STROMOVĚ ORGANIZOVANÁ STRUKTURA - OBECNÁ TECHNOLOGIE ULOŽENÍ A PŘÍSTUPU K DATŮM V COBOLU POD OPERAČNÍM SYSTÉMEM CP/M

RNDr. Vít Lokoč, Miloš Krejzlík, prom. mat.
Dopravní stavby n.p. Olomouc

Mikropočítač TNS, na jehož vzniku a dalším vývoji se nemalou měrou podílejí pracovníci Dopravních staveb Olomouc, se stal díky tisícovým ročním dodávkám z JZD Slušovice na našem trhu pojmem. Chtěli bychom tímto příspěvkem pomoci k rozšíření softwarových možností mikropočítačů pracujících na bázi mikroprocesoru Z80 pod operačním systémem CP/M (MIKROS), zároveň však může být příspěvek chápán jako obecná metoda, kterou lze aplikovat i na jiných typech počítačů.

1. Úvod

Operační systém CP/M je dodáván s překladači problémově orientovaných jazyků - COBOL, FOKTRAN, BASIC, PASCAL a dalšími jazyky. Byli jsme postaveni před problém vybrat z této škály "nejvhodnější". Rozhodli jsme se pro COBOL z těchto hlavních důvodů:

- určení pro hromadné zpracování dat
- programátoři pracují na centrálním počítači EC 1033 rovněž v COBOLu, z čehož plyne jednoduchá zaškolitelnost a možnost přenesení programových standardů do mikropočítačového prostředí
- silná systémová podpora pro práci s displejem

COBOL a obdobně i jiné jazyky jsou vybaveny "klasickými" přístupovými metodami:

- sekvenční (u větších souborů neúměrně dlouhý přístup, nemožnost vložení nové věty)
- relativní (přímá, jednoúrovňová metoda s pevnou délkou věty bez uživatelského klíče)
- indexová (přímá, jednoúrovňová metoda s pevnou délkou věty s uživatelským klíčem, při větších počtech vět časová náročnost při operacích vyhledávání a změnách)

Uvedené přístupové metody jsou vhodné pro jednoduché datové modely. Pro složitější struktury, které jsou při praktickém řešení úloh mnohem častější, jsou limitujícím prvkem a znamenají ve svém důsledku podstatné zvýšení časových a paměťových nároků.

Rozhodli jsme se vytvořit přístupovou metodu, která odpovídá stromové logice úloh, jako víceúrovňovou přímou přístupovou metodu s proměnnými délkami vět.

2. Užití metodiky stromových struktur na modelování typového příkladu

Jako typový příklad užijeme číselník pracovníků s kmenovými údaji a informacemi o složkách mzdy. Číselník obsahuje 4 typy vět:

1...úvodní věta obsahující datum poslední změny číselníku, v souboru se vyskytuje pouze jednou (D)

2...evidenční číslo pracovníka, počet výskytů je roven počtu pracovníků (P_i)

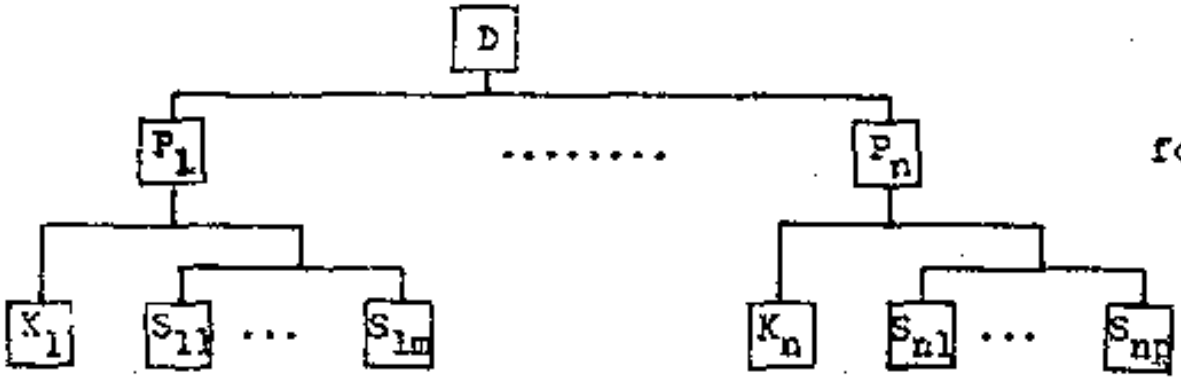
3...kmenové údaje o pracovníkovi, ke každému evidenčnímu číslu právě jeden výskyt (K_i)

4...složky mzdy, ke každému evidenčnímu číslu přísluší vektor složek mzdy (S_{ij})

Číselník můžeme graficky znázornit buď pomocí lineárního zápisu

1 - 2 - 3 - 4 -4 - 4 - ... - 2 - 3 - 4 - 4 - ... forma /1/

nebo užitím stromové metodiky



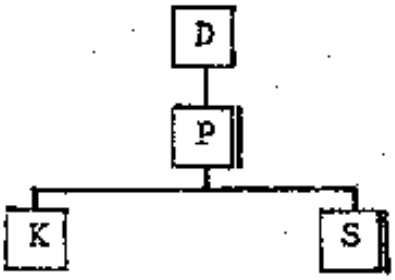
forma /2/

Grafický model typového číselníku můžeme s výhodou zjednodušit zavedením následující konvence značení uzlů stromu:

 ... jednoduchý uzel, tj. uzel s pouze jediným výskytem pod nadřízeným uzlem

 ... násobný uzel, tj. uzel s vícenásobným výskytem pod nadřízeným uzlem

Zjednodušený grafický model má potom tvar



forma /3/

Definujme některé další pojmy, které budeme v dalším textu používat:

- vrcholový uzel... stojí na vrcholu stromu, není podřízen žádnému uzlu, může být pouze jeden
- stupeň větvení uzlu... počet typů podřízených uzlů
- kořen stromu... uzel, který nemá žádné podřízené uzly, tj. s nulovým stupněm větvení
- sousední uzly... výskyty uzlů v detailním stromu formy /2/, které jsou přímo podřízeny témuž nadřízenému uzlu, např. P_1 až P_n, S_{11} až S_{1m}

3. Kumulace uzlů do logických vět

Tak, jako jsme sdružili symbolickým způsobem sousední uzly grafu /2/ do grafu /3/, je možno stejným způsobem vytvářet logické věty (dále jen věty) jako vektor sousedních uzlů, které budeme v dalším nazývat pole. Podle počtu polí pak budeme rozlišovat jednoduché a násobné věty. Každé pole se skládá z datové části, tvořené obsahem příslušného výskytu uzlu a směrníkové části, tvořené vektorem diskových adres podřízených vět. Počet adres se rovná stupni větvení daného typu věty, tj. pro kořen stromu je směrníková část prázdná.

Utvořme si strukturu vět našeho typového příkladu. Bude-li mít pracovník JOSEF NOVAK osobní číslo ØØ2243 a složky mzdy 12, 14 a 22, budou věty vypadat takto:

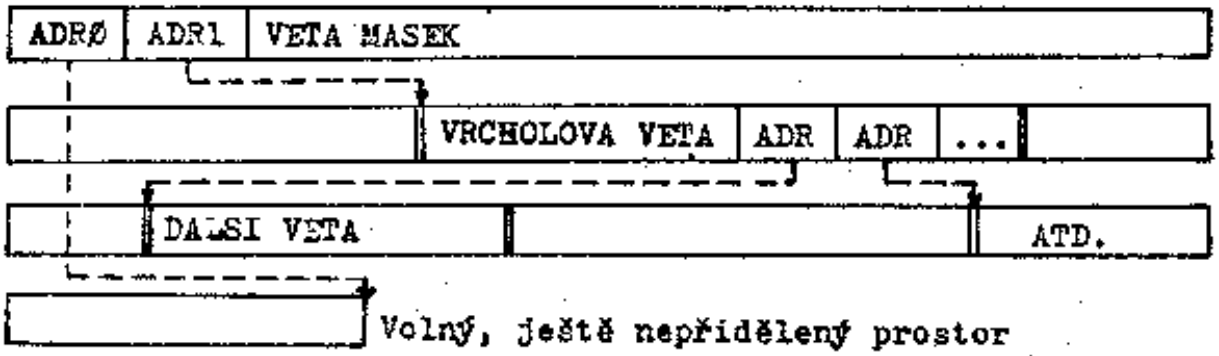
věta 1 - datum posl. aktualizace	86Ø611	AØ				
věta 2 - osobní číslo pracovníka	ØØ2243	A1	A2	.....		
věta 3 - kmenové informace	JOSEF NOVAK, bydliště atd.					
věta 4 - složky mzdy	12	36ØØ	14	2ØØ	22	5ØØ

kde AØ, A1, A2 jsou diskové adresy výše uvedených výskytů uzlů typu 2, 3, 4. Věty typu 1 a 2 se u daného příkladu vyskytují pouze jedenkrát, věty typu 3 a 4 mají počet výskytů roven počtu pracovníků. Věta typu 2 je vektor polí.

4. Organizace dat na diskovém médiu

Počínaje prvním bytem přiděleného diskového prostoru mají data následující strukturu:

forma /4/



kde:

- ADRØ je adresa prvního volného bytu v souboru, od kterého je možno zakládat nová data na disk. Tato adresa se zvětšováním použitého diskového prostoru odpovídajícím způsobem aktualizuje.
- ADR1 je adresa vrcholové věty stromu, je právě jedna.
- věta masek je speciálně vytvořená násobná věta, sloužící k řízení přístupů k datům. Počet polí této věty je roven počtu typů vět souboru. Pojem maska typu věty bude objasněn v dalším textu.

Věty jsou zakládány na disk od adresy ADRØ. V případě násobných vět je možné při zvětšení věty provádět její segmentování tak, že přetokové oblasti jsou zakládány na konec souboru od adresy ADRØ. Hloubka segmentování vět není omezena.

Věty případně jejich segmenty jsou na diskovém prostoru uloženy podle toho, jak vznikaly. Vypiňují spojitě podle směrníků prostor od 1. bytu souboru až po volný prostor určený adresou ADRØ. Dojde-li ke zrušení věty případně jejího segmentu, je zrušen směrník, který na tuto větu ukazoval, čímž se o tomto prostoru ztratí informace. Vznikne tzv. "černá díra".

Jak už bylo řečeno, ke každému typu věty souboru přísluší deklarativní maska, obsahující všechny nutné informace o daném typu věty:
- délka datové části pole věty
- délka případného klíče, umístěného na začátku pole
- velikost rezervního prostoru prvního segmentu násobné věty sloužící k tomu, aby při případném zvětšování věty nedocházelo ihned k vytváření přetokových oblastí
- velikost rezervního prostoru dalších segmentů násobné věty
- stupeň větvení daného typu věty
- příznak, jde-li o jednoduchou nebo násobnou větu

Vlastní struktuře datové části pole není přístupovou metodou nijak sledována. Její deklarace je obsažena v hlavním cobolovském programu.

5. Organizace vnitřní paměti

Program využívající popisovanou přístupovou metodu vyžaduje přidělení spojité pracovní oblasti ve vnitřní paměti. Jelikož je možno zpracovávat nezávisle na sobě několik souborů, je podle potřeby pracovní oblast programu rozčleněna na spojité pracovní oblasti souborů s následující strukturou:

VĚTA MASEK	VRCHOLOVÁ VĚTA	...	ZPRACOVÁVANÁ VĚTA	NEVYUŽITÝ PROSTOR

Platí, že
- daný výskyt věty je vždy načten celý, tj. je spojitě složen ze všech případných segmentů do jednoho celku
- pracujeme-li s některým výskytem věty, je tato věta načtená v pracovní oblasti souboru spolu se všemi výskyty nadřízených vět, tj. vrcholová věta je načtena vždy

6. Funkce přístupové metody

Funkce můžeme rozdělit do několika skupin:

a) Operace na úrovni souboru

- kopírování
  umožňuje provést záložní kopii souboru nebo obnovu při jeho poškození
- reorganizace
  odstraní "černé díry" po přetečených nebo zrušených větách, spojí segmentované věty do jediné spojité věty, podřídí fyzické uložení vět logickému uložení, tj. logicky sousední věty leží vedle sebe i na disku, čímž se zrychlí sekvenční přístupy, výsledkem je zmenšený a pro přístupy zefektivněný prostor
- otevření existujícího souboru na disku
  provede se přiřazení pracovní oblasti souboru ve vnitřní paměti, fyzické otevření na disku, načtení věty masek, načtení vrcholové věty na začátek pracovní oblasti souboru
- uzavření souboru
  v případě oprav se aktualizuje na disku vrcholová věta případně ADR0 a ADR1, provede se fyzické uzavření souboru na disku

b) Operace na úrovni věty

- pohyb po stromu dolů
  podle zadaného pořadového čísla pole zpracovávané věty a pořadového čísla větve, přes kterou půjdeme po stromu dolů, se vyhledá směrník podřízené věty a ta se načte do pracovní oblasti souboru ihned za opouštěnou nadřízenou větu
- pohyb po stromu nahoru
  v případě oprav opouštěné věty se provede její přepis případně zápis na disk, změnila-li se adresa opouštěné věty, provede se aktualizace směrníku v nadřízené větě
- poskytnutí celé věty z pracovní oblasti souboru do uživatelského programu
- modifikace celé věty v pracovní oblasti souboru hodnotou z uživatelského programu
- vytřídění polí věty v pracovní oblasti souboru dle hodnoty klíče (definovaný maskou věty)

c) Operace na úrovni pole (pouze ve vnitřní paměti)
- poskytnutí případně modifikace datové části pole, zrušení případně uložení pole do věty, všechno na základě pořadového čísla zpracovávaného pole
- vyhledání pole dle klíče
- sekvenční průchody mezi sousedními výskyty vět
- vyhledání mezi sousedními výskyty vět podle klíče nadřízené věty

d) Operace na úrovni adresy (pouze ve vnitřní paměti)
- poskytnutí případně modifikace adresy dle pořadového čísla pole a pořadového čísla zpracovávané větve

Po každé operaci přístupové metody je aktivizována speciální stavová proměnná, jejíž hodnota určuje kód chyby:
- čtení z nesmyslných dat na disku
- čtení z nezaloženého extentu souboru na disku
- pokus o založení již existujícího souboru
- soubor na disku nebyl nalezen
- nelze uzavřít extent souboru
- chybná operace CLOSE
- přeplněný adresář disku
- přeplněný diskový prostor
- chybná délka dat při čtení nebo zápisu
- chybně zadaný vstupní parametr
- přetečena vnitřní pole
- pole zadaného klíče nebylo při vyhledávacích rutinách nalezeno
- první resp. poslední pole při sekvenčních průchodech
- soubor nebyl po posledních opravách regulérně uzavřen

Z předchozího plyne, že žádnou z deklarovaných funkcí nelze stromově organizovaný soubor založit. Jeho zakládání provádí speciální program "STBCM.CCM", který na základě vyžádaných informací o druzích typů vět založí na disku "prázdný" stromový soubor obsahující pouze větu masek a příslušné údaje.

7. Zkušenosti z aplikací stromových struktur

Výše popsaná přístupová metoda byla naprogramovaná v jazyku COBOL 80 pod systémem CP/M (64kB) a je realizována pomocí modulu "CGSOS.REL", který s hlavním cobolským programem komunikuje přes parametry smluvené struktury. K provádění vlastních diskových a paměťových operací byl vyvinut silný assemblerovský podpůrný aparát, sestávající se z těchto nosných modulů:

- CGDAT.REL .... modul pro práci s diskem, umožňuje založit, otevřít resp. uzavřít soubor, přečíst resp. zapsat řetězec znaků definované délky dle přímé diskové adresy
- CGMOVE.REL ... přesuny znakových řetězců libovolné délky kdekoliv ve vnitřní paměti
- CGFREE.REL ... poskytnutí informací o volné kapacitě vnitřní paměti
- CGFIND.REL ... vyhledání pole dle zadaného klíče v obecné tabulce umístěné libovolně ve vnitřní paměti
- CGSRT3.REL ... vytřídění obecné tabulky umístěné libovolně ve vnitřní paměti
- CGPACK.REL ... oboustranná konverze mezi zónovou a sbalenou formou numerických řetězců libovolné délky ve vnitřní paměti (poloviční nároky na kapacity vnějších médií)

Díky uvedeným modulům jsou maximálně urychleny a zefektivněny procesy přístupů k stromově organizovaným souborům.

Na obecné úrovni je vyřešena oboustranná konverze stromově organizovaných souborů typu /4/ mezi mikropočítačovým prostředím a sekvenční formou typu /1/, zpracovávanou na centrálním počítači EC 1033:

a) konverze ze stromově organizovaného souboru do sekvenčního souboru s proměnnou délkou věty na EC 1033

- soubor se vyhraje ve fyzických 2kB blocích z disku/diskety na magnetickou pásku
- tetno soubor na magnetické pásce se na EC 1033 zpracuje speciálním konverzním programem, který na základě načtených masek

provede vlastní transformaci do sekvenční formy s proměnnou délkou věty smluvené jednotné struktury
- takto vytvořený sekvenční soubor vstoupí do následného zpracování, ve kterém je nutno dle struktur vět zabezpečit:
  . transformaci ASCII do EBCDIC znakových řetězců (binární a pakované čísla mají stejné zobrazení, nemusí se tudíž transformovat)
  . rozklad numerických řetězců vytvořených modulem CGPACK.REL ze sbalené do zónové formy

b) konverze ze sekvenčního souboru smluvené struktury na EC 1033 do stromově organizovaného souboru na disku/disketě

- na EC 1033 vytvořit soubor s proměnnou délkou věty smluvené struktury s ASCII řetězci a sbalenými numerickými řetězci čitelnými modulem CGPACK.REL
- speciálním inverzním programem vytvořit dle zadaných masek na magnetickou pásku obraz zadaného souboru se stromovou organizací
- soubor založit po fyzických 2kB blocích z magnetické pásky na disk/disketu

Přesunem uvedených konverzí z mikropočítačového prostředí do prostředí velkého počítače došlo k řádovému urychlení a zjednodušení manipulací na mikropočítači. Nelze opomenout ani tu skutečnost, že obecná úroveň konverze šetří práci problémově orientovaných programátorů na EC 1033.

Nevýhody stromově organizovaných souborů je možno shrnout v těchto bodech:

- zvýšení nároků na vnitřní paměť, kterou zaujímá zkompilovaný program (CCSOS.REL se všemi návaznými programy zabírá asi 10kB vnitřní paměti)
- zpomalení sekvenčních průchodů oproti ostatním přístupovým metodám
- po zrušených větách resp. jejích segmentech zůstávají již nepřístupné "černé díry"
- součet délek všech výskytů vět v libovolné spojnici "kořen - vrchol" stromu nesmí překročit vymezenou pracovní oblast vnitřní paměti (maximálně 12 až 15 kB)

Oproti nim stojí tyto výhody:

- přímé přístupy se provádějí do 2 sekund dle složitosti stromu
- variabilita stromové struktury - počet uzlů není omezen
- není omezen počet současně zpracovávaných souborů
- pracovní oblast programu je uložena za programem, což vede k plnému vytížení paměti dostupné uživatelskému programu (55kB), nedochází také ke zvětšování COM - verze programu o pracovní oblasti uvnitř programu
- efektivní využití diskového prostoru
- díky opožděným zápisům a podmíněnému čtení minimalizace periferních diskových operací
- kopírování realizováno na fyzické úrovni, což vede k maximální rychlosti úklidů resp. obnovování souborů (časy srovnatelné s programem PIP.COM)

8. Závěr

Popsaná přístupová metoda stromově organizovaných struktur se stala závazným programovým standartem u vhodných typů úloh pro programátory Dopravních staveb Olomouc. Na jejím základě je řešena řada úloh na mikropočítači TNS. Typické je užití u různých druhů číselníků resp. datových souborů, které mají na číselníky přímou vazbu a je žádoucí, aby tato vazba byla realizována interaktivně v reálném čase.

Několik let se zabýváme problémem přípravy a předzpracování dat, jehož výsledkem je vlastní multiklávesový operační systém TNS - DATEL1, jehož výstup je realizován do magnetické pásky. Metoda stromové organizace souborů se stala základem při organizování struktur dat na magnetickém disku nového systému TNS - DATEL2, který v současnosti vyvíjíme.

Závěrem lze konstatovat, že všestranné použití metody stromově organizovaných struktur podstatně svýšilo možnosti zpracování hromadných dat na mikropočítačích TNS.

Literatura:

/1/ Navrátil Z.: OS MIKROS - příručka programátora
KS k. ú. o., 1982

/2/ Rajčan P., p.m.: OS MIKROS - služobné programy, príručka uživatela
KS k. ú. o., 1984

/3/ Mičovič M., RNDr., CSc.: COBOL 80 - príručka programátora
KS k. ú. o., 1982

/4/ Mišovič M., Ing.: COBOL 80 - příručka operátora
KS k. ú. o., 1984