

PŘEVOD PROGRAMŮ NEBO INOVACE ?

Ing. Vlastimil Čevela, VS Ingstav n.p. Brno

Předkládaný příspěvek se zabývá problematikou převodu projektů zpracování dat na jiný typ počítače a operačního systému, než pro který byly původně určeny. V takové situaci bývá zpravidla položena otázka, zda se převod má uskutečnit pouhým mechanickým převodem zdrojových programů, anebo zda má být připraveno zcela nové řešení. Evidentně jasné případy, jako je úplná kompatibilita systémových prostředků i programovacích jazyků, nebo existence emulátoru, zde diskutovány nebudou.

Jsem si plně vědom rozdílnosti přístupu k tomuto problému, i nestejných výsledků, dosahovaných v různých organizacích při jeho řešení. Ve svých úvahách vycházím ze zkušeností, získaných při asi 3 roky trvajícím přechodu 3-směnného provozu z počítače Tesla 200 na 3-směnný provoz počítače EC 1026 ve výpočetním středisku Ingstav Brno. Názory, které budou uváděny jsou proto ovlivněny konkrétním prostředím a nakonec i jiný úhel pohledu na tytéž události by patrně přinesl odlišné poznatky.

Hlavní důvod, proč si přesto troufám formulovat některá provokativní konstatování je skutečnost, že každý přechod na jiný systém vyvolává značné nároky a náklady, především v analyticko-programátorské oblasti. Proto i diskutabilní tvrzení mohou být předmětem k zamyšlení, jak účinně přispět ke snížení těchto nákladů a ke zryhlení a zkvalitnění přechodových prací.

Výchozí podmínky nejsou všude stejné a navíc se mění i s časem. Existují však některé obecnější faktory, které mohou přechod podstatně ovlivnit. V první řadě je to počítačové prostředí, ale i kvalifikace vývojových a provozních pracovníků, úroveň projektové dokumentace, uživatelské prostředí, celkový způsob řízení prací a pod. O některých z nich bude tedy šířeji pojednáno v následujících odstavcích.

Počítačové prostředí

Bez důkladné znalosti konkrétního operačního systému a provozní vlastností počítače není možné vytvořit prakticky použitelný projek

Konkrétně nemohu určit rozdělení do úloh a programů, ani navrhnout datové soubory či způsob práce s číselníky. Nevím totiž, zda mohu využívat index-sekvenční soubory nebo databanku, neznám možnosti ladění a údržby, nedokážu domyslet vlivy nespolehlivosti počítače atd.

Firemní školení o počítačích a jejich operačních systémech, zvl. na počátku jejich zavádění informace tohoto typu prakticky neposkytují. Na kurzu Cobol Dos-4 se sice např. dozvím, že u index-sekvenčního souboru existuje možnost až 15 sekundárních klíčů. Ale teprve až se pracně dostanu k reálnému vytvoření souboru stavu a pohybu zásob materiálu o 150 tisících záznamech tak zjistím, že již přidáním jednoho sekundárního klíče se mi denní cca 2-hodinová aktualizace protáhne o 50 % a samotné zavedení souboru z pásky o 100 %. A protože mám tu smůlu, že pracuji v plně vytiženém provozu, který musí dodržet hodinu expedice pro vlakovou poštu na závody - tak se musím rychle vzdát lákavé představy, že bych mohl interaktivně poskytovat informace o výskytu materiálů na jednotlivých skladech.

Nebo se brdě hovoří o možnostech 64 terminálů a já ani nevím kolik souběžně možných procesů na počítači EC 1026. Reálná praxe však je taková, že když mají být ve špičkách splněny termíny expedice, mohou jet vedle sebe maximálně 2 dávky a po velikém dohadování se z deseti terminálů pouští programátorům tři anebo jeden pro více-uživatelský editor Luisa, který se navíc musí střídat s bloky překladů. O interaktivním přístupu odborných útvarů přes Duors přitom už vůbec nemůže být řeč. Odesvu 10 vteřin a dále sice vytrénovaný programátor v některých případech vydrží, ale uživatel musí zpravidla mít za určitou dobu nějaké konkrétní výsledky ...

Aby nezůstalo jen u EC 1026 a Dos-4, podívejme se do oblasti aplikace personálních počítačů a představme si, že máme úkol zvážit reálnou možnost zpracování evidence příjmů-výdejmů a vedení skladových karet velkého skladu na decentralizovaném počítači. Ponechme stranou možnosti získání takového počítače, ale když už si představíme konkrétní typ - kde je možno získat profesionálně podložené informace o datových rozsazích, dobách odezvy, časové náročnosti různých přístupů a dalších parametrech, které jsou zcela nezbytné pro posouzení reálnosti projektu pro konkrétně zadané podmínky ?

Nebo otázka zabezpečení dat. Můžeme po zkušenostech s BLR disky a našimi MP mechanismy předpokládat, že pevné disky na PC počítačích nepotřebují zabezpečovací kopírování? A jestliže potřebují, tedy jak často, na jaké médium (zařízení) a jaká je časová náročnost takové kopie a příp. obnovy - a takto bychom mohli pokračovat dále.

Podle mého názoru z těchto a podobných příkladů vyplývá jedno základní poučení. K tomu, abychom mohli zodpovědně rozhodovat o způsobech, jak budeme přistupovat k přípravě projektů v novém počítačovém prostředí, potřebujeme především znát něco jako návod k použití počítače.

Patrně nejvhodnější formou takového návodu by byla stručná, možná týdenní školení, doplněná příslušnou dokumentací, která by ukázala základní vlastnosti určitého typu počítače, se zdůrazněným zaměřením na jeho chování v reálných podmínkách. Počítač je sice univerzální automat, ale využívání některých jeho vlastností je vhodnější a jiné méně vhodné - a právě to by měl programátor či projektant vědět především.

Věcně by se do tohoto návodu například pro počítač EC 1026 zařadit takové body jako:

- možnosti souběžné práce více programů, časová náročnost, kterou stojí terminály (s vazbou na druh práce) a doporučené hranice multirežimu
- časová náročnost různých druhů třídění, kopírování a dalších standardních programových funkcí, včetně příp. žurnálů pro aktualizaci databanky
- nároky různých organizací rozsáhlých souborů (časová náročnost při tvorbě a práci s index-sekvenčními soubory, sekundárními klíči) a mese jejich praktické využitelnosti
- výhody a úskalí stránkování u virtuálních systémů, čím konkrétně lze stránkovací mechanismus kladně či záporně ovlivnit, jak vůbec využívat správy počítače o jeho vytížení
- vhodnost či nevhodnost různých příkazů a funkcí v dodávaných programovacích jazycích
- jaká jsou omezení využitelnosti číselnickových modulů

- praktické dopady reálně se vyskytující poruchovovosti a jak se jim bránit
- doporučení pro systém údržby knihoven programového vybavení a možnosti ochrany jejich prvků před neoprávněným přístupem či přepsáním
- doporučení pro efektivní práci s katalogem magnetopáskových souborů

Je zřejmé, že řada z uvedených problémových okruhů, včetně dalších, jako jsou nabídkové dialogy či systémy aktualizace a výběrů z bází dat platí obecně i pro operační systémy a počítače ostatních typů.

Školení, poskytující ve zhuštěné formě výše popisované informace by ovšem vyžadovalo též příslušně kvalifikované posluchače, tj. pracovníky s víceletou analyticko-programátorskou praxí. Museli by totiž umět posoudit rozdíly mezi tím co znají a novým počítačovým prostředím, a vyvodit z toho též příslušné závěry.

Kvalifikace a dokumentace

Čistě programátorské vědomosti v úzkém slova smyslu, bez věcné (analytické) znalosti řešené problematiky nejsou dostatečnou kvalifikací pro rozhodování o vhodné formě přechodu na jiný počítač a operační systém.

Pokud neznám, co má být zpracováváno, základní principy algoritmů řešení a charakteristiku požadovaných výsledků - nemám možnost volby. Jedinou reálnou cestou je totiž mechanický převod programů.

Mechanický převod programů může obecně přinést některé výhody. Především, jak bylo výše zmíněno, nevyžaduje podrobnou znalost problematiky, takže na něm mohou pracovat i méně zkušení programátoři. To ovšem platí pouze do té doby, dokud se nevyskytnou problémy - např. nový systém nezná některé praktiky, použité v převáděných programech. Pak se toho bez znalosti věci moc nepořídí. Další výhodou může být čas, tj. tento způsob bude pravděpodobně rychlejší.

Podívejme se však i na nevýhody mechanického převodu zdrojových programů. Za nejzávažnější je nutno považovat fakt, že se takovým přechodem konzervují morálně zastaralé programy, se všemi nevhodnými či nedotaženými algoritmy a chybami. Odpadá tak i možnost revize víceleté postupné aktualizace programového vybavení a vyřazení překonaných prvků a pravidel. Dále se dobrovolně vzdáváme většiny přínosů, které vyplývají z přechodu na kvalitativně lepší nový systém, jako je např. využití indexsekvenčních souborů, virtuálních výstupních front, velkých vnitřních a vnějších pamětí a pod.

Základní podmínkou úspěchu při rozhodnutí o inovaci projektu ovšem je kvalifikovaná věcná znalost řešené problematiky. Analýza je totiž záležitost pracná a především časově náročná. Jde o práci s lidmi - v tomto případě s odbornými uživatelskými útvary, a tu zpravidla není vhodné příliš uspěchat.

Významnou pomocí může být kvalitní dokumentace. Nepředpokládejme, že ji získáme od uživatelů - ti po úspěšné víceleté počítačové realizaci jejich agendy často už ani neví, jakými algoritmy jsou jejich data zpracovávána. Navíc pak bývají pochopitelně značně nedůtkliví, když se po nich znovu vyžaduje ucelené a důsledné definování problémů, kterých se přece už dávno zbavili.

Ideálním případem je existence aktivních všeobecně srozumitelných dokumentačních prvků, jako jsou např. číselníkové moduly, dále dobře komentované datové popisy, funkční popisy programových prvků a podobně. Neexistující nebo neudržovaná projektová dokumentace pak způsobuje vážné potíže. Nejen, že není možné získat znalosti o věcné náplni a způsobu počítačového řešení jako podklad pro inovaci, ale i jakýkoliv problém při mechanickém převodu pak vyžaduje detailní práci se zdrojovými texty programů. A čtení předhistorických zdrojových textů z doby, kdy nebyla známa užitečnost 3 základních struktur sekvence-větvení-cyklu a bez omezení se používaly skoky, to je většinou pořádně náročná a zdlouhavá křížovka.

Nejspolehlivější základnou pro inovaci projektu je tedy důkladná věcná znalost řešené problematiky, platná dokumentace a dobré kontakty s uživateli. Potom je možné zapomenout téměř vše, co se v kdysi existujících podmínkách vytvořilo a později více či méně vhodně přidávalo, přizpůsobovalo a předělávalo. Vaníká zde totiž šance začít řešit projekt úplně znovu, s využitím všech dosavad-

ních zkušeností i nově se nabízejících možnostech počítače i operačního systému.

V zájmu serióznosti je nutno uvést, že tento postup rovněž přináší různá úskalí. Jde přece o nový vývoj, vhodnost některých řešení je nutno ověřovat, uvažované předpoklady nemusí být vždy splněny atd. Znalost základních informací o tom, co má být vyřešeno a jakých cílů má být dosaženo jsou však tou nejspolehlivější zárukou pro dosažení konečného úspěchu. I když možná o něco později, než při pouhém mechanickém převedení programů, avšak s jedním základním rozdílem. Inovace projektu, provedená se znalostí věci poskytuje profesionálně spolehlivý základ pro další rozvoj počítačového zpracování příslušné oblasti. Nová realizace překonaných programů naproti tomu zpravidla přinese jen opakování dřívějších problémů a rozčarování uživatelů.

Jak je z uvedeného též zřejmé, velice důležitým faktorem je též kvalita převáděného programového díla. Přehledně a srozumitelně řešené programy mohou být úspěšně převedeny mechanicky, avšak méně kvalitní, nepřehledné a morálně zastaralé programy je lépe nahradit novými. Tak jako v jiných případech i zde však nejdůležitější roli hrají odborné znalosti zainteresovaných řešitelů.

Řízení převodu projektů

Jedním z nejméně příznivých příspěvků k řešení převodu projektů je přehnaně optimistická atmosféra a reklama, která uživatelům slibuje, že s novým počítačem se vyřeší všechny dosavadní problémy, vše bude dokonalé a pokud možno bez práce.

Zvláště pokud se použije pouze mechanický převod, tak zpravidla vůbec nedojde ke zlepšení. V důsledku násilného přizpůsobování se novým podmínkám může dojít dokonce i ke zhoršení některých užitečných vlastností projektu - např. ve formě sestav. Navíc, jakmile projekt na novém počítači funguje, uživatelé předpokládají další rozvoj a nechápou, že by bylo užitečné nejdříve programové řešení dotáhnout na úroveň, kvalitativně odpovídající novému systému.

Zdá se, že by bylo vhodné hovořit v uvedených souvislostech o vnitřní a vnější inovaci projektů. Vnitřní inovaci, která znamená přizpůsobení se novým podmínkám operačního systému a počítače

zpravidla uživatel nevidí, avšak může pocítit její kladné dopady, např. zkrácení termínů, větší pružnost při další aktualizaci programového vybavení a pod. Za vnější inovaci pak považujeme takové změny, které z pohledu uživatele přinášejí nové možnosti při zpracování informací pro jeho potřeby - např. dosud nevyužívané vazby, časové řady, větší pohotovost informací atd.

Dalším hlediskem může být rozlišení programové a datové inovace. Jestliže např. v novém řešení začnou být důležité datové údaje, které dříve nebyly využívány nebo důsledně kontrolovány, může dojít k závažným poruchám při nábětu takového zpracování. U hromadných dat totiž nelze např. ihned pro několik tisíc druhů skladovaných materiálů doplnit druhovost dle jednotné klasifikace či jiné podobné údaje. V takových situacích bývá nutno provádět různé přípravné, kontrolní, nebo i analytické práce a programy pro docílení potřebné korektnosti využívaných datových fondů. Těmito vyvolanými pracemi se pak samozřejmě zvyšuje především časová náročnost příslušného přechodu.

Závěrem je třeba zdůraznit jeden z nejdůležitějších vlivů. Při řešení přechodu na nový počítač a operační systém je totiž bezpodmínečně nutné omezit údržbu programového vybavení na nezbytné minimum a nepřipustit řešení žádných dalších vývojových úkolů. Jinak se analyticko-programátorská kapacita, která by měla zajistit přechod vyplývá na řešení problémů, v dané situaci vyložené okrajových. Důsledná opatření v tomto směru jsou sice z hlediska uživatelů nepopulární, ale naprosto nezbytná.

Předložené úvahy nevyčerpaly všechny problémy, související s převodem projektů. Jsou zde ještě další otázky, jako je inovace dokumentační metodiky, resp. technologie projektování pro nové podmínky, změny chování uživatelů v novém systému a pod. To by však již přesahovalo vymezený rozsah příspěvku.