

Dokumentační systém výpočetního střediska.
Kamil Kalina, prom.mat., ing. Zdeněk Dyk
První brněnské strojírna

V řadě výpočetních středisek je nějakým způsobem vyřešena evidence, případně dokumentace softwarových produktů výpočetního střediska s pomocí výpočetní techniky. Obvyklou nevýhodou těchto evidencí, ať už se vedou ručně nebo na počítači, je to, že se někdo musí o takový systém starat, doplňovat údaje o nově vzniklých produktech (entitách typu program, soubor, tisková sestava, job atd.) a hlavně - někdo musí dodávat informace a pořizovat data. Málokdy se podaří udržovat takovou evidenci ve skutečně aktuálním, skutečnosti přesně odpovídajícím stavu.

Náš dokumentační systém se nezabývá plánovanými nebo rozpracovanými úlohami - eviduje pouze ty produkty, které se rutinně využívají. Z toho plyne největší výhoda našeho systému, totiž to, že není potřeba pořizovat žádná další data (s výjimkou názvů subsystémů, skupin úloh a úloh, což je zanedbatelné).

Údaje o uvedených entitách se získávají z knihovny rutinních jobů (RUT.JOBLIB), tedy rozbořením JCL, a dále z archivní knihovny zdrojových textů programů, z níž se přebírají komentáře, povinně uváděné za hlavičkou procedury.

Údaje, získané z obou knihoven, se ukládají do báze dat IDMS. Udržuje se prakticky stále aktuální stav. Dospěli jsme k názoru, že pro naše výpočtové středisko stačí týdenní až měsíční cyklus aktualizace. Datum poslední aktualizace je rovněž uloženo v bázi dat a zobrazuje se ve výstupech na terminálu nebo tiskárně.

Pomocí interaktivních programů lze získat informace o údajích, které jsou v bázi dat uloženy a rovněž o jejich vzájemných vztazích.

Dokumentační systém je užitečný jak pro provoz počítače, tak pro programátory, analytiky a lze ho použít i pro potřeby uživatelů - např. k tisku sestavy se soupisem všech tiskových sestav, které výpočetní středisko produkuje pro zadaného uživatele. Příležitostně se ukáže potřeba vytisknout nějakou novou tiskovou sestavu, což se obvykle řeší tím, že se napíše speciální program.

Programy pro analýzu JCL i pro plnění báze dat jsou psané v jazyku PL/I a použitím jeho rozšíření pro práci se sítěmi v operační paměti - tzv. systému 0, vypracovaného v našem výpočtovém středisku /viz. 1/. Program pro analýzu JCL je poměrně složitý a musí kromě běžného JCL zpracovávat i texty, určené pro modifikátor jobů, používaný v našem středisku (jedno zadání JCL uložené v knihovně může být podkladem pro generování několika jeho variant - běžně 3 - 4, ale třeba i 10 - 15 variant), musí rozpoznat běžně používané procedury, atd.

Při každé aktualizaci RUT.JOBLIB se doplňuje tabulka jmen modifikovaných nebo nově zařazovaných členů knihovny. Vlastní zpracování JCL se pak provádí obvykle při týdenní údržbě systémových knihoven. Při tom vznikne sekvenční soubor, který se použije v dalším kroku jako vstup pro aktualizaci báze dat. Podobně se při týdenní údržbě knihovny zdrojových textů vytváří seznam jmen zdrojových textů programů, které byly v knihovně aktualizovány a jednou za měsíc se provede aktualizace komentářů k programům.

Stručný popis vět, uložených v bázi dat a jejich některých datových položek:

Věty SUBSYSTEM, SKUPINA_ULOH a ULOHA obsahují pouze identifikátor a název příslušné entity. Pokud se při aktualizaci vyskytne např. job z úlohy, která se dosud v bázi dat nevyskytuje, vznikne nová věta ULOHA, která obsahuje pouze identifikátor, a její název se doplní ručně kdykoli později.

Věta JOB obsahuje jméno jobu a jméno jeho autora.

Věta VARIANTA_JOBU obsahuje jméno varianty jobu, účtovací údaj a údaje o prioritě, regionu. Věty variant jobů vznikají při zadání varianty jobu pro u nás používaný modifikátor jobů.

Věta KROK obsahuje jméno kroku jobu a parametr, předávaný programu.

Věta KOMENTAR obsahuje text popisující funkci programu, jméno autora, datum odladění a značn zdrojového textu.

Věta SESTAVA obsahuje číslo sestavy, účtovací informaci, jméno dávky tiskových sestav, v níž se tiskne, jméno tiskového souboru a jméno nosiče souboru.

Věta SOUBOR obsahuje jméno souboru, nosič, typ souboru vzhledem k jobu (interní, externí).

Věta VAZBA_KROK_SOUBOR obsahuje typ souboru (vstupní, výstupní, update nebo tabulka) vzhledem ke kroku jobu.

Při prohlížení báze dat dokumentačního systému lze vyjít od libovolného vstupního bodu - věty a umístění calc. Nejčastěji se zadává klíč věty JOB, PROGRAM, SOUBOR nebo SESTAVA (viz. obrázek). Přitom např. při zadání klíče pro job je možno dále volit tři typy zobrazení: P (přehled), S (seznam sestav) nebo V (přehled vazeb).

Věty SUBSYSTEM, SKUPINA_ULOH a ULOHA mají význam pouze pro úplnost a přehlednost dat a používají se obvykle jen při výběrech pro tiskové sestavy. Tyto věty je možno aktualizovat on-line (např. doplnění názvu úlohy).

Prohlížení báze dat z terminálu je řešeno jedním externím taskem v pseudokonvezačním režimu a počítačnickou obrazovkou pro menu a dalšími šesti obrazovkami, které uživatel volí podle potřeby. Obsluha tasku je jednodušší než snaha popsat možnosti, které jsou ostatně nejlépe vidět z obrázku, na němž je uveden diagram báze dat dokumentačního systému.

Příklady:

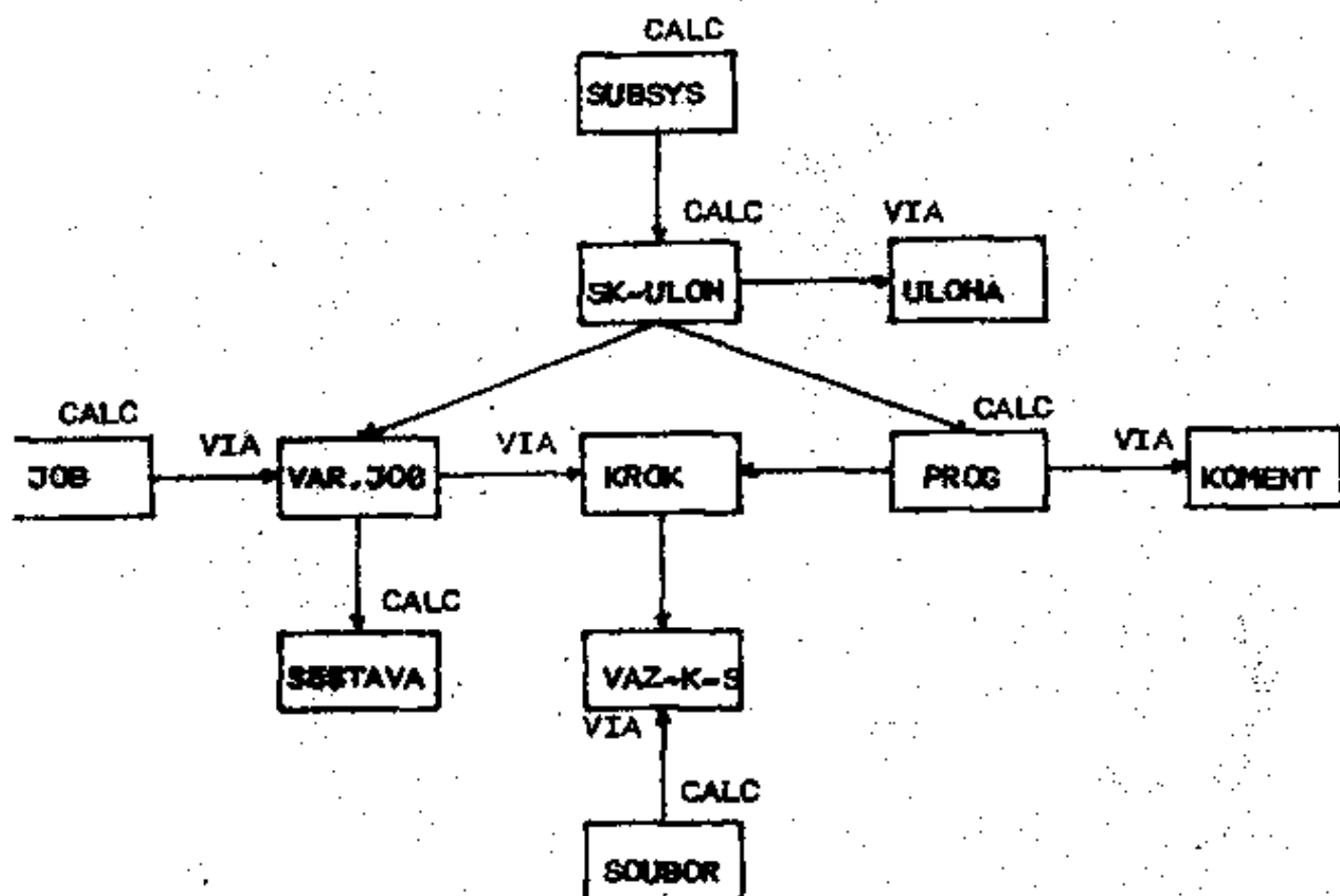
Zadáme-li jméno jobu, pak při typu zobrazení P se zobrazí v záhlaví obrazovky jméno jobu, autor, priorita, region. Další řádky jsou seřazeny podle kroku a na řádku se objeví: krok, program, parametr, jméno a typ souboru (I,O,U,T), nosič a pokud krok používá více souborů, budou uvedeny na dalších řádkách ve sloupcích soubor, typ souboru, nosič, atd. pro další kroky jobu.

Při typu zobrazení S se pod stejným záhlavím vypíše přehled tiskových sestav, vystupujících z jobu.

Při typu zobrazení V se k výše uvedené hlavičce zobrazí přehled externích souborů jobu, tj. jméno jobu, ve kterém soubor vznikl (vlevo od jména souboru) nebo jméno jobu, do něhož soubor vystupuje (vpravo od jména souboru).

Podobně, vyjdeme-li od jména programu, lze např. zjistit jména všech jobů i jejich variant a kroků, z nichž je zadáný program volán, včetně jmen a typů souborů, které program používá.

Z diagramu báze dat je vidět další možnosti. Např. při požadavku na změnu struktury souboru může programátor nebo analytik pohodlně zjistit, do kterých programů se změna promítne, tak že za vstupní bod zvolí SOUBOR. Pokud vznikne nějaký speciální požadavek, pro který nevyhovuje žádný standardní postup, může se každý programátor, který zná schéma báze dat dokumentačního systému, zjistit cokoliv pomocí IDJ (integrováný dotazovací jazyk).



Hrubý diagram báze dat IDMS pro dokumentační systém

Literature:

/1/ M. Zvoniček: Použití systému D pro řešení úloh zpracování hromadných dat. Sborník Programování '88. DT ČSVTS Ostrava, 1988

/2/ Fiktivní literatura IDMS