

Ing. Otakar P O L A C H

NHKG Ostrava

Dokumentační prostředek HIPO

Vedoucí pracovníci dnešních výpočetních středisek jsou postaveni před složité rozhodování - jak uspokojit uživatele, kteří očekávají komplexnější a složitější aplikace než v minulosti a současně jak vyřešit otázku rostoucích kapacitních potřeb svého útvaru. Zjišťují, že stále větší podíl programátorů se zabývá údržbou, která má jedinou tendenci - jen roste. Je dnes běžným jevem, že 60 % programátorů a analytiků v organizaci se zabývá údržbou programových systémů.

Co mohou dělat? Jestliže odmítnou požadavky uživatele, potom ti budou nespokojeni dnes a vedení zítra. Jestliže se budou snažit je uspokojit, potom jejich pracovníci musí pracovat přesčas nebo musí získat více pracovníků. V každém případě výsledkem budou zvýšené náklady a vyhlásky ve zdůvodňování proč potřebují další pracovníky a zvýšení nákladů.

Podívejme se na tento problém z hlediska uživatele. Je oprávněn požadovat pokračování, rozšíření či zrychlení služeb poskytovaných výpočetním střediskem? Výpočetní středisko je servisní funkce a uživatel žije v předstávě, že počítač je zde proto, aby pracoval jen pro něj.

Vychází z toho, že zpracování dat je servisní funkcí, která je podstatou celé organizace, která musí růst, měnit se a která nebude nikdy ukončena. Z těchto důvodů vedení uživatelů očekává, že se bude pokračovat ve vývoji a zlepšení servisních služeb poskytovaných v minulosti.

I když ve výpočetním středisku dojde k inovaci hardwaru, efektivnost tvorby programových zaostává. Z hlediska tempa vývoje zůstává tvorba softwaru na úrovni manufakturní výroby proti vysoce vyspělé výrobě hardwaru.

Tento neuspokojivý stav vyvolal potřebu oprostit se od klasických metod a vytvořit nové techniky, zaměřené na zvýšení produktivity analytických a programovacích prací při vývoji a implementaci aplikací, na zjednodušení údržby programovacích systémů a zjednodušení projekční dokumentace. Tyto techniky mají různé označení, např. strukturované programování, strukturované analýza, zlepšené technologie programování apod. Většina z nich zahrnuje strukturované programování, metodu vývoje směrem shora-dolů, organizaci týmu vedoucího programátora, strukturované používání a koncepci knihovnicka programu. Firma IBM doplnila tyto metody a techniky o dokumentační techniku HIPO a celý soubor technik pojmenovala IPT (Improved Programming Techniques).

Dnešní dokumentace programových systémů se skládá ze slov, tabulek a vývojových diagramů. Vytvořit dobrou dokumentaci, která by byla úplná a přesná je časově náročná věc. Je proto samozřejmé, že dokumentace se často připravuje pozdě a v době napjatých termínů se v ní provádějí změny. Rovněž orientace v dnešních popísených a objemných dokumentačních svazcích je složité. Co dokumentaci chybí je popis funkcí systému, t.j. co program dělá, místo toho jak to dělá. Zdůraznění funkcí v dokumentaci může přispět k vyšší efektivnosti nejen při vývoji, ale i při údržbě programových systémů.

Z hlediska těchto potřeb vyvinula firma IBM dokumentační techniku HIPO (Hierarchy plus Input - Process - Output), která je určena všem pracovníkům, kteří pracují s dokumentací. HIPO dokumentace se skládá ze souboru diagramů, které znázorňují funkce v grafické podobě, uspořádané od nejzákladnějších informací až po nejpodrobnější detaily, se znázorněním všech vstupů a výstupů na každé úrovni.

Návaznost HIPO na ostatní techniky

HIPO předpokládá, že programový systém je organizačně uspořádán podle hierarchické struktury funkcí. Do oblasti funkce s nejvyšší úrovní patří všechny jí podřízené sub-funkce. Ty funkce, které potřebují další vyjasnění se berou jako hlavní funkce, které obsahují další úroveň subfunkcí. Tento proces členění pokračuje v tolika úrovních, kolik je jich třeba pro definování funkcí.

Hierarchická struktura HIPO je velmi vhodná pro vypracování funkčního návrhu směrem shora-dolů, od nejvyšší funkce k nižším úrovním. Při vývoji shora-dolů se funkce implementují ve stejné posloupnosti a danou strukturou. Nejvyšší modul obsahuje nejvyšší úroveň řízení a rozhodování pro každý program v systému a předává řízení modulům na nižší úrovni. Části každého programu se postupně integrují.

Jestliže se při implementaci používá strukturované programování, potom se funkce berou jako samostatné jednotky. Program se píše po segmentech, z nichž každý má jen jeden vstup a jeden výstup. Segmenty se mohou vytvořit přímo z HIPO diagramů, které jsou rovněž zakreslené pouze s jedním vstupem a jedním výstupem. Podle praktických zkušeností je vhodné označit kód segmentu na jeden list

*) je možné provádět i v rámci 1 programu a "segmenty" chápat jako perform (problem identifikatory, rozboru po stránkách s vstupem a výstupem)

(přibližně 50 řádek) a zřetelně oddělit subfunkce nebo substruktury. Tato doporučení zvyšují čitelnost a umožňují snadnou pochopitelnost.

Kombinace metody vývoje shora-dolů a strukturovaného programování vede k vysoké modularitě programu, a to jak ve funkční tak i logické struktuře. HIPO diagramy jsou logicky rozdělení funkcí, identifikovaných při vývoji shora-dolů a poskytují potřebnou dokumentaci od zahájení prací na projektu až po jeho implementaci.

Jinou koncepcí, která se používá s vývojem shora-dolů a strukturovaným programováním je organizace týmu vedoucího programátora. Týmová spolupráce představuje změnu v organizačním přístupu, od volně strukturovaných skupin k vysoce strukturovanému týmu programátorů - specialistů. Jádro týmu se skládá z vedoucího programátora, záložního programátora a knihovníka, případně jiných pracovníků podle potřeby. Hlavní programátor navrhuje hlavní funkce systému a kóduje moduly s nejvyšší úrovní. Záložní programátor pomáhá vedoucímu programátorovi v jeho činnostech. Knihovník je zodpovědný za uložení všech informací do knihovny, jeho hlavní povinností je průběžně udržovat aktualizované a vizuálně přehledné programy, zkušební data a vést evidenci o vývoji projektu. V knihovně by měla být i dokumentace HIPO a měl by ji udržovat knihovník. Jestliže tým se doplní o nové programátory pro vývoj programů na nižší úrovni, potom tito mohou pracovat nezávisle se samostatnými HIPO diagramy.

V programovacích skupinách, které používají strukturované programování, metodu řešení směrem shora-dolů a organizaci týmu vedoucího programátora, se zavádí strukturované posuzování. Posuzovatelé "procházejí" systémem krok po kroku a simulují zkoušené funkce. HIPO jako pomůcka pro sestavení návrhu směrem shora-dolů a jako dokumentační technika je velmi vhodná pro strukturované

rované posuzování. Grafická reprezentace funkcí poskytuje něco konkrétního a hmatatelného, co je možné posuzovat krok po kroku směrem k většímu podrobnosti.

Pomůcky pro přípravu HIPO diagramů

Pro kreslení HIPO diagramů existují dvě pomůcky. První z nich je formulář HIPO, který má dvě části -
- první obsahuje sekce pro vstup, zpracování a výstup,
- druhá obsahuje rástr pro rozšířené informace.

Druhou pomůckou je šablona HIPO, GX20-1971, pro zakreslování HIPO diagramů. Je upravena pro formuláře HIPO a obsahuje všechny potřebné symboly.

Hlavní cíle HIPO

- vytvořit takovou strukturu, která umožní rozoznat jednotlivé funkce systému. Diagramy jsou uspořádány do hierarchické struktury, podobně jako organizační schéma, kde každý diagram jakékoli úrovně je podmožinou diagramu vyšší úrovně. Složitě systémy nebo programy se tak mohou rozdělit do snadno srozumitelných celků,
- specifikovat funkce, které má program provést, aisto programových instrukcí, potřebných pro provedení těchto funkcí,
- vytvořit pro každou funkci vizuální popis použitého vstupu a vytvořeného výstupu na každé úrovni programu.

bylo by třeba obkreslit + popsat význam symbolů

HIPO abecník se skládá ze tří druhů diagramů:

- obsahová tabulka - obsahuje jména a identifikační označení všech přehledných a podrobných diagramů v HIPO sborníku, znázorňuje strukturu sborníku a vztahy funkcí v hierarchii. Obsahová tabulka má formu organizačních schémat. Podle obsahové tabulky může si čtenář vyhledat potřebnou úroveň informací nebo specifický diagram, bez prohledávání celého sborníku.

- přehledné diagramy - HIPO diagramy vyšší úrovně, které popisují hlavní funkce a obsahují odkazy na podrobné diagramy. Přehledné diagramy znázorňují základní vstupy, výstupy a zpracování. Zpracovací sekce diagramu obsahuje sérii očíslovaných pracovních kroků, popisujících funkci, která se má provést. Vstupní sekce obsahuje jen ta data, se kterými se bude pracovat ve zpracovací sekci. Šipky a spojnice spojují vstupní data s pracovními kroky. Výstupní sekce obsahuje jen ta data, která byla vytvořena nebo modifikována v pracovních krocích, ke kterým jsou připojena šipkami a spojnicemi. Všechny sekce se mohou doplnit o další informace v oddílu "Rozšířený popis".

- podrobné diagramy - jsou základní prvky sborníku. Popisují specifické funkce, znázorňují vstupy a výstupy konkrétních dat a obsahují odkazy na ostatní HIPO diagramy nižší úrovně nebo jinou dokumentaci než HIPO, jako jsou vývojové diagramy, rozhodovací tabulky, popisy vět apod. Podrobné diagramy obsahují také oddíl "Rozšířený popis", který rozvádí postup zpracování a obsahuje i odkazy na jednotlivé části programu.

Dokumentace HIPO se obvykle připravuje ve dvou sbornících, které obsahují prvotní návrh a podrobný návrh. V některých případech se vyhotovuje i sborník pro údržbu programového systému. Všechny druhy sborníků obsahují uvedené tři druhy diagramů.

- Sborník prvotního návrhu - připravuje navrhovatelé skupina při zahájení práce na projektu. Popisuje základní funkce projektu a používá se jako podůbrka při navrhování.
- Sborník podrobného návrhu - připravuje řešitelé skupina. Analytici a programátoři vycházejí z prvotního návrhu, specifikují podrobnosti, doplňují další potřebné úrovně HIPO diagramů a jinou dokumentaci než HIPO. Během kódování diagramu doplňuje se rozšířený popis o návěští použité v programech a jiné informace, týkající se implementace. Po skončení programu je k dispozici úplná, přesná a snadno pochopitelná dokumentace.
- Údržbářský sborník - používá se pro údržbu systému. Je to v podstatě sborník podrobného návrhu, z kterého se vyřadily některé diagramy nižší úrovně, původně určené pro implementaci.

Příprava HIPO diagramů

Nejdříve se vypracovává obsahová tabulka, které znázorňuje hierarchickou strukturu systému. U prvotního návrhu se obvykle funkce specifikují bez ohledu na to, jak budou zapřazeny do systému. Nejvyšší blok v obsahové tabulce uvádí hlavní funkci systému, v následující nižší úrovni se tato funkce rozčleňuje do logických subfunkcí. Dělení na stále nižší subfunkce pokračuje až do úrovně, z které lze snadno pochopit všechny funkce systému (obvykle stačí čtyři až pět úrovní).

Přísná návaznost bloků na přehledné diagramy je zajištěna tím, že každý blok obsahuje identifikační číslo diagramu. Obsahová tabulka se tak zároveň stává "přůvodcem" v dokumentaci.

Přehledné diagramy poskytují čtenáři základní informace o konkrétních funkcích a jsou zároveň i vodítkem k diagramům na nižší úrovni. Z hlediska plnění funkcí programu nemusí se v této době specifikovat jejich posloupnost plnění. Pokud některé pracovní kroky budou vyžadovat dalšího rozvedení, potom se tyto kroky uzavřou do bloku, který se doplní o identifikační číslo diagramu s nižší úrovní, ve kterém jsou uvedeny další podrobnosti.

Přehledné i podrobné diagramy jsou si z grafického hlediska podobné, avšak posláním podrobného diagramu je poskytnout čtenáři všechny informace potřebné pro pochopení funkcí, které jsou specifikované na nejbližší vyšší úrovni. Ve vstupní a výstupní sekci se uvádějí jen konkrétní prvky dat se kterými se pracuje.

Přehledné i podrobné diagramy se doplňují o rozšířené popisy, které obsahují více informací, než je možné uvést ve zpracovací kroku.

Pro zakreslování přehledných a podrobných diagramů se doporučuje:

1. Používat formuláře HIPO.
2. Uvést všechny známe výstupy do výstupní sekce. Některé výstupy je možné přímo specifikovat z požadavků na systém. V přehledném diagramu se výstupy uvádí jen v obecné formě, t.j. ve formě souborů nebo vět.
3. Souběžně zpracovat vstupní a zpracovatelskou sekci, uvést všechny mezivýsledky, které dosud nebyly specifikované.
4. Na vstupní straně uvést jen nutné vstupy - prvky dat. Jestliže některá data se nebudou používat, potom by se neměla uvádět, vstupní sekce bude přehlednější.
5. Zpracovací sekce obsahuje seznam činností, které se mají provést. Každou funkci je vhodné specifikovat co nejmenším počtem slov.

5. Pracovní kroky ve zpracovací sekci se spojují se vstupní a výstupní sekci pomocí spojnic a šipek.

Úvahy o prvním použití HIPO

Je důležité začít s používáním HIPO na projektu co nejdříve, aby návrháři mohli dokumentovat své myšlenky souběžně s návrhem zpracování a aby se funkce neztratily při kódování.

Příprava HIPO diagramů je spíše přirozeným vedlejším produktem procesu myšlení než činností navíc. Při vývoji každého systému se musí v určitých fázích specifikovat požadavky na systém a jeho funkce (včetně jejich vstupních a výstupních informací). Tyto informace se často předávají neformálně ve formě konzultací nebo poznámek na útržkách papíru nebo na školní tabuli. HIPO zavádí způsob formálního záznamu těchto informací.

Ačkoliv HIPO je nejužitečnější když se používá po celou dobu, t.j. od fáze prvotního návrhu až po implementaci, může být příprava HIPO sborníků vhodným prostředkem pro výuku o použití HIPO, jestliže ten kdo bude sborník připravovat dobře zná funkce dané aplikace. Systém může dokumentovat pomocí HIPO v kterékoliv jeho vývojové fázi. Pracovník, který bude připravovat HIPO dokumentaci by měl ignorovat organizaci existujícího programu a měl by se koncentrovat na jeho funkce. Nejdříve by měl zakreslit obsahové tabulky pro funkce nejvyšší úrovně a přehledné diagramy pro všechny hlavní funkce. Potom se mohou rozšířit přehledné diagramy do podrobných diagramů a zaznamenat identifikační údaje do obsahové tabulky. U částečně dokončených systémů se mohou v rozšířených popisech provést odkazy na existující vývojové diagramy a pro ostatní funkce se mohou vypracovat HIPO diagramy.

Pro zahájení práce s HIPO diagramy není třeba žádných zkušeností s kreslením diagramů. Jedinými požadavky jsou základní znalosti o charakteristikách HIPO sborníků, šablona a pracovní formuláře.

Ačkoliv první pokus o sestavení HIPO diagramů může být obtížný, protože lidé nemají zkušenosti s izolováním funkcí (co se má dělat) od zavedeného zvyku popisovat jak to udělat, zkušenosti ukazují, že po identifikaci dané informace trvá zakreslení HIPO diagramu průměrně 30 minut.

Při přípravě prvních HIPO diagramů mohou vzniknout některé problémy, které mohou mít příčinu:

- v nejasných představách o systému
- v nejasnosti, které funkce se mály rozdělit do subfunkcí
- ve snaze vypracovat diagramy nižší úrovně bez diagramů vyšší úrovně
- myslenkovou orientací na logiku programu místo ve funkci programu

V souhrnu může říci, že dokumentační technika HIPO je výhodná jak v oblasti tvorby dokumentace programového systému, tak i v jeho údržbě, tím, že umožňuje:

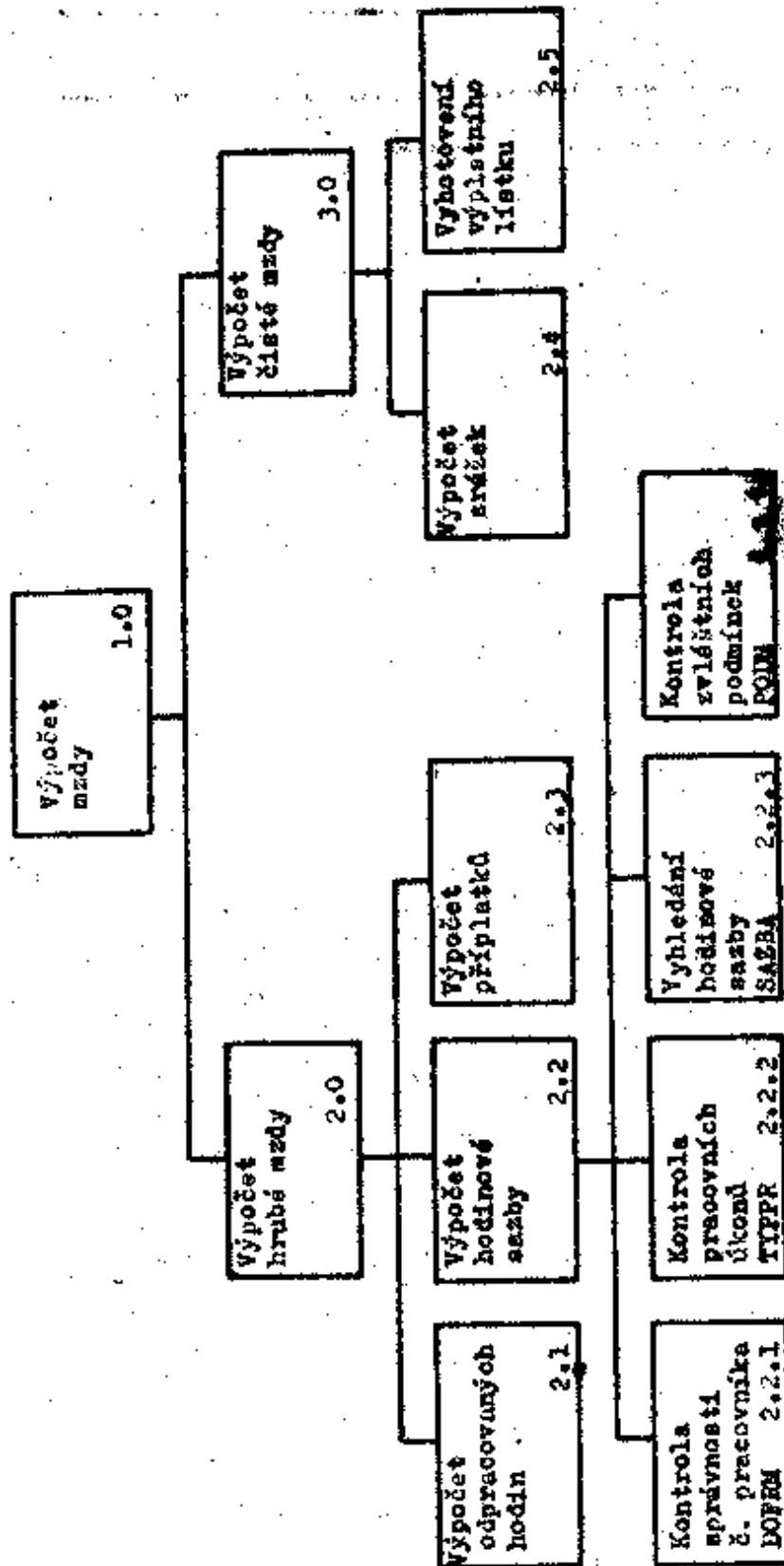
- systematický způsob specifikace požadavků projektu pomocí hierarchické struktury
- snadnější vyhledávání programových chyb a modifikací programu pomocí hierarchické struktury a grafické přehlednosti návaznosti dat
- vypracovat dokumentaci, která umožňuje snadnější pochopení systému a které snižuje pravděpodobnost vzniku chyb nebo neúplné implementace
- vypracovat dokumentaci, která je nedílnou součástí projektu, a to od jeho zahájení až po údržbu systému
- dosáhnout hlubšího přesunu informací a menších duplicít ve vývojovém cyklu

- zjednodušení kódování programů, protože návaznosti dat jsou znázorněny v grafické formě
- zvýšení významnosti technického posouzení projektu, protože systém je rozčleněn na malé snadno kontrolovatelné úseky
- usnadnění funkčních testů, protože zkušební příklady se mohou vypracovat přímo z HIPO diagramů.

Přílohy:

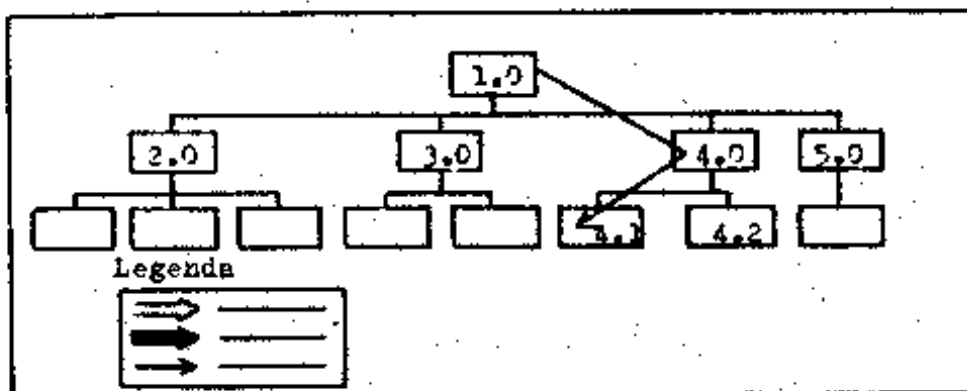
- . Příklad HIPO sborníku
- . Hierarchický diagram
- . Přehledný diagram

Hierarchický diagram

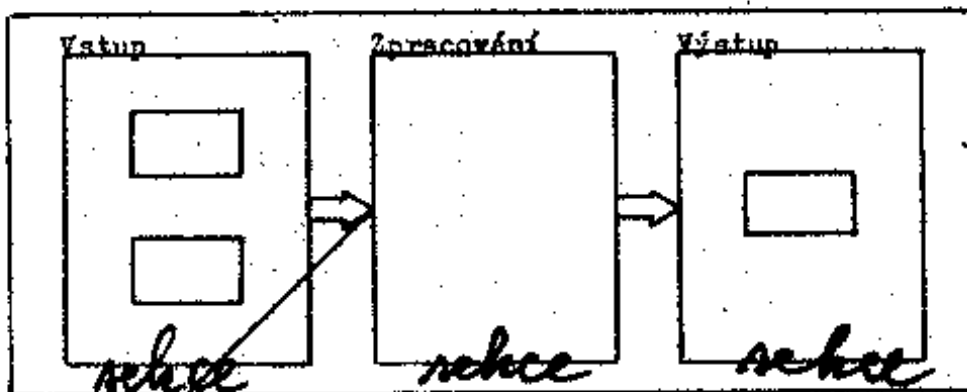


Příklad RIPO sborníku

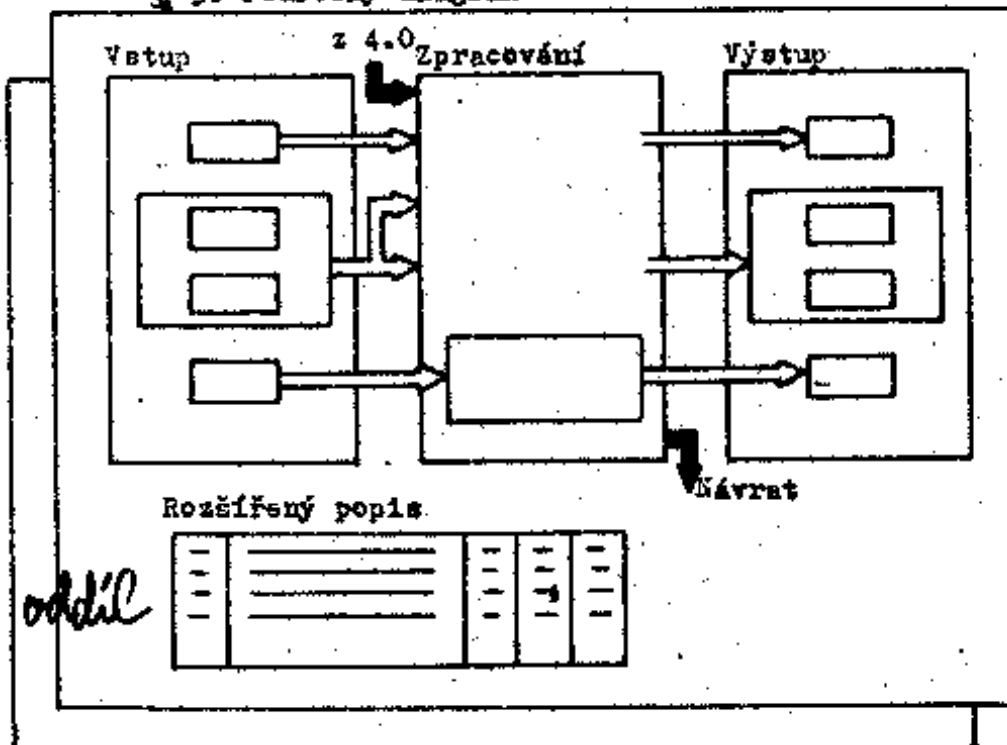
1. Obsahová tabulka



2. Přehledný diagram



3. Podrobný diagram



Přehledný diagram

