

1.3 Kriteria hodnocení:

Na základě dobře míněných rad budeme provádět hodnocení hledáním odpovědí na tyto otázky:

- a) Kdyby přišel náš programátor na průměrné pracoviště kdekoliv na světě
 - aa) ušivil by se ?
 - ab) vynikl by ?
- b) Kdyby přišel náš programátor na špičkové pracoviště
 - ba) ušivil by se ?
 - bb) vynikl by ?
- c) Je náš software prodejný (exportovatelný) ?
- d) Vznikl u nás softwarový produkt, který by představoval světovou špičku (něco jako dBASE, Sidekick, Lotus 1-2-3 a pod.) ?

Laskavý čtenář nechtě se nyní pokusí odpovědět na tyto otázky (zajímavé je též sledovat, co všechno přitom člověka napadne); teprve pak je dobře připraven pro hltání dalšího textu.

2. Vývoj úrovně

2.1 Doba klasická:

Na počátku bylo slovo a to slovo mělo 6, 12, 16 ale i 32, 48 nebo 64 bitů. Instrukce vypadala třeba takto:

OP30500R610805

a v paměti bylo možno uložit až 128 údajů a 384 instrukcí (citují slavný stroj Cellatron SER2d o výkonu 38 parních duší - dle S. Lema).

S jásotem byly uvítány superstroje s pamětí až 4000 slov (šestibitových) a s ďábelsky vyayšleným assemblerem; instrukce měly příznaky, přívlastky, doplňky, flag-bity a jiná kouzla, adresování bylo možno považovat za thriller. Když programátor zapomněl napsat na konci příslušně opentlený HALT, stroj vběhl do dat a začal je interpretovat jako instrukce, což

začasté vedlo k žertovným scénám a jevům. Program nikdy nespádl do systému z toho prostého důvodu, že žádný operační systém nebyl.

Atd.

K obveselení mládeže bylo by možno dále pokračovat v řešení pravěkých problémů, ale to není cílem tohoto pojednání. Existuje však ověřený a z hlediska zvoleného tématu zajímavý fakt: naši programátoři si tyto klasické počítače ochočili a ve své době představovali světovou špičku. Na vlastní oči jsem viděl na zmíněném již Cellatronu zpracovávat agendu finanční účtárny středně velkého podniku a jiná, stejně grandiózní kouzla; i světoví znalci se v tisku rozplývali chválou na vysokou úroveň našich programátorů.

Naše programy byly tehdy ve světové špičce (pokud existovala technická a administrativní možnost srovnání); namátkou vzpomínám na moduly, které vznikly u nás a byly sařazeny do systémových knihoven počítačů Hewlett-Packard.

2.2 Doba moderní:

V klasických časech bylo programování kombinací dřiny, žemesla, umění, vědy, logiky a fantazie; programovali jenom duševně i tělesně odolní fanďové, kterým jedinou odměnou byla poslušnost stroje. Profese byla atraktivní pouze záhadně znějícím názvem. Řada slabších jedinců po nedlouhé době končila v léčebných ústavech.

Pak se objevily operační systémy, Fortran, Coboly a jiné pomůcky. Ve světě začínala tovární výroba programů. U nás se programování stalo slušně placenou profesí a práce s výpočetní technikou módním povoláním. Jak jsem kdysi (Havířov 1977) napsal, dostali se mezi analytiky a programátory lidé, kteří neměli valné znalosti, zato ale tím více hovořili a sepisovali studie a systémové koncepce. Aby se vyhnuli namáhavé práci (a protože normální programátoři neměli na kariéru čas), stávali se tito bystří šuhajové šéfy, specialisty a poradci; vybudováním teplých, příjemných a neotřesitelných postů dosáhli značné moci. K nim se posdějí přiřadila protekční dítko, kterým hodní strýcové dali do vínku akademické a vědecké tituly.

Na principu zhoubného bujení vznikla také rozsáhlá grupa vyhytralých chasníků, kteří se shodou okolností nedostali mezi kapitány našeho software; ti pak vzhledem k neschopnosti programovat stali se metodiky.

2.3 Současný stav:

Výše popsaný přerod v řídicí sféře nezlikvidoval (ač je to a podivem) programátory jako třídu. Měl však některé velmi nepříznivé důsledky. Pro posílení moci bylo na př. pro softwarovou vládnoucí kastu účelné udržovat enklávy vlastních programátorů v izolaci a zajišťovat jim dostatečný přísun (jakékoliv) práce. Zde má kořeny na př. opovrhlivé odmítání typového software, neexistence výměny programů atd.

Izolace programátorů spolu s rozmnožením selfmademandů vedla také k tomu, že profesní vyspělost našich programátorských pracovišť je dnes naprosto různorodá. Z odborného tisku i z různých seminářů sjišťujeme propastná rozdíly mezi způsoby myšlení programátorů. Někde vznikají slušné a užitečné programy, jinde najdeme přístup a styl myšlení první, nejvýše druhé generace. Někde pracuje s uživatelem moderně (Dr. Jiříšek - Ostrava 1986, Inorga - "Prototypový přístup" a pod.), jinde prohlašuje vedoucí programátorů (a myslí to smrtelně vážně): "ať uživatel sepíše vstupy a výstupy a dodá popis algoritmu, my to pak sprogramujeme"; takový jinoch působí velké ekonomické i morální škody a není si toho vůbec vědom, neboť s celé Computer Science a přílehlých oborů pochytil pouze jedinou větu - "se ASK odpovídá uživatel".

Naznačený historický vývoj ukazuje závažný vliv úrovně řídicích a organizačních prací na rozvoj programování u nás (blíže se k němu vrátíme v odst. 3.3) a vede ke dvěma předběžným závěrům:

- úroveň našich programátorů je velmi různorodá
(na jejich vlastní vinou)
- úroveň našich programů celkově zaostává.

Ke konečnému soudu musíme však prostudovat ještě řadu jevů, které si rozdělíme na objektivní (týkající se programování

jako profese) a subjektivní (týkající se programátora jakožto člověka).

3. Objektivní vlivy

3.1 Technika:

Jestliže jsme před 15 lety konstatovali, že náš stát se stává "technickým muzeem výpočetní techniky", měli jsme všichni na myslí nesmírnou různorodost typů výpočetních prostředků, které se u nás vyskytovaly; nebylo snad ve světě firmy, která by k nám nedodala alespoň jeden počítač.

V současné době sice už ve světě tolik nenakupujeme, ale o rozmanitost se starají naši výrobci; není snad ve světě typ počítače, který by naše výroba nezkusila alespoň v malém množství udělat (neříkám sáhnout "vyrobit"). V pojmu "technické muzeum" nabývá převahy funkce historická: typová rozmanitost se snižuje, zahrnuje však produkty vesměs zastaralé. A co je nejhorší, řada prostředků nám chybí vůbec.

Malá paměť je přece jen paměť. Dobrý koutelník v ní ledačco dokáže. Nemám-li však grafickou obrazovku ani plotter, prostě nemohu dělat grafiku. Nemám-li prostředky pro práci s barvou, nemohu barvu využívat v programech. Nemám-li winchester, nemohu na něj uložit data. Atd.

3.2 Obchod software:

Jakýkoliv obchod (nebo výměna nebo distribuce nebo něco podobného) s programovými produkty u nás neexistuje. S tím souvisí problém typových projektů, který jsem popsal dříve (Ostrava 1987); ač sice vznikají, nevíme o jejich širokém využívání. Co je však horší, nemáme ani obecně užitečné univerzální programy (viz můj příspěvek pro Ostravu 1985), kterým se podle nové terminologie říká "funkční moduly". V poslední době se u nás rozšiřují některé produkty světových firem, na př. dBASE, který lze dokonce sehnat ve verzi anglické, německé nebo i české. Naši spíčkovi tvůrci software se zřejmě zaměřili

na překládání textů v příkazech PRINT a WRITE, především však na výměnu firemních klaviček za vlastní. Do světového rozvoje jsme žádným novým nápadem nepřispěli.

Pokud se vykytne něco dobrého a zajímavého, pak je to na bázi zlepšovacích návrhů (jediný náš databankový systém REDAP - v době svého vzniku moderní a efektivní - je rozšiřován jako ZN; do profesionální dokonalosti mu chybí právě to, že byl dělán neprofesionálně, t.j. ve volném čase a po pracovní době). Chybí propagace a reklama (o tom, že v Kancelářských strojích vznikl velmi elegantní ASŘ pro personální počítače různých typů se musíte dozvědět od známých, popisy a nabídky nutno získat pokoutně a nemáte-li Xerox, musíte si je opsat).

Chybí dokonce i odborný tisk, který by se zabýval výhradně projektováním a programováním.

A stejně jako v technice některé programové prostředky nám chybí vůbec (znáte někdo na př. EXPERT SHELL ??).

3.3 Organizace prací:

Stalo se u nás dobrým zvykem (a nejen v programování), že lidé nedělají to, co umějí. Špičkový programátor vědeckotechnických výpočtů projektuje MZ, odborník na databanky píše programy pro výpočet vetknutých nosníků. Jestliže je v podniku klíčová agentura, na kterou kdeco navazuje a kterou kdeco potřebuje, pak je koncepce i realizace svěřena partě mláďenců, kterým činí potíže pochopit příkaz REDEFINES.

Rozsáhlý programový systém pro rezervaci místenek ČSD ("ARES") byl včetně software objednan v zahraničí; osobně znám spoustu lidí, kteří by na takové lahůdce rádi dělali.

Vyjímečně se dostanou správní lidé k odpovídající práci. Pak nastoupí další - ryze organizační - neduhy: lidí je vždy o něco méně, než by práce vyžadovala (ačkoliv na zcela nesmyslných a naprosto neúčinných projektech se šíví nadměrné kapacity po neuvěřitelně dlouhou dobu); do koncepčních prací mluví kde kdo; není: strojový čas, tabulační papír, písárka,

fungující počítač; platové podmínky a "hmotná zainteresovanost" jsou - slušně řečeno - podivuhodné. Atd.

Situaci komplikují "organizátoři", zmínění v odst. 2.2 a 2.3. Velký šéf se někde dolechne, že existují počítačové sítě. Druhý den oznámí nadřízeným, že za účelem progresivního rozvoje vytvoříme předpoklady a zabezpečíme opatření pro realizaci rozsáhlé počítačové sítě. Pak svolá podřízené a uloží jim rozpracovat svůj geniální nápad. Odborníci se tváří skepticky (jsem malý samostatný podnik s 300 zaměstnanci, umístěný v jediné budově), věci se tedy chopí vždy přítomná "hvězda" (která léta nic nedělá a v poslední době jí hrozí nebezpečí nějaké opravdové práce) a myšlenku převede do pestrých schemat doplněných obširnými texty. Konec je zajímavý: na věci se (bez konkrétních výsledků) živí několik "hvězd" tak dlouho, dokud se nevyskytne nový, ještě módnější pojem; na ten se pak hvězdy s jásáním vrhnou, původní téma je zapomenuto. Vytváří se pověst špičkového nesmírně progresivního pracoviště. Znalci ovšem vědí oč jde, ale kromě pocitu uspokojení je jim tato vědomost k ničemu.

3.4 Výchova a výuka:

Vždycky platilo, že dobrý pracovník vyrůstá tehdy, pracuje-li s někým zkušeným, kterému se může "dívat přes rameno". V programování dokonce nebylo (v dobách klasických) jiné cesty.

Dnes už nelze tento způsob výchovy obecně doporučit. Jak jsme zjistili v odst. 2.3, existují u nás obrovské rozdíly ve způsobu myšlení programátorů; tabulková kvalifikace, praxe a postavení v zaměstnání nikterak nesouvisí s úrovní programátora. Mladý adept nedokáže odhadnout, komu se dívat přes rameno a může se mu stát, že je (třeba i v dobrém úmyslu) vychován úplně špatně.

Škola může naučit teorii, programovací jazyky, strukturované programování atd.; zájmové kroužky podněcují zářavé fandovství. To všechno je nezbytné, ne však postačující (nutno zde vyslovit dík našemu školství, Svazarmu a dalším organizacím, které se bez ohledu na různá hlediska a stanoviska snaží

dát mládeži alespoň slušný základ). Také se nám probudili teoretici (které rychlý rozvoj počítačů původně zaskočil) a rozvíjejí spory typu "Pascal versus Basic", "Jackson versus něco jiného" a podobně, aby to nebylo tak jednoduché.

Mládež většinou zdatně odolává a nenechává se zastrešit.

Shrnutí: Začínající programátor může u nás slušně odstartovat. Zůstává však otázkou, co z něj udělá práce.

4. Subjektivní vlivy

4.1 Je programátor člověk?

Při zkoumání činitelů, ovlivňujících úroveň programátora se nevyhneme zajímavému jevu, kterým je postavení programátora ve společnosti. Najdeme zde též diametrálně odlišné extrémy: v některých lokalitách je programátor chápán jako bezprávný otrok, jinde je suverénním pánem a despota.

Otrok je špatně placen, neustále plísněn a zavalován neúměrným množstvím (často zbytečné) práce; jakmile se otrokové postavení zafixuje, pak programátora sužují nejen přímí i nepřímí nadřízení, ale také uživatelé a v pokročilém stavu i náhodné návštěvy. Otrok nemůže ani pomyslet na získání odborné literatury, na účast na seminářích, na styk s kolegy z jiných pracovišť a pod.

Z pána a despota mají hrůzu především uživatelé (i potenciální), ale kupodivu i celý řídicí aparát organizace. Pán tyranizuje okolí, dožaduje se západních časopisů (a dostává je, ač umí jen česky a to ještě bídě) a musí mít na svém stole nejnovější PC. Jestliže občas zplodí program, bývá to dost úděsné.

Extrémní postavení nepůsobí na vývoj úrovně programátora příznivě. Ani despota ani otrok nebývají dobrými programátory. Je-li však otrok přesazen do normálních podmínek, může být ještě zachráněn; pán a despota nikdy.

Nejpříznivější pro růst úrovně programátorů jsou nenápadná dobře řízená pracoviště, kde programátor zaujímá odpovídající

společenské postavení. Špičkový programátor pak vzniká tam, kde je mu navíc umožněno dělat dobrou a především užitečnou práci.

4.2 Běbrův zákon a jeho aplikace:

Autor tohoto příspěvku formuloval na základě dlouholetých výzkumů všeobecně platný zákon:

"Eroticky zdatný programátor je vždy dobrý programátor."

Zákon platí statisticky pro přibližně 93,8 % celé populace programátorů; platí stejně pro muže i ženy. Nelze ho aplikovat zpětně (dobrý programátor nemusí být eroticky zdatný) ale samozřejmě platí negace: špatný programátor není nikdy dobrý v erotice. Na podrobnou analýzu zde není místo, připomeňme si jen, že pro erotiku i programování potřebujeme stejné předpoklady: představivost, fantazii, přesnou kombinaci něhy a násilí, umění vojit se do skrytých potřeb partnera (programátorovi má být partnerem nejen počítač, ale i řešená agenda) atd.

Pro účely naší studie plynou z citovaného zákona poměrně radostné závěry: protože ani špatná technika ani špatná organizace práce nemohou zlikvidovat eroticky zdatné jedince, můžeme na otázku, zda u nás existují dobří a špičkoví programátoři odpovědět jednoznačně kladně. Důsledná aplikace zákona pak naznačuje, že ani v budoucnu nemusíme mít o úroveň našich programátorů obavy.

4.3 Programátorky a jejich význam:

V anketě "Programování '87" byl na autora vznesen požadavek, aby věnoval celý příspěvek problému "žena - programátorka". Osobně si myslím, že se ženská otázka v programování poněkud přecanuje; vyšlo by asi nastejno sepsat pro naše kolegyně rozbor "muž-programátor a co s ním".

V době klasické bylo programování doménou mužů. Ženám se z hlediska profese nepřikládala žádná důležitost, neboť prý nemají potřebnou exaktní logiku. Omyl spočíval v tom, že programování žádnou "exaktní logiku" nevyžaduje; programátor(ka) potřebuje zcela jiné vlastnosti, jejichž výčet přesahuje

rámec tohoto příspěvku (viz též odst. 4.2; téma "vlastnosti potřebné k úspěšnému programování" považují nícméně za atraktivní a doporučitelné alespoň ke kulturním diskusím).

Z hlediska našeho tématu můžeme konstatovat, že programátorská produkce přináší kladné výsledky a pozitivně ovlivňuje úroveň programátorů (oběho pohlaví). Muž vytváří nejlepší díla, když se chce vytažovat před ženou; parádní program však ocení jedině programátorka (oč jednodušší te mají umělci, stavitelé mostů, lovcí, sportovci atd.). Už jen tato zásluha programátorek opravňuje jejich existenci. Programátorky však také zvládají některé úlohy lépe než muži (o tom viz můj příspěvek o interaktivních systémech - "Programování '84").

Programátorský kolektiv, složený ze samých mužů psychicky strádá a snaží se oslnit alespoň operátorky, což může vést k celkovému poklesu úrovně (nemluvě o nedostatku erotických stimulů - viz Bébrův zákon). Kolektiv, složený ze samých programátorek není ovšem radno ani si představovat.

5. Shrnutí

5.1 Odpověď:

Nyní již můžeme jednoznačně odpovědět na otázku, položenou v úvodu tohoto příspěvku:

- a) Úroveň našich programátorů je velmi různorodá. Existuje však zdravé jádro, které dlužno považovat za normální. Každý náš "normální" programátor by se uživil i na špičkových pracovištích a sáme i pracovníky, kteří by kdekoli vynikli.
- b) Naše průměrné programy nedosahují úrovně světového průměru a špičkové programy u nás nevznikají.

Jestliže někomu připadá toto tvrzení poněkud pesimistické, nechť si přečte motto v záhlaví příspěvku; autor by byl rád v této oblasti příjemně překvapen.

5.2 Příčiny a vlivy:

Jak vyplývá z výše uvedených úvah, jsou hlavní příčiny současného nepříliš dobrého stavu v oblasti programového vybavení zhruba tyto:

- nevhodná organizace tvorby a "distribuce" software
- nedostatečné vybavení (technika, literatura, pomůcky, ...)
- celkové tápání ve sféře typového a univerzálního software,

Úroveň programátorů je výrazně ovlivňována

- celkovým pracovním klimatem pracoviště
- postavením programátora na pracovišti, v podniku i ve společnosti
- výchovou programátora v pracovním procesu
- mírou izolace programátora (ať už vzhledem k literatuře, kolegům a kolegyním nebo k světovým špičkovým produktům).

Velké nebezpečí pro úroveň programů i programátorů vzniká tam, kde bují konzervativní a zápecnické přístupy.

Stejně nebezpečí přináší formální "využívání" moderních a progresivních prvků jako módních pojmů bez zřetele k obsahu.

Nakonec bych chtěl upozornit ještě na jednu velmi důležitou a závažnou skutečnost: úroveň programů i programátorů se obráží v celkovém přístupu společnosti k výpočetní technice jako celku i k programování jako profesi.

5.3 Závěr:

Očekával-li někdo v závěru tohoto příspěvku jednoznačné rady a pokyny (co dělat, jak to dělat, proč tak a ne jinak) bude zklamán. Úmyslem autora bylo pouze snést náměty k vlastním úvahám vyspělých čtenářů. Situaci nelze změnit výnosem, příkazem či nařízením.

Jestliže však předchozí kapitoly přispějí k tomu, abychom si lépe uvědomili reálnou skutečnost, pak tento článek splnil svůj účel.