

Přenos projektů Cobol DOS-4/EC pod DOS-NOVELL/PC

Vlastimil Čevela

Programátorské služby Cobol, Bencšova 279, 664 42 Modřice, Česká republika

Abstrakt

Předkládaný příspěvek je pokusem o prezentaci některých zobecněných zkušeností z převodu větších podnikových agend z mainframe prostředí DOS-4/EC do prostředí PC.

Při převodech byla využita jak metoda přesného "překlopení", s minimálními změnami zdrojových textů, tak úplné přeprogramování při zachování vnějšího uživatelského rozhraní s doplněním o možnosti interaktivní práce v síti Novell 3.x.

1. Nároky inovace programových projektů

Volba nejvhodnějšího způsobu inovace programového řešení určité agendy je vždy ovlivňována mnoha, často velice specifickými faktory. Za nejdůležitější vliv však lze stále považovat analytickou kvalifikaci řešitelů [1].

I když totiž existují jazyky 4. generace a řada velice výkonných programovacích nástrojů, těžiště problémů stále zůstává ve zcela konkrétní formulaci nejen obecných cílů, ale i všech detailů zadání.

A programátor dostatečně podrobnou analýzu řešené oblasti buď zná, a pak může realizovat skutečnou inovaci, anebo nezná, a pak musí existující algoritmy zpracování co nejpřesněji opsat, tj. převést s co nejméně změnami.

Nová analýza problému je totiž vždy velice náročná nejen na čas, ale i na úroveň komunikace a to jak mezi uživatelem a programátorem, tak mezi jednotlivými pracovníky v rámci těchto kategorií navzájem.

2. Charakteristika "algoritmu" zpracování

Je důležité si uvědomit, že algoritmem např. mzdové agendy nejsou pouze vzorce pro výpočty mezd. Konkrétně v tomto případě se jedná o velice složitý komplex činností od získání personálních údajů o pracovníkovi, přes informace o jeho pracovních výkonech v průběhu měsíce, až po souhrnné údaje ze štipačky.

Z uvedeného vyplývá, že pokud se nechceme omezit na triviální pořízení a zpracování dat jednorázově na konci měsíce, musíme znát nejen vzorce, ale i celou organizaci zpracování mzdové agendy v daném podniku.

3. Interakce verzus dávka

Klíčová slova současného programování jsou "interakce, klient-server, přímé přístupy ...". Zdaleka ne všechny podniky jsou však schopny okamžitě přejít na totálně transakční zpracování, kdy všechna data průběžně pořizuje a následně přímo využívá uživatel.

A není to jen problém nákladů na stovky pracovních stanic a vybavení sítě, ale i potřeba výrazně jiné organizace práce, technologické úrovně a pracovní kázně, než na jakou jsme v našich podmínkách dosud zvyklí.

Proto stále přetrvává potřeba dávkového zpracování, tj. vyvolání posloupnosti příkazů a programů k souvislému provedení více různých akcí. Dosažená rychlost procesorů a disků i na dnes již "obyčejných PC" navíc v řadě případů prakticky setře rozdíl mezi odpovědí interaktivního programu a předáním výsledku, získaného spuštěním dávky.

4. Interakce a dávka ve vývojovém prostředí

Pokusme se o bližší analýzu jejich užití při programování. "Interakce" je metoda, kdy počítač bezprostředně reaguje na právě zadaný příkaz, který je zpravidla vybírán z více možností. Za "dávku" v našem pojetí pak považujeme postup, kdy předem definujeme posloupnost akcí, včetně příp. větvení dle zadaných nebo průběžně získaných parametrů.

V interaktivním vývojovém prostředí tedy programátor vybírá z variant možných postupů a reaguje různým způsobem na vzniklé situace. Zdánlivě vynikající koncepce, avšak jen do té doby, než si uvědomíme, že programování je především opakování. Návrh - ověření - změna návrhu - ověření - atd atd atd.

Uvedený charakter práce se uplatní zvláště důrazně při přenosu dávkových úloh na PC. Jen zcela výjimečně totiž ladíme izolovaný program, ale téměř vždy musíme zajistit správné fungování určité posloupnosti programů, někdy i s hodně složitým algoritmem datových toků.

Problémem zde není jen zcela zbytečné opakování interaktivních akcí, ale v neposlední řadě též intelektuální náročnost dodržování stejných podmínek a postupů pro jednotlivé ladící cykly.

Při ladění formou "dávek" je sice nutné větší počáteční úsilí pro definici jednotlivých kroků, spojené mj. i s analýzou předpokládaných variant realizace. Ziskem však je nejen snadná opakovatelnost, ale především větší systematickosti celého postupu a průběžně vznikající dokumentace.

5. OMS pro kompilátory MS/MF Cobol a MX Cobol

Nástrojem, který se ve výše uvedených souvislostech nejvíce uplatnil, bylo kompletně české vývojové prostředí Objektově modulární stavebnice (OMS), popisované v [2].

Kromě běžných interaktivních funkcí, analogických např. s IDE Turbo C, obsahuje univerzální dávkový manažer a desítky prvků tzv. systémové podpory aplikací (SPA), které byly vyvinuty právě za účelem řešení výše popisovaných problémů, a to jak při ladění a ověřování, tak při rutinním zpracování dat.

Podstata použité dávkové filosofie spočívá kromě jiného v tom, že se počítá s automatickým prováděním řady funkcí nad množinou programů i dokumentačních souborů. Bez problémů je možno vyhledávat textové řetězce v desítkách či stovkách zdrojových textů, srovnávat obsahy celých adresářů a spouštět jedním příkazem aktualizaci překladů celého projektu nebo interaktivně ovládané a velmi snadno modifikovatelné ladící, ověřovací i rutinní dávky.

S využitím vhodných řídicích souborů a seznamů, vytvářených nástroji SPA pak lze pomocí hypertextového editoru EDA zajistit bleskový přímý přístup ke všem programovým a dokumentačním prvkům prakticky libovolně rozsáhlého systému.

6. Zdrojová kompatibilita MX Cobol

Kompilátory jazyka Cobol byly jako součást operačních systémů počítačů řady EC1021 až 27 (ale i dalších) pro VÚMS Praha vyvinuty v tehdejšími n.p. Kancelářské Stroje - závod Brno.

Díky invenci svých tvůrců se vyznačovaly nejen běžnými vlastnostmi dle požadavků norem jazyka Cobol, ale obsahovaly i mnoho nadstandardních možností navíc. Kromě různých systémových programů řady "IC" to byla především výrazně volnější pravidla pro tvorbu programátorských jmen.

Počítalo se i s přechodem do prostředí DOS/PC a UNIX. Proto byl vytvořen kompilátor MX Cobol, který překládá ze zdrojového textu Cobol do zdrojového textu jazyka C. Ve viru změn po roce 1989 se však podnik KS Brno orientoval do jiných oblastí, a tak počátkem roku 1993 zakoupila všechna práva pro distribuci i další rozvoj tohoto kompilátoru firma Čevela - programátorské služby Cobol.

Současný vývojový systém MX Cobol 2.0 OMS sice nemá všechny vlastnosti dle normy ANSI Cobol 85 a má proti ní i některá omezení. Zachovává však zdrojovou kompatibilitu na EC a to i se všemi nadstandardy, a umožňuje tak spolehlivý přenos programů do prostředí DOS/PC.

Navíc oproti EC má prostředky "screen section" pro tvorbu interaktivních aplikací i možnosti sdílení a zamykání souborů pro vývoj programů k užití v síti.

Kromě toho je vybaven objektovým generátorem programů GPC s rozsáhlou knihovnou typových modulů a vlastními nástroji na podporu některých prvků "CASE" [3].

Technologie práce s interaktivním nabídkovým manažérem MXM pak slouží nejen ladění a ověřování v duchu výše popisovaných zásad, ale umožňuje i tvorbu takových aplikací, ve kterých je řada činností koncového uživatele realizována pouze na pracovní stanici, bez zatěžování serveru.

7. Závěr

Při omezeném rozsahu byly uvedeny jen vybrané skutečnosti, související s generační inovací větších programových celků. Dalo by se hovořit i o ladících nástrojích pro sledování toku dat formou I/O charakteristiky, o výhodách objektového přístupu a prototypového způsobu řešení při komunikaci s uživateli a řadě dalších zkušeností.

V případě zájmu je možné získat další informace v rámci diskuze, nebo na kontaktní adrese.

Literatura

1. Čevela, V.: Převod programů nebo inovace?, Sborník Programování '88, DT ČSVTS Ostrava 1988
2. Čevela, V.: Objektové programování v Cobolu pomocí OMS Sborník Programování '94, DT Ostrava s.r.o. 1994
3. Čevela, V.: "CASE" nástroje MX Cobol OMS a zkušenosti s jejich využíváním, Sborník Tvorba software '95, Tanger s.r.o. Ostrava