné se tím bavit.

* * *

GO TO by se měla používat nejvíce v rámci jednoho programového bloku, tedy nejvíce v rámci jedné stránky. Poprvé jak může být program strukturalizován.

* * *

Cháti bych se souhlasit s ještě neobdobnějším prostředkem pro nepřehlednější program než je občas GO TO. Jde o vředy, programátory při zapínání počítače. Tyto vředy umožňují určitě se cíle programu, zejména jsou-li plány na více místech programu různé hodnotami.

* * *

Myslím, že příliš důvěra na autority, že to říká nějaký slavný programátor. Vždyť by jeho většinou skočení programátoři. Proč tedy číst na nějakou autoritu, která vás něco přikazuje, když není proč.

* * *

Cháti se, že politické hlasy pro nívtala, pro vedoucí pracovníky. Měl jsem příležitost vidět různé směry těchto lidí. Je to silná silovitost. Otčená právě pro ty nívtala. Tam se právě pro lidi, kteří mají připraven nějaký svůj referát, se kterým chtějí právě tuto říkat.

Jeden je specialista na Booleovu algebru i přednese studákům
znalákům této algebru. Druhý je specialista na feritovou
paměť i namaluje jim ten prstýnek, takhle jde jeden drát,
takhle druhý a přes to jde ještě jeden takhle. A teď jim vy-
světluje hysteremní smyčku. Oni na to koukají a říkají:
"A to je ten počítač?". Nikdo jim neřekne, se vlastně od
něj mohou chtít, že je to prostě nějaká buňka, která umí to
a to. Navíc je tam ještě nějaká stravná tíra, že je uří sí-
klady Algolu.

* * *

Myslím si o programátorech jako takových, že hodně
programátorská chybí pořádkovítovnost. Přiznám si sami,
že jsme dost velmi nepořádníci. Pakoň ve středním máte
nějaké normy, tak se je snažíme obcházet a škodíme tak
sami sobě.

* * *

Vědu se vyvíjejí nějaké univerzální programy, generáto-
ry, databáze a podobně. A vědu se to dělá jinak. To je v
podstatě hromada slytečně vykonané práce a sde je nutno se
zamyšlet, má by ten vývoj našel nějak sboordinovat. Jestli
by aspoň měla vytyčit nějaká koncepce. Surovějšně nikdo to
nevtyčí takhle celostátně, ale jestli by našel vydávat ně-
jaký časopis nebo něco takového.

* * *

Proč se obecně nepředávají takové výhodné systémy, o
kterých sde například mluvili s Teely. Ony jsou tedy bariéry.
Mají tak jednodušné vnít program od někohe, není také jed-
nodušné někomu program předat. Souvisí s tím dokumentace, sou-
visí s tím opravy. V takových programech mohou být chyby.
Jak to potom řešit? Souvisí s tím taková určitá neochůť.
Jednak dároc, protože ten si říká: "Proč? Co já s toho budu
mít? Když to někomu předám, tak s toho budu mít jen potíže.
Oni mi budou telefonovat, že jim to nejde. Když to budou
jejich chyby, ale může to být i moje chyba." Souvisí s tím
ale i neochůť přijmout, protože přijmout chyby samozřejmě ne-
má rád a xino to jich má dost vlastních.

* * *

O čem bychom se měli bavit přibližně. Za prvé a hlavně bychom se měli bavit s většinou analytů - programováni a nějaká analytická podchyzení a nějaká šifrová dokumentační techniky. K této otázce bychom se měli sejít s analytiky, protože i analytici pasivně hledají nějakou cestu, jak jednoduše komunikovat s programátorem. Za druhé a zejména po tom bychom se měli bavit o technice programování, a to více asi na tři témata: řešení modulů, řešení univerzálních programů pro takové rychlé řešení triviálních problémů a řešení systémových problémů v rámci DOS a OS.

* * *

Otázka aplikovat novou metodu pro jedince je otázka náročná. Otázka aplikovat novou metodu na celý kolektiv programátorů je větší záležitost. Měli bychom se vyznat, jestliže zjistíme, že jsme nedostali nějakého výsledku do ústředí ani po přečtení nějakého článku.

* * *

Mám-li se otáznout, co vůbec lze na takové semináři vyřešit. Možná, že někteří jsou skeptičtější, protože zjistili, že žádný konkrétní problém nás nemohli. Může jít například. Důvody, jak se nás a nich objevilo, jsou například tyto:

- Některé problémy nelze kvantifikovat. Nelze měřit kvalitu programu.
- Jiné problémy představují různé úrovně zkomplikovanosti. Může se řešit s analytiky, s procesními prvky.
- Můžeme spolu komunikovat. To je v našich možnostech přehledat. Často jsou totiž lidé schopni se nějaké nové slovo, svléknout nový termín. Navíc by měli také přispívat k tomu, abychom se dohodli analyticky.

* * *