Ing.Martin Zralý
INORGA PRAHA

NÁVRH UŽIVATELSKÉHO JAZYKA VYŠŠÍ ÚROVNĚ (pro) DBS/25

0. Úvod. Charakteristika DBS/25

Předlohou pro DBS/25 byl v jistém smyslu projekt DL/I fy IBM. DBS/25 se od DL/I značně liší, i když zachovává do značné míry stejné vnější interface (tvar příkazů pro popis databáze i tvar příkazů pro práci s daty, hostitelský jazyk assembler nebo vyšší programovací jazyky).

Protože principy systému DL/I, respektive hierarchického datového modelu, se kterým pracuje jak DL/I, tak DBS/25, jsou již programátorské veřejnosti poměrně známy, upozornil bych stručně jen na některé odlišnosti.

—

DBS/25 je koncipován jako součást operačního systému DOS-3, což by mělo zaručit vyšší účinnost výpočetního systému v multiprogramním režimu.

Integrace DBS/25 do operačního systému má na jedné straně za následek nepřenositelnost DBS/25 pod jiný operační systém. Na druhé straně dovoluje využít nejrychlejších operací vstupu a výstupu, které jsou v DOS-3 k dispozici. DBS-25 nepoužívá pro své I/O operace žádné z tradičních metod (SAM, DAM, ISAM), ani rutin pro fyzický přístup, ale

využívá stránkovacího mechanismu virtuálního systému. Příslušná aktivní část databáze (tj. část s daty, se kterými pracujeme) se stane privátním odkládacím souborem (page data set). Tento úkon se projeví pouze úpravou systémových tabulek, aniž je třeba provádět I/O operace. Tak se zpracovávaná část databáze vlastně stane součástí virtuální paměti uživatele. I/O operace se provádí až v okamžiku odkazu na nějakou virtuální adresu směřující do těchto dat. Operace proběhne v optimálním režimu v rámci strategie používané pro stránkování.

Moduly DBS/25 jsou součástí sdílené systémové fáze a aplikační programy pracují s databázemi obdobně jako s jinými prostředky LIOCS. Program může otevřít kteroukoliv databázi během své činnosti (OPEN), provádět manipulace s daty (CALL), databázi uzavřít (CLOSE). Tento způsob práce minimalizuje nároky na počet diskových jednotek.

Důležitým rysem DBS/25 je sdílení výkonných modulů DBS/25 a sdílení dat. Obojí snižuje požadavky jak na reálnou paměť, tak na stránkování. Sdílení výkonnových modulů je umožněno umístěním reentrantních výkonných modulů DBS/25 do sdílené části virtuálního prostoru. Znamená, že v reálné paměti je v každém okamžiku přítomna toliko jedna "kopie" modulů DBS/25, kterou mohou využívat všechny současně aktivní aplikační programy.

Sdílení dat je chápáno ve smyslu sdílení dat v reálné paměti. Jediná "kopie" těchto dat je schopna obsloužit řadu aplikačních programů. Výhoda spočívá i v tom, že je zaručena práce se skutečně aktuálními daty.

Tato úvodní informace o DBS/25 byla do příspěvku zařazena na přání redakční rady; dále bych se chtěl věnovat interpretaci návrhu jednoho z prostředků, který podle našeho názoru může významně přispět k využitelnosti DBS/25, totiž uživatelskému jazyku vyšší úrovně (UŽJ/II).

1. Účel UŽJ/II

Uživatelský jazyk (UŽJ/II) je nástroj, který je určen pro aplikační programátory. Má usnadnit zápis operací pro manipulaci s daty v uživatelských programech pro zpracování v dávkovém režimu. Vychází z jazyka 1. úrovně, který je integrální součástí základních modulů DBS/25 a vhodnou agregací jeho elementárních prvků vytváří základní potřebné příkazy pro zajištění datového toku mezi uživatelským programem a bází dat DBS/25. Důvodem pro vytvoření uživatelského jazyka II je snaha omezit nutnost práce se základními moduly DLS/25 při hromadném využívání. Přímá práce s těmito moduly na fyzické úrovni se zdá být v rámci aplikační problematiky neefektivní, zejména z důvodů náročnosti na hloubku znalostí funkčních principů struktury vnitřní výstavby a metodiky ošetřování mimořádných stavů samotných modulů. Další nevýhodou takovéhoto přístupu je příliš těsné propojení aplikačního a systémového software, které omezuje možnost rozvíjení prvků některé z těchto složek promítáním změn, neboť důsledkem je nutnost provádět odpovídající úpravy i druhé složky.

2. Koncepce řešení UŽJ/II

Koncepce řešení uživatelského jazyka II je postavena na využití standardní funkce CALL. Z charakteru funkce CALL plyne jedno zásadní omezení, že totiž v příkazu definované parametry se přenášejí do volaného programu ve formě adres. Tím je striktně determinována forma i variantnost zadání parametrů v příkazu CALL uživatelského jazyka.

Důvody pro toto rozhodnutí byly především:

a) Toto pojetí umožňuje využít uživatelský jazyk II v programech psaných v assembleru i ve vyšších programovacích jazycích (PL/1, COBOL).

b) Lze navrhnout takovou formu jazyka, která si vynutí strukturalizaci uživatelského programu na část vstupní (včetně příkazu CALL UŽJ/II) a části výkonné, psané jako

samostatné moduly (uživatelské exity) v souladu se zásadami strukturovaného programování.

c) Příkazy uživatelského jazyka II lze formulovat tak, že umožní na jeden příkaz provést definovanou operaci s daty mnoha (všech) výskytů určitého typu segmentu nebo závislých skupin segmentů.

d) Syntax uživatelského jazyka II je obdobná syntaxi jazyka základní úrovně, který je také postaven na využití funkce CALL.

3.1. Prvky UŽJ/II

Uživatelský program, který komunikuje s bází dat prostřednictvím UŽJ/II musí obsahovat:

1. Popis vyhledávacího argumentu segmentu (SSA) pro všechny segmenty cesty.
2. Popis komunikačního bloku programu II (PCB/II).
3. Příkazy CALL UŽJ/II.

Pozn.: Aby uživatelský program mohl komunikovat s bází dat, musí být pro něj vytvořen (vygenerován) tzv. popisný blok programu (PSB). PSB obsahuje výčet typů segmentů, se kterými bude program pracovat a způsob, jakým bude data využívat (čtení, aktualizace, rušení, ukládání) a vytváří takto vlastní masku struktury databází. Do PSB patří též komunikační bloky probramu (PCB), jichž je právě tolik, s kolika databázemi uživatelský program pracuje a které slouží zejména pro uložení zprávy o průběhu zpracování.

3.1.1. Vyhledávací argument segmentu

Vyhledávací argument segmentu (SSA) specifikuje typ segmentu (resp. jeho určitý výskyt):

a) pro cílový segment, tj. segment, se kterým si uživatelský program žádá pracovat,

b) pro segmenty cesty, tj. segmenty tvořící hierarchickou cestu k cílovému segmentu.

Je vhodné rozlišit - kvalifikovaný SSA,
- nekvalifikovaný SSA.

a) Kvalifikovaný SSA přímo určuje určitý výskyt, resp. určité výskyty požadovaného typu segmentu splňující zadanou podmínku. Je tvořen těmito údaji:
   - jméno segmentu
   - jméno třídícího pole (pole klíče) nebo jiného pojmenovaného pole segmentu
   - porovnávací operátor ($<$, $>$, =, $\neq$)
   - porovnávací hodnotu.
b) Nekvalifikovaný SSA obsahuje pouze jméno segmentu a určuje tedy pouze určitý typ segmentu.

Z odlišností v definování kvalifikovaného a nekvalifikovaného SSA vyplývá i rozdílnost při použití. Nekvalifikovaný SSA je vhodné použít tenkrát, chci-li pracovat se všemi výskyty daného typu segmentu. Ve všech ostatních případech by měly být používány kvalifikované SSA, a to i v případech, kdy to věcně není nezbytné, neboť jejich použití urychluje zpracování příkazu.

3.1.2. Komunikační blok programu

Komunikační blok programu II (PCB/II) (označení II znamená, že jeho struktura se liší od PCB základních modulů) umožňuje přenést informaci o průběhu manipulace s daty, tj. hlášení o bezvadném průběhu nebo o chybových stavech v modulech uživatelského jazyka nazpět do uživatelského programu.

3.1.3. Příkazy CALL UŽJ/II

Příkazy UŽJ/II mají standardní formu příkazu CALL, přičemž jsou navrženy tak, aby mohly respektovat způsob zápisu odpovídající hostitelskému programovacímu jazyku COBOL, PL/1 nebo Assembler.

Příkaz UŽJ/II se skládá z těchto prvků:

a) symbol CALL
b) druh operace s upřesněním, tj. typ operace (jméno CALLu)
c) jméno PCB/II
d) jméno pracovní oblasti pro data
e) jméno uživatelského exitu
f) jméno SSA cílového segmentu
g) jméno SSA segmentů cesty k cílovému segmentu, resp. segmentů větve či úrovně (viz dále).

K příkazu se vztahuje též specifikace hodnot pro vyhledávací argumenty segmentů, jež jsou uvedeny v CALL UŽJ/II. Pro každý kvalifikovaný vyhledávací argument segmentu je třeba specifikovat hodnoty pro daný příkaz, tedy jméno klíče nebo jiného pojmenovaného pole, porovnávací operátor a porovnávací hodnotu, což může být hodnota klíče nebo pojmenovaného pole, nebo určité mezní hodnota pro vyhledání. Tato specifikace musí předcházet příslušnému příkazu CALL UŽJ/II.

3.2. Popis prvků UŽJ/II

Druh operace

UŽJ/II umožňuje čtyři základní druhy operací s daty. Jejich popis je v tabulce 1.

druh operace označení	popis funkce
VYHLED	vyhledá na paměťovém mediu a přesune do určené pracovní oblasti určené datové segmenty
ZMENA	vyhledá na paměťovém mediu a přesune do určené pracovní oblasti určené datové segmenty a po provedené operaci uloží modifikované segmenty na jejich původní místo

DOPLN	provede dodatečné uložení segmentů (tj. dalších výskytů segmentů definovaného typu) do struktury databáze
ZRUS	zruší určené segmenty

Tabulka 1 : Základní druhy operací UŽJ/II

Upřesnění druhu operace

Základní druhy operací je možno modifikovat pěti způsoby, které blíže specifikují strukturu datových segmentů, na kterých se mají operace provést. Alternativy upřesnění, jak je připouští UŽJ/II, shrnuje tabulka 2.

označení upřesnění	zajišťuje operaci pro:
A	jednotlivý segment
W	všechny výskyty daného typu segmentu
S	všechny výskyty segmentu v rámci dané stromové struktury
V	všechny výskyty segmentů přináležející určité vybrané větvi pod cílovým segmentem; pozn.: jména segmentů větve jsou specifikována za jmény segmentů cesty k cílovému segmentu
U	všechny výskyty definovaných segmentů v úrovni pod cílovým segmentem; pozn.: žádají-li se pouze výskyty určitého typu segmentů, doplní se jejich jména za jména segmentů cesty k cílovému segmentu.

Tabulka 2: Alternativy upřesnění druhu operace v UŽJ/II

Z analýzy uživatelské potřeby zpracování dat vyplynula tabulka povolených upřesnění základních operací (viz tab. 3).

druh operace	označení upřesnění
VYHLED	A, W, S, V, U
ZMENA	A, W
DOPLN	A
ZRUS	A, W

Tabulka 3: Povolená upřesnění základních operací v UŽJ/II

Znak upřesnění operace se píše bez mezery za druh operace. Společně pak označují typ operace (z hlediska programátorského tvoří jméno CALLu).

Jméno komunikačního bloku programu (PCB/II)

Zajišťuje komunikaci mezi aplikačním programem a popisem dat pro zpětné předání informace o průběhu manipulace s daty do aplikačního programu.

Jméno pracovní oblasti

Jméno a velikost pracovní oblasti pro data definuje aplikační programátor podle rozsahu dat datového segmentu (datových segmentů), jež vyžaduje. Je-li jedna společná pracovní oblast pro různé typy segmentů, její velikost musí být rovna velikosti největšího segmentu.

Jméno uživatelského exitu

V uživatelském exitu programátor specifikuje:
- způsob a postup zpracování dat získaných daným CALL UŽJ/II
- případné upřesnění příkazu CALLUŽJ/II ve smyslu:
  - výběru daných segmentů splňujících další dodatečné podmínky, tj. podle výsledků porovnání hodnot určených pojmenovaných polí s definovanou referenční hodnotou

- požadavku na přenos i dat segmentů cesty do pracovní oblasti.

Může se zdát v některých případech zbytečné využívat exitu (zejména pro operace s upřesněním A), ale z důvodu jednotnosti prvku pro všechny příkazy CALL UŽJ/II použití exitu považujeme za účelné. Bude minimálně obsahovat ošetření kódu stavu z PCB/II charakterizujícího průběh manipulace s daty.

Předpokládáme, že v další etapě prací by měly být zpracovány výše uvedené možnosti, jakož i další vybrané funkce, pro které by v uživatelském exitu byly zapsány parametry, s jež by byly z uživatelského exitu volány.

Jméno SSA cílového segmentu

Je to symbolické jméno SSA typu segmentu, s jehož daty si uživatelský program žádá pracovat, případně určující pro specifikaci struktury segmentů pro zpracování (viz upřesnění S, V, U).

Jména SSA segmentů cesty

Slouží k přesnému určení datového segmentu, s jehož daty si uživatelský program žádá pracovat. Jsou vlastně adresou cílového segmentu. Parametr může být buď zcela vynechán v tom případě, že uživatelský program žádá všechny výskyty daného typu segmentu bez ohledu na to, ke kterým vyšším patří. V tomto případě se cesta určí podle jména cílového segmentu a tabulky popisu databáze (DBD), která se vytváří při ukládání databáze. Úplnost specifikace cesty závisí na požadovaném rozsahu výběru. Úplná specifikace cesty je nezbytná v případě, že uživatelský program žádá ať již jeden, nebo všechny výskyty daného cílového segmentu, ale pro pevně definované vyšší. Tohoto parametru se též využívá pro určení segmentů požadované větve nebo pro určení požadovaných segmentů nižší úrovně. Jména těchto

segmentů se uvedou za jmény segmentů cesty, přičemž jejich oddělení je zřejmé dle jména cílového segmentu.

3.3. Syntax UŽJ/II

Shrneme-li hlavní zásady platné pro syntax UŽJ/II, můžeme ve stručnosti formulovat tato pravidla:

- Forma popisu prvků UŽJ/II (SSA, popis PCB/II, příkaz CALL UŽJ/II a jeho specifikace) v uživatelském programu odpovídá pravidlům zápisu v programovacím jazyku, ve kterém je psán uživatelský program;
- Pro příkaz UŽJ/II platí:
  a) je třeba dodržet pořadí parametrů příkazu takto:
     1. typ operace, tvořen druhem operace s upřesněním, povinné
     2. jméno komunikačního bloku programu (PCB/II), povinné
     3. jméno pracovní oblasti, povinné
     4. jméno uživatelského exitu, povinné
     5. jméno SSA cílového segmentu, povinné
     6. jméno SSA segment cesty, nepovinné, ale doporučujeme je používat;
  b) může být použit v libovolném místě programu v závislosti na logice programu
  c) specifikace hodnot pro vyhledávací argumenty segmentu patřící k danému CALL UŽJ/II se musí uvést před příslušným příkazem CALL UŽJ/II
  d) druh operace a upřesnění je možno kombinovat v alternativách uvedených v tabulce 3. Upřesnění se píše bez mezery za druh operace a vytváří dohromady typ operace UŽJ/II.
- Pro popis komunikačního bloku programu platí:
  a) uživatelský programátor si definuje popis PCB v rozsahu údajů, jak je uveden v předchozím
  b) každé použité databázi přísluší jeden PCB a každý PCB musí mít v uživatelském programu svůj popis.

- Pro definování vyhledávacího argumentu segmentu platí: Každý kvalifikovaný SSA musí být v programu zvlášť popsán a označen jménem; toto jméno se použije v příkazu CALL. Pro nekvalifikované SSA někdy postačuje jediná společná definice SSA.

3.4. Způsob komunikace uživatelského programu s DBS/25 při využití UŽJ/II

Zajištění komunikace aplikačního programu a DBS/25 prostřednictvím UŽJ/II je postaveno na těchto zásadách:

1. Aplikační program je hlavní, volající, moduly UŽJ/II jsou podřízené, volané. Jsou ovšem hlavní, volající, vůči uživatelskému exitu a výkonným modulům DBS/25.
2. Popis mechanismu fungování:
   Příkaz CALL operací UŽJ/II v aplikačním programu vyvolá příslušný modul uživatelského jazyka. Ten střídavě vyvolává základní moduly a uživatelský exit tak dlouho, až se provede předepsané operace na všech datech. Následuje návrat do aplikačního programu. Schéma komunikace aplikačního programu s moduly UŽJ/II je na obrázku, kde je též zakresleno výhledové využití standardních zpracovacích podprogramů.

4. Závěr

Předpokládáme, že modulární pojetí koncepce UŽJ/II umožní, aby UŽJ/II byl základem jazyka vyšší úrovně pro komunikační režim. UŽJ/II sám vychází z jazyka základní (první) úrovně pro DBS/25, je jeho nadstavbou a bude využívat jeho výkonné moduly.

Literatura:

V.Horák, H.Petroušková, M.Zralý: Uživatelský jazyk II; INCRGA 3/1977
Kolektiv: Technický projekt DBS/25; INCRGA 12/1977
R.Cher: Databankový systém DBS/25; Sborník ze semináře "Programovací systémy"; Poprad 10/1978