

Ing. Alexandre Winkler

OKR Automatizace řízení, k.ú.o. Ostrava

## OBEČNÉ ZKUŠENOSTI Z PROGRAMOVÁNÍ ON-LINE TELEPROCESSINGOVÝCH SYSTÉMŮ

Zkušenosti, které jsou diskutovány v příspěvku, byly získány kolektivem pracovníků OKR AŘ-k.ú.o. při tvorbě dispečerského on-line systému řízení výroby Generálního ředitelství OKD. Jedná se o zobecnění zkušeností ověřených více než ročním provozem výše uvedené aplikace. Úspěšnost dispečerského systému byla umožněna zejména velkou invencí svých kolegů programátorů, kteří se na jeho tvorbě podíleli a svými nápady inspirovali systémovou analýzu celého projektu.

Nejprve považuji za vhodné uvést podmínky a prostředky, v nichž systém vznikl a v nichž je provozován:

- Projekce i programování tohoto systému byly zpracovány modulárně s využitím dokumentační techniky HIPO (Hierarchy-Input-Process-Output).
- Hostitelským počítačem je IBM 370/145 s centrální pamětí 512 kbyte.
- Operační systém je OS-VS1.
- Podpůrný softwarový aparát je systém STELA (systém teleprocessingových aplikací) vyvinutý na VSŽ Košice.
- Programovací jazykem byl zvolen ASSEMBLER.

- Terminálová síť pracuje v local-mode a obsahuje čtyři displejové terminály a jednu mozaikovou tiskárnu IBM.
- Systém pracuje nepřetržitě 24 hodin denně s výjimkou volných sobot a nedělí.

Z výše uvedeného vyplývá, že pro budování teleprocessingových systémů jsou nutné určité softwarové i hardwarové předpoklady, bez nichž nelze tyto systémy realizovat. Jestliže jsou všechny tyto vnější předpoklady dodrženy, nezabezpečuje to však ještě úspěch uživatelských aplikací. Přechod k teleprocessingovým systémům totiž znamená obrovský průlom do vžitých norem a způsobů uvažování analytiků i programátorů. Výpočetní technika je zde distribuována formou terminálů mimo výpočetní středisko k uživatelům, kteří s ní nejsou profesně spjati a mnohdy trpí averzí k počítači nebo v lepším případě jakýmsi ostychem.

Na zpracovávaná data číhají různá nebezpečí ve formě nekvalifikovaného, resp. zlomyslného uživatele nebo výpadku systému. Pod výpadkem systému zde rozumíme buď výpadek podpůrného software reprezentovaný zpravidla rozpadem terminálové sítě nebo výpadek hostitelského výpočetního systému, např. porucha počítače. Je tedy zapotřebí si uvědomit, že je podstatný rozdíl např. mezi abnormálním ukončením práce a jeho ošetřením v dávkovém prostředí a v prostředí teleprocessingu. V dávkovém režimu se jedná téměř vždy pouze o poškození relativně izolované uživatelské aplikace. V teleprocessingu může takovéto ukončení poškodit všechny ostatní uživatele sítě.

Cílem příspěvku je tedy podat informace zejména těm, kteří s budováním obdobných systémů začínají a umožnit jim vyvarovat se chyb, kterých jsme se my sami dopustili.

Teleprocessingové, dále jen TC, systémy můžeme rozdělit do dvou velkých skupin:

- Systémy pracující v rámci terminálové sítě, kde jednotlivé uživatelské programy, dále jen transakce, jsou nezávislé podúlohy (subtásky) pracující v rámci jedné oblasti paměti (partition).
- Systémy pracující bez sítě, kdy každá uživatelská transakce vyžaduje vlastní oblast paměti a je využíváno např. přístupové metody BTAM (basic telecommunications access method), zpravidla pouze na úrovni READ-WRITE.

V dalším se budeme zabývat první skupinou, která je vývojově vyšším stupněm, ale vyžaduje podpůrný teleprocessingově orientovaný software.

Podpůrný teleprocessingově orientovaný software, dále jen PTS, musí zabezpečovat dvě základní oblasti:

- ovládání samotného PTS na úrovni jeho komunikace s operačním systémem a operace nad terminálovou sítí,
- práce nad uživatelskou transakcí a její podpora při komunikaci s terminály, přístupu k datům na externích paměťových médiích a možnost přechodu na dávkový způsob zpracování v případě, že některý ze zdrojů nárokovaných uživatelskou transakcí není momentálně k dispozici.

První oblast zabezpečuje při detailnějším strukturování tyto funkce:

- spouštění (startování) teleprocessingové sítě (teplý, studený start), startování linek, pracovních oblastí subtásků (regionů), přidělení priorit atd.,
- adresování a komunikace s jednotlivými terminály,
- ošetřování pozastavených (stopnutých) linek a stopnutých transakcí,
- poskytování informací o stavu sítě, aktivních subtáscích,
- testování linek a terminálů,
- testování konverzačních kroků uživatelských transakcí formou snímků (SNAP-ů),

- startování uživatelských transakcí.

Tyto funkce jsou aktivovány pomocí operátorských povelů PTS vysílaných ze systémové konzoly nebo tzv. MASTER-terminálů (vedoucí nebo hlavní terminál), na který může být distribuována část pravomocí systémové konzoly po nastartování TC sítě.

Druhá oblast se rozpadá do tří podoblastí, z nichž každá plní určitý souhrn funkcí při podpoře uživatelských transakcí a je schopna pracovat i samostatně.

Oblast TC zabezpečuje obousměrnou komunikaci uživatelské transakce a terminály. Jedná-li se o transakci jednoduchou, pak se tato transakce po skončení práce sama vymaže. Jedná-li se o transakci konverzační, je pro ni budován odkládací zásobník, v němž je vždy uložen poslední krok konverzace. Oblast TC tedy vytváří podporu uživatelské transakci směrem k terminálům (displeje, tiskárna, dálnopis).

Oblast DB (přístup k datům na externích médiích) zabezpečuje tyto funkce:

- řízení přístupu k datům na externích paměťových médiích,
- zabezpečení integrity a bezpečnosti dat,
- podpora při budování složitějších organizací dat podle potřeb uživatele,
- umožnění práce v dávkovém i on-line prostředí.

Práce DB probíhá ve třech fázích:

1. Program vyvolá DB a předá řízení
2. DB - splní požadavek a předá řízení zpět programu
3. Program pokračuje dále do doby, kdy opět potřebuje data (celý cyklus se opakuje).

V případě paralelní práce více transakcí nad společnými daty, jsou jejich požadavky zpracovávány postupně.

Poslední oblastí je zabezpečení přechodu na dávkové zpracování v případě, že některý ze zdrojů požadovaných

transakcí není k dispozici, předávání dat mezi transakcemi, sběr dat a výstupy typu SYSOUT (systémový výstup). Pro tyto účely slouží soubor, který nazýváme SPOOL. Je obdobou SPOOL-u operačního systému.

Z výše uvedeného je zřejmé, že PTS je složitý softwarový produkt, který je lépe převzít než budovat svépomocně, protože nároky na jeho vytvoření jsou několik člověkoroků. Spolehlivě pracující PTS je výchozím předpokladem, ale nezabezpečuje sám o sobě kvalitní chod uživatelských transakcí tak, jako operační systém nezabezpečuje bezchybnost špatně napsaných programů. Dále se již budeme zabývat problematikou programování konverzačních transakcí v on-line prostředí.

První částí problematiky, odlišnou od běžného dávkového zpracování, je přímý kontakt s počítačem prostřednictvím terminálu. (V našich dalších úvahách se budeme zabývat pouze terminály, s nimiž máme praktické zkušenosti, tj. alfanumerické displeje a mozaikové tiskárny IBM.) Terminál předává do počítače určité kombinace čísel a znakových řetězců a první úkol, který při této komunikaci vyvstává je, aby data z terminálu byla kontrolována na úrovni numeričnosti či ne-numeričnosti. Zde je třeba zabezpečit, aby z pole, do něhož mohou být vyplňována pouze čísla, popř. desetinná tečka a znaménka, nebylo možno přenést jiné znaky, popř. písmena.

Formát dat, který je přenášen z obrazovky, je možno budovat externě ve formě hotových podprogramů v load-formě uložených na knihovně formátů nebo budovat formát v závislosti na momentální situaci přímo v programu. Při programování transakcí je zapotřebí volit mezi těmito dvěma možnostmi. Obecně lze doporučit pevný formát tam, kde předpokládáme časté modifikace formátu ze strany uživatele. Pevný formát totiž lze velmi dobře parametrizovat a pomocí obslužného programu generovat zdrojový text formátového podprogramu. Pro

různé speciální funkce, zprávy typu upozornění či instrukcí uživateli, nebo formát, který je závislý např. na struktuře dat jím prezentovaných nebo vstupujících, je vhodnější formát budovat přímo v programu.

Každý formát by měl v sobě obsahovat verbální instrukce jak z něj přejít na předchozí nebo následující formát, pokud to dialogový způsob práce vyžaduje.

Velmi důležité je při komunikaci "terminál - počítač" zabezpečit nějakým způsobem informaci, zda byl přenos zejména směrem od terminálu k počítači úspěšný. Tuto informaci lze zabezpečit zprávou nebo např. odskočením vyplněných políček v rámci pole na opačnou stranu, než ze které bylo vyplňováno atd.

Při směru zpráv od počítače k terminálu je vhodné zabezpečit při přetečení číselných výsledků přes rozsah výstupního pole alespoň částečnou informaci např. automatickým přechodem na exponenciální tvar tak, aby nedošlo k úplné ztrátě informace. Mozaikovou tiskárnu používat v závislosti na její rychlosti, tedy nesnažit se pomocí ní nahradit rychlotiskárnu počítače.

Ovládání TC aplikací je vhodnější pomocí funkčních kláves kombinovaných s nabídkovými panely typu "jídelníček", než-li pomocí klíčových slov, které je vždy zapotřebí znovu vypisovat a která kladou nároky na paměť uživatele.

Posledním z problémů při komunikaci "terminál - počítač" je rychlost odezvy. Zde je důležité správně rozhodnout, co držet ve vnitřní paměti a co na externích paměťových médiích.

Na základě uvedených problémových okruhů komunikace "terminál - počítač" se lze zabývat dekompozicí TC-online systémů. Z této dekompozice lze pak dále odvodit oblasti vhodné pro parametrizaci těchto aplikací. Niže uvedená dekompozice není jediná, ale z našich zkušeností se jeví jako

účelná a ověřená praxí.

Nyní si popíšeme jednotlivé moduly, na které se TC-online systém rozpadá.

Startovací modul (prolog) transakce: v tomto modulu je zapotřebí zabezpečit natažení dat do pracovních oblastí ve vnitřní paměti (např. hodnoty, které jsou progreai za určitý časový úsek). Vyvolání úvodního nabídkového panelu. Kontrola správnosti parametrů, s nimiž je transakce startována. Kontrola hesla, je-li transakce chráněna heslem. Kontrola oprávněnosti přístupu k datům na základě tvaru sítě (např. první linka, první terminál je oprávněn číst i zapisovat, ostatní linky a terminály smějí pouze číst). Startovací modul je tedy přípravným úvodním modulem, v němž se konverzace zpravidla omezuje pouze na kontrolu oprávněnosti úvodní nabídky, popř. zprávu o tom, jak prolog proběhl. Obsazování pracovních oblastí není v tomto modulu nezbytné a lze jej někdy realizovat jako úvodní práce následujících komunikačních modulů.

Komunikační modul terminálů (paměť-displej, displej-paměť): v tomto modulu jsou zabezpečeny obousměrné přenosy z terminálu do vnitřní paměti počítače a přenosy zpět z paměti. Do paměti jdou data naplňovaná přes displej, zpět jdou kromě těchto dat i data zpracovaná pomocí zabudovaných algoritmů. V tomto modulu je kromě přenosů zabezpečována také adresnost dat a jejich kontrola, tj. plnění výpočtových algoritmických modulů příslušnými kontrolovanými daty a aktualizace těchto dat v případě změny. Hlavním kritériem je u komunikačního modulu rychlost odezvy a správnost poskytovaných výsledků.

Komunikační modul dat (vnitřní paměť - externí paměťová média a zpět): tento modul má dvojitou funkci. V případě práce ve vnitřní paměti kopíruje při každé aktualizaci celou paměť-

čovou oblast na externí paměťový soubor s dispozicí KEEP. Tento soubor slouží při výpadech výpočetního systému pro zabezpečení dat, která byla z terminálu plněna. Druhou funkcí tohoto modulu je přímá spolupráce s externími paměťovými médii, např. při sběrech dat nebo opravách dat na těchto souborech uložených.

Tiskový modul pro práci s mozaikovou tiskárnou: tiskový modul zabezpečuje prezentaci dat na mozaikové tiskárně a to nejen ve formě hardcopy, kdy se na tiskárně kopíruje obsah obrazovky, ale také pro tisky informací, které nemusí být současně vůbec zobrazeny na displeji. S ohledem na poměrně malou rychlost této tiskárny, je zapotřebí tisky omezit na skutečně potřebný rozsah a nepoužívat tyto tiskárny pro rozsáhlé tisky. V případě, že se netiskne celých 120 znaků, dosáhneme určitého zrychlení, nedojíždí-li vozík tiskárny vždy až na celý rozsah, ale tiskne-li se pouze aktuální délka řetězce a po ní je návrat na začátek.

Uzavírací modul (epilog): v tomto modulu provádíme veškeré akce spojené s ukončením práce transakce. Jedná se o uzavření souborů, přesun požadovaných dat do bezpečnostních souborů a archivních souborů. Vlastní ukončení transakce. Tento modul je vhodné také chránit heslem, popř. dvou-  
stupňovou varovací funkcí, protože se zpravidla jedná o akce, které se obtížně rekonstruují. Pod dvoustupňovou varovací funkcí rozumíme to, že po vyvolání určité akce se uživatele formou zprávy na terminálu zeptáme, zda opravdu na požadované akci trvá, teprve při opětovném potvrzení požadovanou akci provedeme.

Po této dekompozici lze vytípotvat oblasti vhodné pro parametrizaci:

- startovací modul (prolog): parametricky zabezpečíme přenos z externích paměťových médií do vnitřní paměti s tím, že



je možno jen pomocí parametrů měnit strukturu toho, co má být do vnitřní paměti posláno. Další oblastí je snadná změna hesla pro startování transakce s ohledem na to, že pokud má být heslo účinné, je vhodné jej měnit alespoň 1x měsíčně.

- Komunikační modul terminálů (paměť-displej, displej-paměť): v tomto modulu, pokud se má pružně přizpůsobovat požadavku uživatele, je zapotřebí parametrizovat vazbu mezi formátem obrazovky a vnitřní pamětí v obou směrech. Zde pomocí parametrů popisujeme tuto relaci: ze kterého místa obrazovky se má přenést řetězec o určité délce do určitého místa ve vnitřní paměti a zpětně co z vnitřní paměti má být přeneseno zpět na obrazovku ve formě nezpracované a co je zapotřebí zpracovat určitým algoritmem před zpětným přenosem na obrazovku. Popis parametrů pro tyto přenosy je nejob-  
sáhlejší částí parametrizace systému, ale po počáteční investované práci umožňuje v podstatě ze dne na den akceptovat změny požadované uživatelem.
- Komunikační modul dat (vnitřní paměť - externí paměťová média a zpět): oblast bezpečnostních kopií není zapotřebí parametrizovat, poněvadž struktura seznamů ve vnitřní paměti se vhodnou parametrizací předchozího modulu prakticky nemění. V oblasti sběrových, popř. opravných operací je zapotřebí parametricky zabezpečit obousměrný převodník pro přesun určité části vnitřní paměti do určitých položek v rámci věty na externím paměťovém médiu a opačně. Parametry je vhodné vybavit možností dynamicky volat určité obslužné moduly, které např. na základě určitého čísla zakázky přidělí jiné oblasti na externím paměťovém médiu text apod.
- Tiskový modul pro práci s mozaikovou tiskárnou: zde je vhodné parametricky popsat formát tiskové sestavy, do ně-

hož lze přesunout data pomocí aparátu vybudovaného pro komunikační modul terminálů (paměť-displej, displej-paměť).

- Uzavírací modul (epilog): tento modul má obdobné parametry pro spolupráci s externími soubory jako komunikační modul dat (vnitřní paměť - externí paměťová média a zpět). Parametry zde udávají přechod z vnitřní paměti na externí soubory, tedy co je zapotřebí uchovat pro opětovný start transakce a co nikoliv.

Vhodně vypracovanou parametrizací zvýšíme odolnost systému proti změnám a navíc zabezpečíme změny systému pomocí externích parametrů v oblastech, kde změny přímo do programů mohou být těžko a zdlouhavě realizovatelné. Popsané parametry jsou dosti složité a je pro ně zpravidla nutno vybudovat speciální překladač, který indikuje syntaktické a některé obsahové chyby. Vlastní transakce si pak bere již interpretované parametry zpracované překladačem buď z vnitřní paměti nebo externích souborů.

Poslední problémovou oblastí je bezpečnost dat. Bezpečnost dat je zapotřebí zajistit ve vztahu k nekvalifikovanému nebo neoprávněnému zásahu uživatele a ve vztahu k výpadku výpočetního systému, na němž je daná TC aplikace provozována. Při práci uživatele zabezpečujeme data pomocí kontrol na formální správnost (např. numeričnost) a na obsahovou správnost (např. že součet určitých položek nesmí překročit určitou mez). Kontroly na obsahovou správnost jsou dosti náročné a je zapotřebí volit míru důvěry ve správné vyplnění v souladu s potřebou rychlé odezvy. V obsahové správnosti se nám zpravidla nikdy nepodaří systém dokonale zabezpečit. Proti výpadku výpočetního systému lze systém zabezpečit pomocí logové magnetické pásky a nebo cyklického logu na discích. Pro aplikace menšího rozsahu je však tento způsob dosti nákladný a navíc neřeší beze zbytku určité problémy, které vznikají

při přenosu z vyrovnávacích pamětí (bufferů) na externí paměťové médium. Jestliže transakce již předala data do vyrovnávacích pamětí a k výpadku počítače došlo při přenosu z vyrovnávacích pamětí na externí paměťové médium, pak se data ztratí, aniž to uživatel bezprostředně pozná. Zde je nezbytné data zabezpečit pomocí kontrolních byte zabudovaných v úvodní větě souborů na externích paměťových médiích. Zabezpečování obsahu vnitřní paměti bylo již popsáno výše.

V TC systémech je zapotřebí zabezpečení dat věnovat značnou pozornost, protože výpadek systému zde může snadno zničit několikahodinovou nebo několikadenní práci uživatele bez možnosti zpětné rekonstrukce.

Na závěr je zapotřebí si uvědomit, že oblast TC aplikací je velmi široká a prochází v současné době bouřlivým rozvojem. Rozsah příspěvku neumožňuje se detailně zabývat všemi oblastmi z TC aplikací. Lze však obecně říci, že se jedná o aplikace poměrně náročné na programování a vyžadující poměrně zkušené programátory, kteří ještě v době programování sami ošetří různé nedefinované stavy, které není možno ani v systémové analýze problému zachytit. Hlavní těžiště práce by mělo spočívat v důmyslné parametrizaci systému, zabezpečení zpracovávaných dat proti poškození nebo ztrátě a zajištění rychlé odezvy systému.

- LITERATURA:
1. Manuály systému STELA (kol. autorů VSŽ Košice)
  2. Specifikace programu pro "dispečerský on-line systém řízení výroby GR OKO" (OKR AR-k.ú.o., Jemelka, Kuča, Winkler)
  3. Firemní manuály IDM