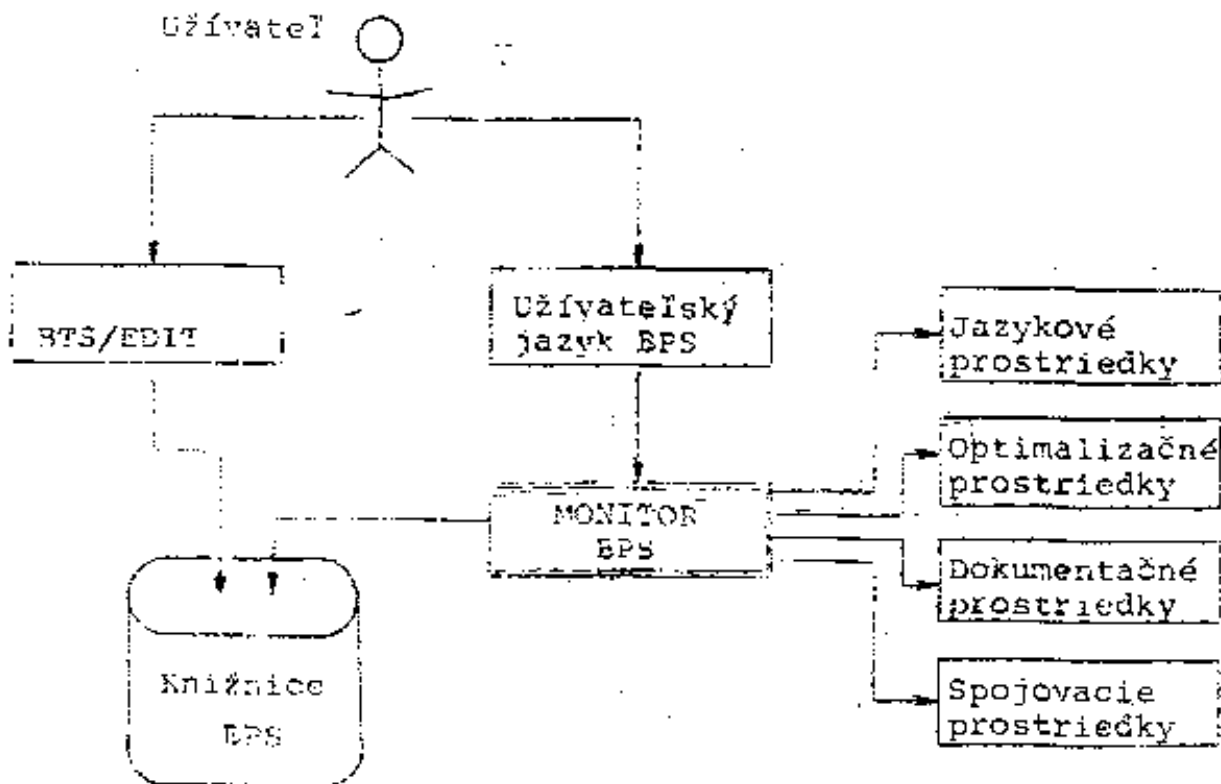


EPS 1. VERZIA, APLIKÁCIA ABSTRAKTNÝCH DÁTOVÝCH TYPOV

1. Stavba systému BPS

Prvá existujúca verzia systému BPS (dokončená bola koncom r. 1978) sa skladá z viacerých programovacích prostriedkov (podsystemov), ktoré schématicky znázorňuje obr.1.



obr. 1

Ako vidieť z obrázku, užívateľ priamo komunikuje iba s monitorom systému BPS (prostredníctvom jednoduchého užívateľského jazyka), pomocou ktorého ovláda činnosť celého systému. Monitor EPS (riadiaci program) je z hľadiska realizácie BPS jadrom celého systému. Preberá od užívateľa pokyny, odovzdáva mu správy o úspešnosti (neúspešnosti) príslušnej činnosti a riadi a koordinuje činnosť ostatných podsystemov BPS. Z ostatných častí systému BPS sa v ďalšom podrobnejšie zmienime o jazykových prostriedkoch (najkomplikovanejšia a najzaujímavejšia).

šia časť), ktoré sú tvorené kompilátormi dvoch jazykov (hlavného systémového jazyka BPS/L a štandardného assemblera pre príslušný počítač. Kompilátor BPS/L je štvorprechodový a jeho jednotlivé prechody uskutočňujú nasledovné činnosti:

1. prechod : syntaktická analýza programu
2. prechod : sémantická analýza programu
3. prechod : sémantické transformácie (generácia mezikódu)
4. prechod : generácia cieľového kódu

Integračným prvkom systému BPS sú knižnice, v ktorých sa nachádzajú všetky programy, ktoré užívateľ pomocou BPS vyvíja (zdrojové- textové, preložené-binárne, absolútne) a informácie o nich (pre každý modul sa uchováva zoznam použitých modulov, globálne deklarácie, tzv. referenčné číslo udávajúce verziu globálnych deklarácií modulu, atď.). Knižnice systému BPS vznikli formou rozšírení systémových knižnic interaktívneho systému BPS [1] a off-line systému EDIT [2].

BPS pracuje len s programami uloženými vo svojich knižniciach. Týmto sa dosahuje centrálna evidencia programov, čo prináša viaceré výhody: je možné automatizovať i tie činnosti, ktoré sa doteraz robievali ručne (spojovanie programov do vykonateľných celkov apod.), alebo ich alebo možné vôbec vykonávať (kontrola správnej štruktúry programu, väzieb medzi modulmi, automatická rekompilácia pri zistení chybných väzieb a pod.). Užívateľ BPS má evšiem priamy prístup iba k textovým tvarom programov prostredníctvom systémov BPS alebo EDIT. Ku všetkým ostatným častiam knižnic EPS prístupuje už len samotný systém, čím sa odstraňuje množstvo prác potrebných pre údržbu knižnic a možný zdroj chýb pri manipuláciách s knižnicami. Knižnice majú jednoduchú štruktúru a sú opakovateľne prístupné.

Medzi tretím a štvrtým prechodom kompilátora systémového jazyka BPS/L (sémantické transformácie a generátor kódu) sú zapojené pretriadky na optimalizáciu vytváraného kódu. Tento prechod vykonáva viaceré počítačovo nezávislé optimalizácie kódu, akými sú napríklad výpočet konštánt a konštantných výrazov ("folding"), vyhodnocovanie logických operácií

(podľa Mc Carthyho: ak sa dá z hodnoty prvého operandu určiť hodnota celej operácie, druhý operand sa vôbec nevyhodnocuje), odstraňovanie spoločných výrazov a podvýrazov, otvorené operácie (INLINE-OPERATIONS), t.j. substitúcia vyvolávania operácií ich telom, atd.

Dokumentačné prostriedky systému BPS produkujú rozne typy dokumentácie na rôznych úrovniach (počas rôznych etáp) procesu programovania. Užívateľ (na požiadanie) obdrží výpis zdrojového tvaru svojho programu, protokol křížových odkazov (cross-reference-listing) na úrovni vnútromodulárnej, ale aj medzomodulárnej, graf štruktúry programu (tzv. graf viditeľnosti, v ktorom sú zachytené vzťahy medzi modulmi) [3] .

Spájanie modulov do vykonateľných celkov je ďalšia činnosť, ktorá sa pomocou systému BPS automaticky uskutečňuje. Užívateľ stanoví iba meno hlavného modulu vykonateľného celku a BPS na základe medzomodulových väzieb zistí všetky moduly, ktoré do daného celku patria. Na základe týchto väzieb skonštruuje graf viditeľnosti, skontroluje sa acykličnosť takéhoto grafu, ako i správnosť väzieb medzi modulmi. V prípade pozitívneho výsledku uvedených kontrol sa vytvorí vykonateľný celok.

2. Jazyk BPS/L

BPS/L (hlavný systémový jazyk BPS) vychádza z najnovšieho jazyka prof. N. Wirtha "MODULA" [4] . Jazyk BPS/L je modulárnym jazykom. Základnou kompilačnou jednotkou je modul. Program alebo programovaný systém kódovaný pomocou BPS/L je zostavený z modulov. Z toho vyplýva, že modul je i organizačnou jednotkou v zmysle štruktúry systému BPS. Z hľadiska jazykového modulu realizuje abstraktný dátový typ [5] : užívateľ narába s dátami modulu (dátové štruktúry modulu) len prostredníctvom operácií modulu (funkcie alebo procedúry). Dátová štruktúra je pre užívateľa práve to, ako sa navonok javí. Užívateľ vidí iba dátové operácie a len cez ne vie, ako sa správa a mení dátová štruktúra. Nemá prístup k reprezentá-

cii dátovej štruktúry. Bližšie, pozri [3].

Každý modul v jazyku BPS/L obsahuje časť deklaračnú (deklarácie konštánt, typov a premenných). Premenné sa alokujú staticky. Okrem toho môže modul obsahovať i deklarácie operácií (pravých procedúr alebo funkcií). Každá operácia modulu môže obsahovať opäť deklarácie konštánt, typov a premenných (sú lokálne v rámci príslušnej operácie), ale už žiadne ďalšie deklarácie lokálnych operácií. Vkladanie operácií do seba ("operations vesting") nie je prípustné. Lokálne operácie sa môžu navzájom vyvolávať, pričom každá z nich môže priamo (bez obmedzení) vyvolávať dovezené operácie z iných modulov. V súlade s princípmi separátnej kompilácie (ktorú systém BPS na rozdiel od jazyka MODULA umožňuje) je teda možné komunikovať medzi modulmi pomocou operácií, resp. objektov vyvázaných (deklarujú sa ako GLOBAL). Cyklické volanie operácií je možné iba vtedy, keď všetky operácie jedného cyklu sú deklarované ako dynamické (rekurzívne). Nerekurzívne operácie sa alokujú staticky, zatiaľ čo rekurzívne dynamicky.

Dovážať sa môžu objekty (konštanty typy, premenné, operácie) len do hlavnej časti modulu a spolu s globálnymi a lokálnymi objektami modulu sú viditeľné (použiteľné) vo všetkých operáciách modulu. Princíp dovážania, resp. vyvážania objektov do - a z - modulov sa v rámci jazyka BPS/L odlišuje od podobného v jazyku MODULA. Zatiaľ, čo v jazyku MODULA je treba deklarovať identifikátory všetkých objektov, ktoré sa do daného modulu dovážajú z iných modulov, v BPS/L stačí uviesť mená modulov, čím sa sprístupnia automaticky všetky objekty daného modulu.

Vyvázané premenné sa mimo modulu, v ktorom sú deklarované, správajú ako konštanty (nikde sa im nemôže priradiť hodnota). Vyvezené konštanty sa správajú ako normálne lokálne deklarované konštanty. Vyvážanie konštánt neporušuje princíp abstrakcie v BPS/L, zaviedli sme ho za účelom zjednodušenia a zefektívnenia narábania s objektami v separátne kompilovaných moduloch (konštanty a premenné sú v podstate funkcie vracajúce príslušnú hodnotu). Okrem konštánt a premenných je možné z modulu vyvievať i typy, ktoré majú tú zvláštnosť, že mimo definujúceho modulu nie je známa ich štruktúra.

Dovezené identifikátory, globálne a lokálne identifikátory objektov modulu a lokálne identifikátory každej jednej operácie musia byť navzájom odlišné. Týmto sa zabráni tomu, aby identifikátory, ktoré sa prekrývajú, mali viacnásobnú intenciu. Operácie medzi sebou môžu mať rovnaké identifikátory s roznoými intenciami, Lokálne premenné operácií si nezachovávajú svoje hodnoty medzi dvomi vyvolaniami.

Celý problém spoločného používania konštánt, typov, premenných a operácií, t.j. princíp separátnej kompilácie je veľmi efektívne vyriešený pomocou konceptu knižnic systému BPS ako integrálnej časti kompilátora BPS/L. Koncept modulu s vyváženými objektami sa veľmi dobre spája s myšlienkou knižnice: všetky vyvážené objekty jedného modulu sa ako celek uložia do knižnice a ako celek sa privezú - okopírujú - do modulu, kde sa použijú. Týmto spôsobom je zaručené veľmi štruktúrované spoločné používanie konštánt, deklarácií typov, ako aj kontrola konzistencie spoločných premenných a parametrov operácií. Toto všetko sa dá dosiahnuť vnútri kompilátora bez použitia zvláštnych mechanizmov, ktoré by už neboli obsiahnuté v princípe modulu.

Chápanie konceptu modulu značne zeleгантnejšie spojenie modulu s princípom otvorených operácií (vloženie tela operácie na miesto jej vyvolania, t.j. priama aplikácia sémantiky vyvolania operácie: Pravidlo kopírovania "COPY FILE"). Týmto "spojením" nie je viac treba priväzovať do modulu konštanty a premenné. Konštanty a aj premenné sa dajú chápať ako bezparametrové funkcie.

Z doteraz uvedených charakteristických črt modulu vyplýva, že i keď si každý modul zachováva svoju autonómiu, má isté vzťahy navonok, k ostatným modulom. V súlade s princípom súčasných metodologických poznatkov v jazyku BPS/L sú tieto vzájomné vzťahy hierarchické. Hierarchiam rozumíme vzťah medzi modulmi, ktorý dovoľuje požívať objekty modulov na hierarchicky nižšej úrovni a teda aj vytvárať objekty, ktoré môžu byť použiteľné modulmi z úrovne hierarchicky vyššej. Modulová hierarchia musí vytvárať acyklický graf (acykličnosť tohto grafu sa systémom automaticky kontroluje).

Takýmto ponímaním medzimodulárnej komunikácie sa BPS/L najväčšmi odlišuje od jazykov postavených na báze blokového princípu.

Jazyk BPS/L dovoľuje používať viacero typov modulov (majú zvláštne označenie): modul na najvyššej úrovni hierarchického grafu (vrchol grafu) je označený ako hlavný (prefixovaný rezervovaným slovom MAIN). Takýto modul nemá žiadne globálne objekty, ale iba lokálne. Vyvolaním tohto modulu sa vlastne odštartuje výpočet celého programového systému a tým sa i nepriamo riadi jeho priebeh:

Moduly, ktoré obsahujú trézne strojovo závislé manipulácie (vstupne-výstupné operácie, konverzie typov a pod.) musia byť deklarované ako strojovo závislé (prefixované sú rezervovaným slovom DEVICE).

Okrem týchto dvoch druhov modulov (MAIN a DEVICE) existujú ešte i špeciálne moduly, ktorých všetky operácie sú napísané v jazyku strojových inštrukcií, v assembleri. Tieto je nutné označiť ako ASSEMBLER moduly. Záverom uvedieme ešte definíciu modulu v jazyku BPS/L.

Modul v jazyku BPS/L pozostáva :

z hlavičky modulu :

```

MAIN
DEVICE      MODULÉ  modulename ;
ASSEMBLER

```

zo zoznamu modulov,

z ktorých sú do daného

modulu dovážané objekty : USE module₁, module₂, ... module_n;

zo špecifikácií vyvázaných

objektov :

```

GLOBAL
... deklarácie      CONST .....
                   TYPE  .....
                   VAR   .....
                   OP    .....

```

z deklarácií lokálnych

objektov :

LOCAL

..... deklarácie

CONST

TYPE

VAR

OP

z inicializačnej časti :

VALUE

z tela modulu :

BEGIN

z deklarácií tiel

globálnych a lokálnych

operácií :

OPERATION-BODY *

Zoznam použitej literatúry :

1. FISCHER, K. : BTS, užívateľská príručka, prac. mat. OPS/VVS, VVS, Bratislava, 1977
2. VODA, P. J. : EDIT, užívateľská príručka, prac. mat. OPS/VVS, VVS, Bratislava, 1976
3. FISCHER, P. a kol. : BPS, Bratislavský programovací systém, výskumná práca č. 144, VVS, Bratislava, 1978
4. WIRTH, N. : MODULA : A Language for Modular Multiprogramming, Institut für Informatik, ETH, Zürich, 1977
5. GRUSKA? I., WIEDERMANN, I., ČERNÝ, I. : Abstraktné typy dát, Zborník referátov, SOFSEM'78, VVS, Bratislava, 1978
6. FISCHER, P. a kol. : BPS, Rozpracovanie prvej verzie, výskumná práca, VVS, Bratislava, 1979