

Prof. dr. ANDREJ P. JERŠOV

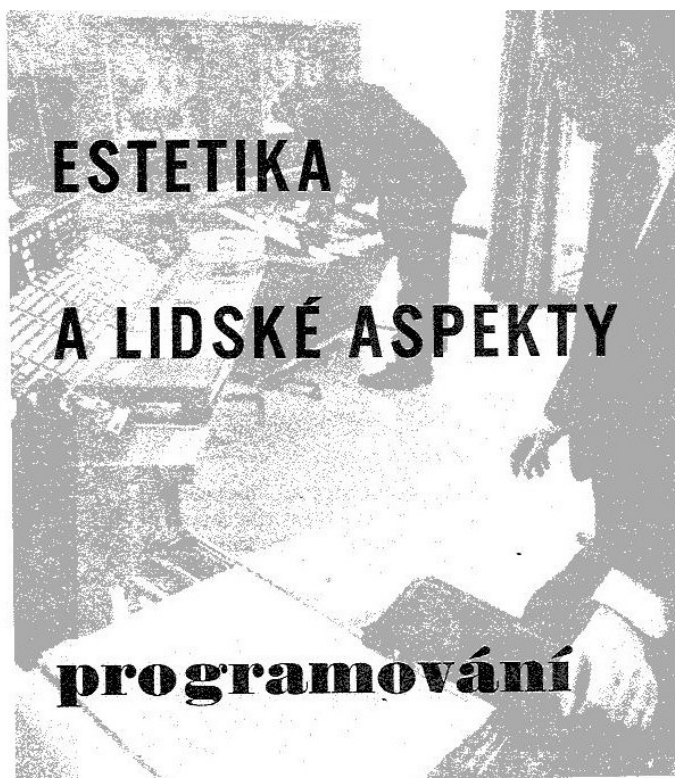
Autor článku, profesor sibiřské odbočky Akademie věd SSSR v Novosibirsku a známý odborník v oblasti moderní výpočetní techniky, přednášel loni na jaře na Spring Joint Computer Conference v USA. Tato konference se každoročně zabývá stěžejními otázkami vývoje výpočetní techniky a — dalo by se říci — její filozofii. Vystoupení prof. dr. Jeršova bylo označeno za jeden z vrcholů konference nejen v roce 1972, ale za celé uplynulé desetiletí. Závažnost autorových tezí je taková, že chceme čtenáře našeho časopisu s nimi seznámit v podrobném výtahu textu přednášky.

Dnešní doba je velmi obtížná pro programátory (pojem používám v nejširším slova smyslu a týká se samozřejmě také systémových analytiků). Romantická aureola této pro laika záhadné činnosti pomalu bledne. Společnosti pro poskytování služeb v oblasti systémového a aplikačního softwaru mizí jako sníh na jaře. Začíná se pochybovat, zda je oprávněné tvrzení programátorů, že představují zvláštní typ zaměstnanců. Projevuje se nová, mnohými z nich bolestně pocífovaná tendence: Právomoc nad kastou pracovníků dosud zvyklých na nemezenou tvůrčí volnost soustřeďuje se stále více do rukou administrativních a řídicích pracovníků, kteří se pokoušejí analytickou a programovací činnost plánovat, měřit, standardizovat a potlačovat její specifické vlastnosti.

Nepovažují tuto tendenci za zásadně špatnou. Jako administrátor byl jsem dlouhou dobu ve styku s programátory a také jsem hledal metodické zásady, jak systematicky organizovat jejich práci. Přitom jsem došel k závěru, že vztahy programování k okolnímu světu zdaleka ještě nevyzrály.

Problémy nejlépe ilustruje vývoj softwarových společností v posledním desetiletí. Malé firmy tohoto druhu se často ustavují během několika týdnů z iniciativy skupiny schopných programátorů a analytiků, kteří se sblížili ve velké organizaci, kde získali své základní zkušenosti. Pohnutky osamostatnění bývají rozličné, obvykle snaha o zvýšení příjmů kombinovaná nerealistickým přáním zbavit se přehnané kontroly. Hlavním podnětem bývá nějaká zvlášť zajímavá technická myšlenka, jež se stává základem softwaru nové firmy.

Zkušenosti však ukazují, že přežívají jen takové skupiny, kde původní motivy ustoupily přísné hospodárnosti a kde zavládla tuhá disciplína a hierar-



chické vedení. Jinými slovy to znamená, že všechno, co tyto odborníky přimělo opustit mateřskou organizaci, musilo se prosadit i v jejich vlastní softwarové společnosti, měla-li odolat tlaku konkurence. Ovšem, ne všichni ze zakládajících členů dočkají se konsolidace podniku, někteří odcházejí, poněvadž podnikání klade nemilosrdné požadavky. Má pozorování netýkají se jen americké skutečnosti. Také já mám podobné zkušenosti.

To dokazuje, že programování je v zásadě spojeno s velkými organizacemi. Nicméně, zařazovat jednoduše programátory do skupiny vysoce placených zaměstnanců je nebezpečnou tendencí. Má-li se jí zabránit, musí programátor najít ve své odborné práci určitý systém vnitřních hodnot, které mu pomáhají jak přizpůsobit se průmyslovým pracovním metodám, tak je i v případě potřeby překonat.

To mě přivádí k druhému tématu, totiž že takový systém vnitřních hodnot je objektivní součástí odborné činnosti na poli výpočetní techniky. Poněvadž však tyto hodnoty nejsou vždy vědomě vytvářeny a udržovány v myšlení programátorů, je třeba je blíže vysvětlit a obhájit. Jde o mnohostranný systém hodnot, z nichž nejmateriálnější je vědomí profesionálního postavení programátorů a analytiků. Nechci nyní zdůrazňovat materiální stránku, ale estetické a emocionální aspekty programování, nechci mluvit o odměně, kterou programátor dostává za své služby, ale o morálních činitelech ovlivňujících programátora, když je ponechán sám se svým programem a počítačem.

Lze ovšem namítnout, že nemateriální hlediska nejsou tak důležitá. Tvrdím, že jsou, a to nejen proto, že programování a podobné činnosti se stávají pevnou součástí lidské práce, ale hlavně proto,

že podle mého názoru je programování lidsky neobtěžnějším ze všech povolání:

1) Programátoři jsou první velkou skupinou lidí, jejichž práce je přivádí na ty hranice lidského vědění, pro něž jsou příznačné algoritmicky neřešitelné problémy a které se dotýkají tajných stránek lidského mozku a jeho činnosti (tj. rozhodovacích procesů).

2) Schopnost programátora proniknout do hloubky logického problému musí přesahovat oněch pět až šest stupňů, které podle psychologů charakterizují průměrného člověka. Musí být nejen tak hluboká, jak to problém vyžaduje, ale dosáhnout při nejmenším ještě o další dva až tři stupně hlouběji (aby dokázal najít účinnou variantu řešení; pozn. překl.).

3) Práce vyžaduje od programátora různorodé kvalifikace a schopnosti. Musí umět pracovat s logickými abstrakcemi jako kvalifikovaný matematik a současně mít talent takového Edisona konstruovat užitečné stroje z jednoduchých součástí. Musí spojit přesnost bankovního úředníka s bystrostí stopaře, k tomu přidat fantazii autora detektivních románů a střízlivou praktičnost obchodníka. A k dovršení všeho musí mít smysl pro kolektivní práci a pro zájmy svého zaměstnavatele.

Vím, že k překonání těchto potíží je třeba velkého emočního úsilí. Lze toho dosáhnout jen uvědomělou rozumovou kontrolou vlastní činnosti a kladným vnitřním postojem. Kdyby takový Artur Haley*) napsal román Výpočetní středisko, byl by jistě jedním z nejvíce strhujících bestsellerů.

Tím se dostáváme k další základní myšlence: Porozumění pro lidskou stránku programování a jeho estetika jsou nezbytné nejen proto, že jde o hnací síly programátorů, které by si oni sami měli uvědomit, ale jsou nezbytné hlavně pro ty, kdo počítačové odborníky řídí a zejména pro ty, kdo je vychovávají a školí. Uvádím několik organizačních dilemat, k jejichž řešení se můžeme propracovat jen tehdy, když přihlídneme ke všem motivačním a estetickým činitelům, které jsem uvedl:

- Je nutné, je vůbec možné organizovat vývoj softwaru a programů metodami podobnými montážní lince?
- Je možné nebo nutné oddělit projektování rozsáhlých úloh od jejich realizace?
- Při organizaci rozsáhlých projektů je obtížnější najít vedoucí nebo výkonné pracovníky a proč?
- Jak lze sladit na jedné straně elitní a na druhé straně hromadnou povahu prací na koncipování a programování systémů?
- Jaké osobní schopnosti máme hledat u programátora? Jsou tyto vlastnosti dost jasně vymezeny, abychom je mohli zařadit do administrativní rutiny?

*) Autor románu Letiště, který je v anglické jazykové oblasti velmi populární, popisuje s hlubokou znalostí letecký provoz. Kniha vyšla již slovensky, letos je připravováno i české vydání (pozn. překl.).

- Jak mají být programátoři školeni? Pěstováním širokého rozhledu (na školách univerzitního typu) nebo zdůrazněním profesionální obratnosti (v technických institutech)?

Všechny tyto otázky jsou pouze částí obecného problému, jak řídit vývoj softwaru a konkrétních programů.

Nejdříve poznámka týkající se přenášení metod hromadné výroby do oblastí programování a příbuzných činností. Montážní linka zvedla produktivitu na dříve nepředstavitelnou úroveň, současně však setřela individuální rysy práce. **Uplatníme-li v oblasti programování organizaci podle systému montážní linky, můžeme zničit intelektuální uspokojení z práce, které je hybnou pákou programátora.** Rozpor mezi jednotvárností takové činnosti a obtížností programátorova úkolu může způsobit neurózu. Představte si člověka, jenž musí řešit křížovky osm hodin denně, pět dní týdně, padesát týdnů v roce, a pochopíte, co to znamená pro programátora, je-li nucen specializovat se na určitý, omezený, stále se opakující modul (stavební kámen) programu. Proto nelze přijmout jen tak tezi, že produktivita programovacích prací se podstatně zvýší prohloubením dělbý práce, tj. rozložením úkolů programátora na poměrně jednoduché základní operace.

Co se týče vztahů mezi pracovníky řídicími projektem a výkonnými členy projektového týmu, ukazuje se, že schopnost vedoucího projektu dobře řídit technicky vysoce specializované spolupracovníky je často omezená. Vedoucí projektu zpravidla pověřuje odpovědností za realizaci jednotlivých dílčích úkolů raději mladého odborníka, který dokončil vysokou školu teprve před dvěma až třemi roky, než osobu s mnohem většími odbornými a životními zkušenostmi. V takovém případě je preferování poddajnosti mladého člověka vlastně pokusem vyhnout se argumentům zralejší a samostatnější osobnosti v polovině třicítky. Ukazuje to také, že málo pečujeme o odbornou úroveň a prestiž mladých výkonných členů týmu, že nezvyšujeme jejich úroveň souběžně s jejich věkem. Vždyť nejde jen o to, aby mladý odborník byl pohodlným partnerem nynějšího vedoucího projektu, ale aby byl užitečným i svým budoucím vedoucím a sobě samému.

Jakou orientaci vzdělání odborníků volit, zda se zaměřit spíše na získání širokého rozhledu nebo na znalosti přímo potřebné pro programování, je otázkou obtížnou a klíčovou. Odpověď rozhoduje o způsobu výběru i školení počítačových odborníků. Nejde jen o to určit nějakou optimální proporcii mezi absolventy univerzitního typu škol a softwarovými specialisty — techniky (ačkoli i to je důležité pro praktickou personální politiku), ale hlavní problém spočívá v tom, že programovací a analytická činnost vyžadují určité postoje, určitou morální přípravu. Počítačový odborník je nositelem a produktem druhé průmyslové revoluce a jako takový musí také do určité míry myslet revolučním způsobem. Programátoři představují vybranou skupinu pracovníků v dobrém slova smyslu a styl

a metody jejich práce ukazují v mnohém do budoucnosti.

Vraťme se k základní myšlence těchto úvah. Programování zahrnuje nové, bohaté a hluboce působící estetické prvky, jež jsou základem vnitřního vztahu programátora k jeho povolání, a které mu poskytují jak intelektuální, tak plně emocionální uspokojení. Tyto estetické pocity vycházejí z tvůrčí povahy programování, z potřeby překonávat obtíže a konečně ze společenského významu programování.

Není snadné vysvětlit základní estetiku jakékoli odborné činnosti. Vyplývá z určitého souboru subjektivních postojů spojujících příslušníky téhož povolání. Při této příležitosti citují ruské přísloví: „Rybář pozná rybaře na dálku.“

Profesionální estetika ovlivňuje a je ovlivňována etickými zákony daného povolání, předmětem jeho činnosti a jeho právním a společenským postavením. Pokusím se nyní vyložit vnitřní povahu programátorské estetiky.

Tvůrčí povahu programování není třeba zvlášť dokazovat. Domnívám se dokonce, že tvůrčí stránka je zde vyvinutější než u většiny jiných odborných povolání, podobá se například matematice a tvůrčí spisovatelské činnosti. U většiny jiných povolání využíváme pouze přírodních sil. I když pracujeme s fyzikálními a biologickými úkazy, doufáme, že inteligentně a hospodárným způsobem, nemusíme chápat jejich vnitřní zákony. Při analýze a programování musíme však jít až na kořen věci. Nemůžeme programovat to, čemu nerozumíme.

Jiným specifickým rysem programování je požadavek mimořádně vysoké přesnosti hotových vý-

sledků, což je ovšem příznačné pro mnohé druhy inženýrské činnosti. Programování však vyžaduje přesnost, která překonává přesnost většiny inženýrských prací. Mezi zdánlivě hotovým výrobkem a úplně odladěným i zavedeným programem bývá ohromný rozdíl. Nutnost dosáhnout stoprocentní realizace a vyhledávat všechny chyby způsobuje programátorovi mnoho vyčerpávajících potíží, ale současně mu poskytuje hluboké uspokojení v jeho práci.

Jedním z nejpříznačnějších a nejsilnějších aspektů programování a analýzy je pocit intelektuálního vítězství, že se podařilo dosáhnout dokonalé souhry stroje, programu, organizace. Poctivý programátor se chová ke svému počítači podobně jako dobrý závodní jezdec ke svému koni: Zná velmi dobře možnosti stroje, ale nedovolí si být intelektuálně lenivým a řešit problém bez ohledu na vynaložený strojový čas a jiné vypočetní zdroje, které může vázat jeho řešení. Takový čistě estetický vztah odborníka k jeho úkolu je účinnou zábranou proti hromadění neefektivních prvků v hotovém projektu. Podniky prodávající strojový čas sice proti podobným nedostatkům nemají námitek, ale pro uživatele znamenají okrádání a společnost ztrácí možnost plně využívat potenciál strojů.

Druhá skupina estetických problémů týká se společenských a veřejných funkcí programování a analýzy. Kdykoli se setkáváme se společenským úkazem velkého dosahu (a tím bezpochyby příchod počítače na dějinnou scénu je), hledáme historické analogie, které by nám dovolily analyzovat jeho význam a předpovědět jeho budoucí vývoj. Počítač je v tomto smyslu vlastně druhou průmyslovou revolucí charakterizovanou industrializací intelektuální práce. Podobnou analogii můžeme najít pro nová povolání počítačových odborníků. Postupné rozšiřování všech druhů softwaru lze přirovnat v mnoha směrech k jevu způsobenému vynálezem knihtisku. Hromadné tisknutí knih, z nichž každá vyjadřovala autorovy názory na zevní svět, prohloubilo a urychlilo společenský proces vzájemného porozumění.

Obdobně působí i programy, banky dat a projekty. Soustřeďují informace a operační modely světa a dovolují nejen ovlivňovat, ale i předvídat světový vývoj, takže dostáváme do rukou dosud neslýchanou moc nad přírodou.

Být dnes dobrým programátorem je určitou výsadou, jako bylo výsadou v šestnáctém století být gramotným a vzdělaným. Toto výsadní postavení způsobuje, že programátor očekává uznání a respekt společnosti. Jeho naděje se naneštěstí často nesplní. Je třeba vykonat ještě mnoho práce na obou stranách. Pokud se týká programátora, musí přijmout určité obecné etické zásady platné pro všechna odborná povolání, ale se speciálním zřetelem na povahu své činnosti.

Rozeznáváme tři základní typy postoje k práci: práci pro uspokojení z práce, práci pro peníze, práci pro dosažení cíle (tj. pro efektivní, plně použitelný výsledek analýzy a programování). Mezi

ZE ZAHRANIČNÍHO HUMORU



„Tak vám tedy nevaří, že sestava nevyšla?“

počítačovými odborníky platí hlavně první dva typy motivačních činitelů, ačkoliv ten třetí by měl mít absolutní přednost. Proto tvrdím, že programátor může dosáhnout plně harmonických vztahů se společností jen tehdy, jestliže se nerozlučnou součástí jeho vnitřních postojů stane absolutní věrnost konečnému cíli.

Mluvíme-li o společenských funkcích programování, nutno se zmínit o dosud nevyřešeném technickém problému, který brání plnému uplatnění společenského významu programování. Je to otázka hromadného a trvalého využívání vypracovaných projektů a programů. Názory se liší. Někteří říkají, že jen malý zlomek dnešních programů má trvalou hodnotu, jiní tvrdí, že operační systém OS/360 je přímo nesmrtelným programovým systémem [extrémní názor]. Ať tak či onak, je skutečností, že pocit odpovědnosti počítačových odborníků závisí v mnohém na tom, zda uvidí, že jejich projekty i programy mají dlouhodobější stabilní význam.

Nyní ještě několik poznámek k některým dílčím problémům. Důležité je ujasnit si, že jsou značné rozdíly mezi schopnostmi různých počítačových odborníků. S touto skutečností musíme počítat, abychom využili všech možností. Samozřejmě jsou v této oblasti i primadony, někdy nepřijemné, avšak ne vždy mají jen negativní vlastnosti. Setkal jsem se s několika lidmi tohoto typu, kteří přes svůj vyhraněný individualismus a svérázné chování dokázali právě v obtížných situacích pohnout projekčními pracemi.

Zajímavý je vztah mezi vypracováním koncepce a celého projektu na straně jedné a realizací programu na straně druhé. Řídící pracovníci odpovídající za větší projekty se snaží o formální definici hranice mezi projektováním a zaváděním. Vědomě nebo podvědomě doufají, že si zjednoduší řízení celého procesu tím, že zavádění projektu svěří jiné skupině pracovníků. Avšak povaha projekčních a programovacích prací není vůbec vhodná pro dělbu práce, která vytváří konflikty a přehlíží živé lidské činitele a jejich estetické potřeby. Tato estetika snižuje pracovní morálku při pasivním provádění převzatých myšlenek. Tvůrce projektu také instinktivně odmítá předat realizaci vlastní práce do cizích rukou. I když je to někdy nutné, dochází ke ztrátě intelektuálního uspokojení.

V minulých staletích se považovala znalost čtení a psaní za vzácný boží dar, výsadu vyvolených. V naší době podléháme pokušení vytvářet novou elitu, která se stává zprostředkovatelem mezi laickou většinou a tajuplným informačním modelem světa skrytým ve stroji. Myslím však, že vývoj směřuje k tomu učinit z umění programování všeobecné lidské vlastnictví.

Připomíná mi to poznatek z mého posledního pobytu v USA roku 1970, kdy jsem se seznámil s novými myšlenkami v problematice vyučování, které vyvinuli Marwin Minsky a Seymour Papert na Massachusetts Institute of Technology. Dokázali, že se lidé nejlépe učí, když si v hlavách vy-

tvářejí vývojové diagramy svého konání, vymezují hranice svých standardních rutin a informační vazby. Profesor Papert mě naučil za deset minut tolik, co bych sám nezvládl ani za několik hodin, pouze tím, že využil mých návyků myslet jako programátor. To ukazuje, že člověk může značně posílit svůj rozum, jestliže dokáže zvyknout si na plánování svých postupů a na vypracování příslušných obecných pravidel. Definovat pravidla a vyjadřovat je předem stanoveným způsobem, to znamená vlastně programovat. Pro mne je to dalším důkazem, že způsoby myšlení a metody práce pří- značně pro přípravu úloh pro počítač stanou se obecnými způsoby myšlení široké veřejnosti.

Trvalo skoro tisíciletí, než znalost čtení a psaní stala se všeobecnou. Další etapa, kterou jsem naznačil, nepotrvá tak dlouho, ačkoli jsme ještě daleko od tohoto cíle. Avšak každý programátor, i když jen listuje v manuálu, pracuje a je motivován myšlenkami, které ukazují do budoucnosti intelektuálního vývoje lidstva. Jsem přesvědčen, že kolegové a nadřízení počítačových odborníků musí chápat jejich profesionální pohnutky a výhledy.

Mé dosavadní poznámky se vztahovaly k běžným problémům lidského činitele při analýze a programování. Je však jedna snad ještě závažnější otázka týkající se budoucnosti. V dnešní době snažíme se získat prakticky neomezený počet mladých lidí pro odbornou činnost s výpočetní technikou a slibujeme jim modré z nebe. Ale lidské generace se mění mnohem pomaleji než generace počítačů. Musíme se naučit, jak zabezpečit, aby padesátiletí odborníci [za třicet let jich bude na světě milión] byli stejně užiteční jako mladí specialisté. Musíme si přiznat, že dnes nemáme ani zdání, jak využívat schopností starších počítačových odborníků v současných podmínkách neustálých a rychlých změn. Nevíme prostě, jak učinit programování celoživotním povoláním a jak zajistit programátorům pocit osobní jistoty.

Podle zahraničního tisku
zpracoval dr. Josef Gregor

ZE ZAHRANIČNÍHO HUMORU



„A to si říkáš programátor?“