

ZPRACOVÁNÍ HROMADNÝCH DAT  
V SÍTI POČÍTAČŮ TYPU IBM-PC/AT

Ing. Rudolf Ottis

*Tesla Rožnov*

Cílem tohoto referátu je shrnout některé zkušenosti, získané při komplexním analyticko - programátorském zpracování hromadných dat v síti počítačů IBM-PC/AT (XT).

V našem případě se jednalo konkrétně o problematiku zpracování obchodního případu pro PZO TESLA, Elektronické součástky - koncern Rožnov. Uživatel požadoval vybudování aparátu pro aktualizaci obecných číselníků, adresářů obchodních partnerů a číselníků vyvážených a dovážených výrobků; vstup dat, spojených s obchodním případem, tisk všech dokumentů, u zahraničních pak v příslušné jazykové mutaci; statistické zpracování pro vnitřní potřebu i pro nadřízené orgány a eventuelní možnost napojení na další útvary.

Technické zabezpečení celého systému tvoří 3 pevné disky po 120 M byte, cca 20 počítačů IBM-PC/AT nebo XT, některé jsou vybaveny vlastními pevnými disky, 2 společné a cca 14 lokálních tiskáren a 3 streamery pro úklid dat.

### Analytické zpracování celé úlohy

Samotná analýza celého problému je složitý a náročný proces a je nutno předem vycházet z toho, že nebude možno získat vyčerpávající informace, potřebné pro vlastní programování. Důvodů k tomuto tvrzení je hned několik :

Po celou dobu analýzy řešitelé komunikují pouze s vybranými reprezentanty ze strany uživatele. Kvalita celé práce je tak přímo ovlivněna tím, jak jsou tyto pracovníci schopni popsat celý řešený problém. I při pečlivé práci však dojde k

opomenutí různých detailů, odchylek či specifik, které se pak projeví až při vlastním zavádění úlohy do provozu.

Dalším faktorem kvality analytické práce je to, do jaké míry již uživatel má či nemá zkušenosti se zpracováním své problematiky na počítači. V podstatě se dá říci, že uživatelská zkušenost či nezkušenost má své výhody i nevýhody.

Pokud uživatel dosud nepřišel do styku s výpočetní technikou, je výhodou programátora to, že může nabízet postupně její výhody, tj. koncipovat systém zpočátku jako jednodušší s postupným přidáním dalších funkcí. Uživatel se tak nenásilně seznamuje se zpracováním a sám přináší podněty k jeho dalšímu rozšíření. Tento zdánlivě ideální postup však může přinést komplikace v tom, že uživatel opomene sdělit některá důležitá fakta, případně zjistí, že některé zásady, které předem stanovil, neplatí pro celý rozsah zpracování. V tomto případě pak může dojít v průběhu zavádění úlohy do provozu k podstatným změnám v celém řešení, včetně změny datových struktur. To komplikuje obzvláště v případech, kdy je již systém částečně v provozu a změny je nutno provést tak, aby nebyla znehodnocena již uložená data.

Naopak u uživatelů, kteří pouze přecházejí na jiný druh výpočetní techniky, může být ušetřeno spoustu práce, pokud se předem programátor seznámí se stávajícím způsobem zpracování. V novém řešení mohou být využity i některé stávající datové struktury, zvláště pak číselníky. V některých případech je možno přenést i přímo vlastní obsah souboru. Problematika, jak tento přenos uskutečnit, by byla sama o sobě námětem na samostatný referát. Uživatelé jsou pak na základě svých zkušeností schopni lépe formulovat své požadavky, jsou zvyklí svá data kategorizovat, poukazují na slabá místa původního řešení a přicházejí s konkrétními náměty na jeho rozšíření. Na druhé straně jsou však svým způsobem svázáni s původní verzí zpracování. Nové řešení velmi kriticky srovnávají s původním, mnohdy požadují jeho věrnou kopii a dostatečně si neuvědomují, že nový projekt řeší (měl řešit) celou problematiku mnohem komplexněji a tím tak ve výsledném efektu ušet-

řít práci větší skupině uživatelů než tomu bylo doposud.

Úplnost analytické práce je do značné míry také ovlivněna přesnou specifikací výstupních dokladů. Zpravidla se analýza zpracovává pro jeden z úkolů uživatele s předpokladem budoucího rozšíření celé úlohy i na další. Zvláště u dokladů, u kterých se v novém řešení počítá se změnou místa jejich vzniku (např. faktury se tisknou přímo na pracovištích namísto dovadního předávání podkladů k jejich centrálnímu tisku) může vést k řadě nepřesností, protože uživatel zcela nedocení tuto problematiku a k jejímu řešení tak vlastně dochází až při vlastní změně postupu zpracování.

Posledním faktorem je pak určitý maximalistický přístup uživatele, který požaduje plně možnosti zpracování na každém z terminálů a přístup všech uživatelů ke všem datům, i když diferencovaný vzhledem k jejich zařazení v hierarchii organizace. Praxe pak ukazuje, že tento přístup je příliš široký a že některé části řešené problematiky jsou v podstatě lokálního charakteru. Přehnané nároky na přístup k datům mají však většinou vliv na rychlost celého zpracování.

Stěžejní částí analýzy je vlastní návrh datových struktur. U víceuživatelského systému jde o sdílené datové soubory, ve kterých jsou všechny věty, obsahující data, doplněny určitým přístupovým klíčem, který diferencuje přístup k datům na úrovni jednotlivých útvarů uživatele. Tento klíč je tvořen hierarchicky a platí zásada, že čím delší klíč, tím užší přístup k datům. Přístup jednotlivých útvarů je řešen přímo nabídkou v základním uživatelském menu. To znamená, je vybudován vlastní systém přístupových priorit, kde má každý zaměstnanec přiděleno povolené vstupní menu a přístupový klíč v rámci svého útvaru.

#### Realizace řešení v databázovém systému FOXBASE

Pro vlastní realizaci celého projektu byl vybrán databá-

zový systém FOXBASE, verze 2.10, který je schopen pracovat i ve víceuživatelském prostředí. Z hlediska programátora je práce s tímto systémem velmi příjemná, poskytuje dostatečný komfort a přispívá ke značné efektivitě práce. Obsahuje generátory programů pro sběr dat a tisk sestav, i když jejich produkty jsou určeny spíše pro neprofesionální práci. Generační program FOXVIEW je však účinným pomocníkem při návrhu rozložení dat na obrazovce. Oproti systému DBASE III je celé zpracování mnohem rychlejší a navíc je i mnohem širší paleta příkazů, rozšířená o příkazy, vytvářející menu, umožňující uchovat obsah obrazovky, umožňující práci s indexovanými položkami a další.

Není smyslem této části detailně popisovat příkaz od příkazu, přesto bych se však zmínil o některých postupech, které jsou nutné k realizaci ve víceuživatelském prostředí nebo které přispěly k celkovému urychlení celého zpracování.

Jelikož jsou všechny datové soubory sdílené více uživateli, není využívát příkazy, provádějící vlastní fyzické rušení jednotlivých vět, v souborech. Toto je prováděno zvláštním chodem, spouštěným po ukončení běžného zpracování.

Dalším důležitým problémem je uzamykání jednotlivých vět za účelem jejich modifikace. Jsou dvě možné varianty, uzamčení pouze konkrétní věty nebo celého souboru. Jelikož evidovaná data jsou takového charakteru, že jsou rozdělit na části, modifikovatelné pouze jedním uživatelem a při tom zároveň prohlížené či statisticky zpracováváné větším počtem jiných uživatelů, bylo zvoleno uzamykání pouze jednotlivých vět. Toto umožňuje plynulý vstup dat do systému. Riziko toho, že při prohlížení nemusí být zobrazen přesně aktuální stav, lze u úloh tohoto typu připustit.

Potřebné setřídění dat je zajištěno použitím indexových souborů. Protože však zvláště při předčasném vypnutí počítače dochází k porušení rovnováhy mezi datovými a indexovými soubory, je nutno vytvořit zvláštní chod, umožňující přein-

otevření všech datových souborů.

Poměrně časově náročná je operace USE, provádějící otevření souboru. Nejlépe by bylo otevřít všechny soubory předem a nechat je otevřené po celou dobu zpracování. Tím je ale počet použitelných souborů omezen na 10, což je dáno vlastnostmi databázového systému. Proto je každému souboru přiděleno číslo, pod kterým je vždy otevírán a se kterými soubory se může na tomto čísle eventuálně střídat. Na začátku zpracování jsou pak otevřeny pouze soubory, potřebné v celém průběhu zpracování. Při volbě některé ze základních větví zpracování je pak kontrolována otevřenost dalších potřebných souborů a případně pak provedeno jejich otevření. Tento systém minimalizuje použití příkazu USE a zároveň zajišťuje, že nemůže být požadováno otevření souboru, který je již otevřen jinde. Uzavření všech souborů je provedeno po ukončení celého zpracování.

Pro vyhledávání souborů na mediích byly rozděleny do kategorií podle umístění (dovoz, vývoz, číselníky). Toto umístění je popsáno v souboru typu .MEM, který je rozehráván hned na počátku zpracování. Vlastní použití příkazu USE pak vypadá USE CESTA.SOUBOR, kde položka CESTA obsahuje popis jednotlivých adresářů. Tento způsob je rychlejší než použití příkazu SET PATH, který popisuje všechny možné eventuality. Při změně instalace stačí změnit popis v souboru .MEM.

Konečným popisovaným příkazem je příkaz SET FILTER. Tento příkaz demonstruje zásadu, že ta nejjednodušší cesta nemusí být ta správná. Oč jde? Tento příkaz umožňuje vybrat ze souboru věty, vyhovující dané podmínce a s těmito větami pak pracovat jako se samostatným souborem, včetně možnosti testování jeho začátku a konce. V praxi však počítač stále prohlíží celý velký soubor, což znamená, že tato operace je velmi zdlouhavá. Proto je lepší celý soubor vhodně oindexovat, vyhledat příkazem SEEK první větu, vyhovující podmínce našeho výběru a vlastní pohyb po souboru pak realizovat instrukcí skoku s následným testováním, zda je podmínka pro výběr ještě splněna či nikoli. Je ale třeba mít na zřeteli, že při změně některé z

indexovaných položek se věta nejprve zařadí na nové místo v souboru a pak je teprve prováděna operace skoku již v novém okolí.

Osvědčilo se také speciální využití 22. až 24. řádku obrazovky. Řádek 22 je vyplněn dvojitou čarou, na řádku 23 je stručně popsána činnost, kterou má uživatel provést nebo chyba, které se dopustil. Na řádku 24 je pak stručně popsán význam funkčních kláves, které lze v dané fázi zpracování použít. Výhoda tohoto uspořádání spočívá v tom, že v případě ladění lze nastavit parametr SBT STAT ON a sledovat tak pohyb po jednotlivých souborech, aniž by bylo narušeno zobrazení dat na obrazovce.

Toto byly některé z postupů, vedoucích k efektivní činnosti celé úlohy a jistě by bylo možno najít i mnoho dalších. Hlavní zásadou je pochopitelně zabránění vytvoření kritických stavů a minimalizace přístupu na disk, ke které lze s výhodou využít instrukce SCATTER a GATHER, které provádějí přenos položek věty souboru do paměťové indexované proměnné a naopak bez ohledu na způsob jejich uložení.

### Zavádění úlohy do provozu

Prubířským kamenem kvality celého díla je pak uvedení celé úlohy do provozu. V této fázi je vlastně celá práce podrobena ostré a pravdivé kritice všech jejích skutečných uživatelů.

Celé zavádění je možno rozdělit na dvě základní etapy.

První z nich je seznámení uživatelů s vlastní obsluhou výpočetní techniky. Tuto část nelze v žádném případě podcenit a je třeba mít na zřeteli, že většina uživatelů (aniž bych se tímto chtěl jich jakkoliv dotknout) přistupuje k výpočetní technice jako naprostí laici. Proto je nutno je detailně seznámit s obsluhou počítače a vysvětlit jim postup při řešení

eventualních nestandardních situací. Zvláště pečlivě je nutno popsat klávesnici a vysvětlit umístění a funkci těch kláves, které budou při zpracování používány. Kdybych neačel možnost se sám přesvědčit, nikdy bych nevěřil, že i tak pro programátora běžná záležitost, jako je současné používání kláves CTRL a END, může být pro uživatele nepřekonatelným problémem. Většinou se snaží o krátký okamžitý stisk obou kláves současně, místo aby v klidu podržel klávesu CTRL a stiskl krátce klávesu END. Nechci zde popisovat postup, jak naučit pracovat s klávesnicí, myslím však, že tento příklad je dostatečnou ilustrací o tom, že tento problém nelze nikterak podceňovat.

Druhou etapou pak je vlastní seznámení a řešení úlohou. Zde se setkáváme s různým přístupem uživatelů.

Většina z nich má zvláště z počátku strach, že stisknutím chybné klávesy okamžitě zničí data nejen sobě, ale i všem ostatním. Tato obava však trvá jen krátce a s postupným zapracováním se mizí. Uživatelé přicházejí s připomínkami, které je třeba evidovat, posuzovat jejich důležitost a postupně je zapracovávat do řešení. Někdy jde i o připomínky zásadního charakteru, které zcela mění některá fakta, uvedená v zadání. Zároveň je ale třeba je rychle zapracovat do systému, protože v mnoha případech v podstatě brání jeho dalšímu využívání. Setkáváme se ale i s takovými jevy, že uživatelé záměrně bojkotují nové řešení, ať již z jistoty, že nebyli přizváni k tvorbě nového řešení nebo z nechtění změnit zažitý způsob práce, protože každá změna přináší zvláště na počátku více problémů než vlastního užítku.

Důležitým pomocníkem programátorů může být pracovník uživatele, mající na starosti výpočetní techniku, tzv. správce báze dat. Jeho přístup k celé problematice do značné míry ovlivňuje zprovoznění celého systému. Tento pracovník by měl soustřeďovat jednotlivé připomínky a předávat je programátorům a tím je vlastně odstínit od ostatních uživatelů a vytvořit jim tím lepší podmínky k jejich další práci. Zároveň by měl řešit vnitřní organizační problémy, spojené s přechodem

na výpočetní techniku a kompenzovat i některé negativní jevy, popsané v předchozím odstavci.

Zásadní podmínkou úspěšné práce celého systému je zavedení uživatelské kázně a respektování jistých provozních zásad, které jsou dány použitým typem výpočetní techniky. Zejména u víceuživatelských systémů nabývá tento požadavek značné významnosti. Je velkým neštěstím uživatelů, že např. vypnou svůj počítač v průběhu zpracování. Nutno však na druhé straně přiznat, že zvláště, byl-li systém připravován na samostatném počítači, může z počátku dojít ke stavům, kdy není jiné východisko. Po stabilizaci činnosti celého systému by mělo však být každé zpracování řádně ukončeno, neboť jinak může dojít k narušení vazeb mezi jednotlivými daty.

Co říci závěrem? U systémů, určených k současné práci více uživatelů, není možno z výše uvedených důvodů zpracovat úvodní analýzu do všech detailů. Proto je nutno ponechat řešení určitou pružnost, která umožní pozdější realizaci změn nebo specifik jednotlivých částí celé úlohy. Při řešení velkých celků je pak výhodné provádět celé zavádění postupně.

Ve vztahu k uživateli je pak nutno vyvolat u nich určitou tvůrčí aktivitu a tu je třeba využít k tomu, aby jim vytvořené řešení opravdu sloužilo a usnadňovalo jejich práci. Toto vyžaduje od programátorů seriózní přístup, jistou dávku diplomacie a v neposlední řadě i pevné nervy.

Tento referát není návodem, jak přesně postupovat. Je spíše shrnutím několika málo získaných zkušeností a částí některých témat, o kterých by se jistě dalo dlouze diskutovat.