

Okruh 3 otázka 9

Počítač jako prostředek řízení, technické a programové vybavení. Mnohaúrovňová organizace počítače, virtuální počítač. Proces. Konvenční strojová úroveň počítače. Technické a programové prostředky. Mikroprogramová úroveň počítače. Mikroprogramování. Počítačový systém, technické a organizační prostředky. Přerušovací systém, struktura, použití. Obvody reálného času. Sítě procesorů a počítačů. Topologie sítí, vlastnosti, použití.

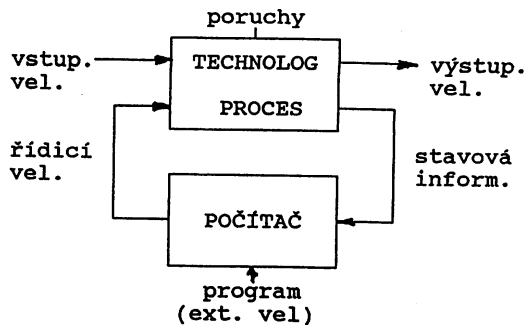
Počítač jako prostředek řízení, technické a programové vybavení

Počítač

Stroj na zpracování informací. Na základě vstupních dat vytváří podle určitých pravidel (programu) výstupní data. Zobrazuje výslednou informaci uživateli nebo ji pro uživatele uchovává v paměti.

Řídící počítač

Pro řízení technologických procesů; má speciální přídavná zařízení I/O, přerušovací systém, reálný čas, D/A, A/D.



Použití počítače :

1. složitý proces (rychlost výp.)
2. levnost výpočetní techniky
3. flexibilita nasazení (program)
4. hierarchická stavba
5. přesnost výpočtů (zobrazení)
6. složité algoritmy

Programové vybavení

software – soubor programů, podprogramů, operační systém + školení operátorů

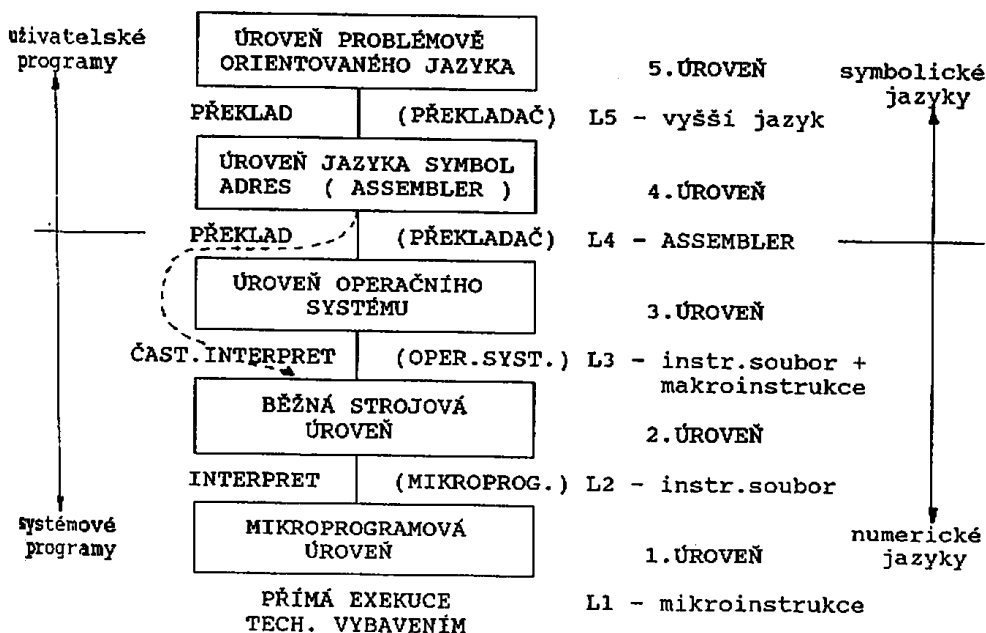
Technické vybavení

hardware = fyzický počítač + školení údržby. Počítač disponuje množinou jednoduchých instrukcí, do níž je nutné převést program pro vykonávání – strojový jazyk.

Mnohaúrovňová organizace počítače, virtuální počítač

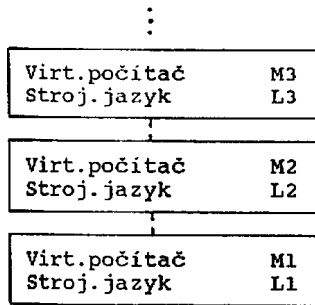
Mnohaúrovňová organizace počítače

jedná se o následující hierarchii vrstev:



Virtuální počítač

Virtuální úroveň počítače – programové i technické vybavení, které nás na dané úrovni zajímá při využití počítače. Na úrovni i zavádíme virtuální počítač M_i s jazykem L_i . Program v L_i je překládán nebo interpretován počítačem M_{i-1} atd.



PRGM v L3 je interpretován programem na M2 nebo M1, nebo je přeložen do L2 nebo L1

PRGM v L2 je interpretován programem běžícím na M1 nebo přeložen do L1

PRGM v L1 skutečně vykonáván technickými prostředky

Interpretace vs. kompilace

Př.: Vykonání programu v L2 na stroji jenž má L1

Kompilace - instrukce v L2 se nahradí posloupností instrukcí v L1

Interpretace - program v L1 zpracovává program v L2 jako data

Proces

Proces

Je probíhající program. (program-pasivní, proces-aktivní)

Stav procesu

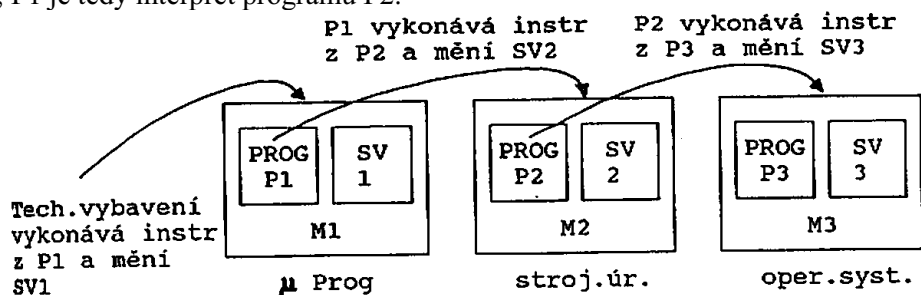
Veškeré informace, které při zastavení procesu umožní jeho pozdější spuštění (*program, následující instrukce, hodnoty proměnných a dat, stavy a polohy všech I/O*) Předpoklad: proces sám nemění svůj program.

Stavový vektor

Všechny proměnné složky stavu procesu. PROCES = PROGRAM + STAVOVÝ VEKTOR.

Změna stavového vektoru

Změnu stavového vektoru nemusí způsobit jenom hardware, ale i jiný program. Př.: Stavový vektor procesu P2 mění proces P1, P1 je tedy interpret programu P2.



Konvenční strojová úroveň počítače

2. virtuálních úrovně počítače – M2. Historicky je první z úrovní, se kterou se operátor setkává.

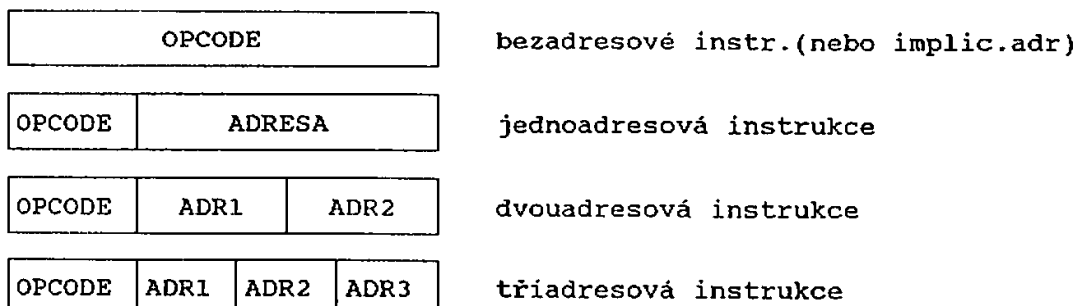
Hard - struktura počítače, procesoru, I/O kanálu, sběrnice, organizace a přístup k paměti, organizace registrů, ...

Soft - instrukční soubor, formát dat a uložení v paměti, adresování, organizace zásobníkové paměti, registry, ...

Technické prostředky

Instrukční formát

Instrukce se skládá z několika polí a může představovat několik po sobě jdoucích slov. Na úrovni M2 rozlišujeme bezadresové, jednoadresové, dvouadresové a tříadresové instrukce.



Adresové části mohou být adresami nebo odkazy na registry (nepřímé určení adresy).

Adresování

Bezprostřední

Operand je součástí instrukce.

Přímé

Instrukce v adresovém poli nebo adresa v následujících datech obsahuje adresu operandu

Nepřímé

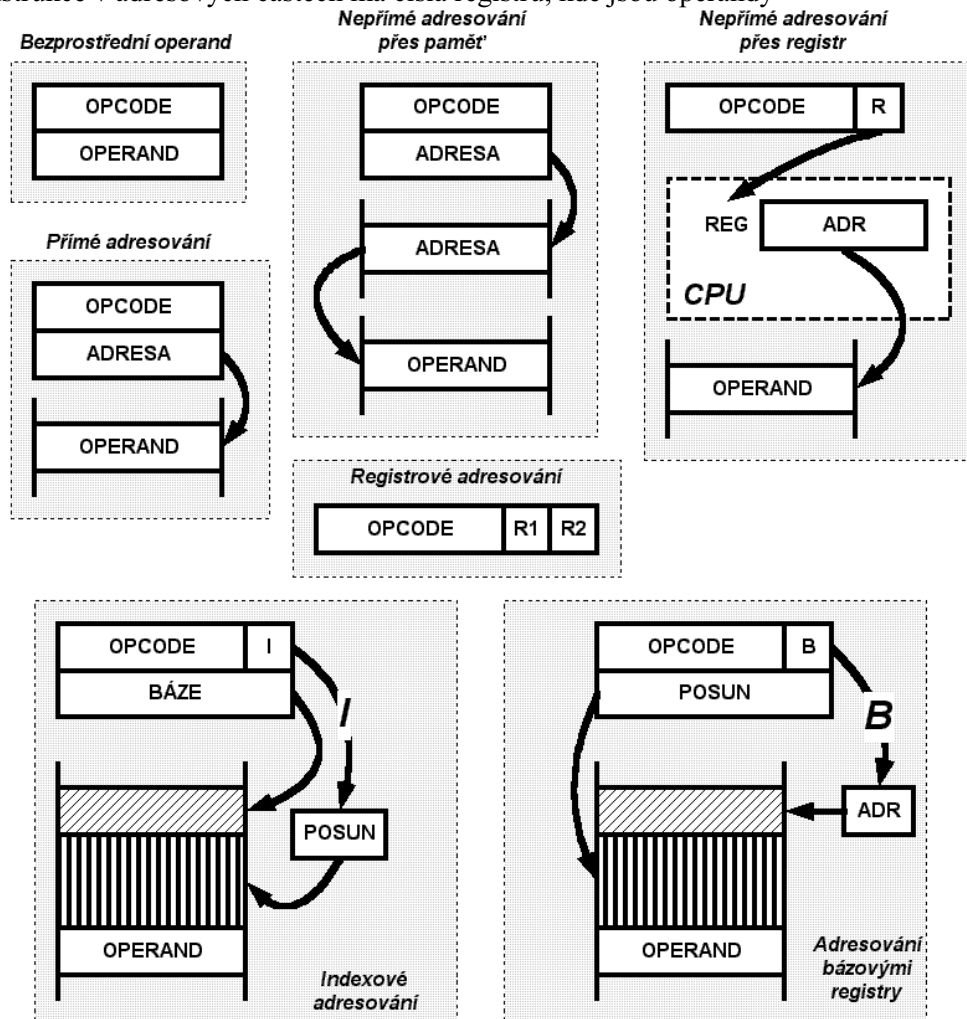
přes paměť - instrukce obsahuje adresu kde je ukazatel operandu

přes registr - instrukce obsahuje číslo registru, který je ukazatelem operandu

indexové - instrukce obsahuje číslo index registru a bázeovou adresu. Adresa operandu je součtem obou adresových částí

bázovými registry - instrukce obsahuje číslo bázeového registru, který obsahuje adresu paměti a adresové posunutí. Adresa operandu je součtem obou adresových částí.

registrové - instrukce v adresových částech má čísla registrů, kde jsou operandy



Reprezentace dat

Typ dat, jejich strukturu a umístění v paměti definuje instrukční soubor procesoru.

Pevná řádová čárka

Minimálně základní paměťová buňka (slovo). Data s vyšší přesností na následujících adresách. Záporná čísla v prvním nebo druhém doplňku.

Pohyblivá řádová čárka

Znaménko, mantisa, exponent. Rozsah dat závisí na instrukčním souboru

Boolská data

0/1. Proměnným je buď přiřazeno celé paměťové slovo (zbytek 0) nebo bitové přiřazení.

Znaky

V jedné paměťové buňce (slově) jeden ASCII znak. Ostatní bity 0.

Řetězce

A) několik znaků ve slově, následující na další adrese

B) každé slovo 1 znak + adresa pokračování

Pole

Používá se báze + indexové adresování. Sousední index na sousední adrese.

Instrukční soubor

Přenosy dat

Vytvoření nové kopie dat na novém místě a zachování původního obsahu (LD, ST, MOV, XCHG)

Dyadické operace

Aritmetika pro integer (+, -, *, /), pro real (+, -, *, /), logika (ADD, SUB, MPY, DIV; FADD, FSUB, FMPY, FDIV; OR, AND, XOR)

Monadické operace (s jedním operandem)

Nulování, inkrement/dekrement, změna znaménka, negace (bitová inverze), aritmetické posuny, logické posuny, rotace, ...

Větvení programu (skoky)

Podmíněné (bity registru, nulovosti registru nebo paměti, znaménko, shoda dvou slov, stavové bity) a nepodmíněné.

Volání podprogramů

Nutnost uchovat adresu návratu z podprogramu. Uchování návratové adresy (do paměti, do registru, do zásobníku) CALL, RET

Řízení cyklů

Podmíněné skoky dané testem počtu průchodů cyklem varianty JCXZ, LOOP

Instrukce I/O

Většinou přesuny IN, OUT

Strojní instrukce

Řízení procesoru NOP, INT, EI, DI, HLT

Ostatní

Další instrukce podle struktury procesoru a jeho funkce.

Programové prostředky

Řízení procesoru programem

Lineární předávání řízení

Předání řízení instrukci s následující adresou, posloupnost narušují pouze skoky, přerušení, větvení...



PC ... Program Counter

Přechody do podprogramů

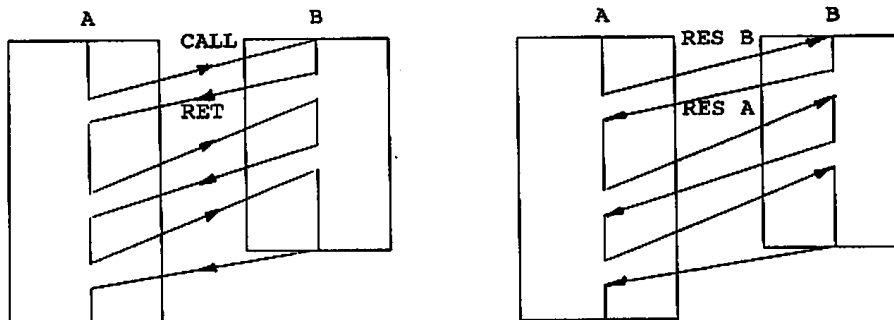
Jedná se o základní techniku strukturovaného programování.

Subroutines

Předávání řízení vždy na počátek, návrat na další instrukci. A používá B jako podprogram. PC->vrchol zásobníku, adresa->PC.

Coroutines

A používá B jako podprogram a obráceně. vrchol zásobníku->pomocné místo, PC->vrchol zásobníku, pomocné místo->PC.



subroutine

coroutine

Přerušování

Automatický přechod na jinou adresu vyvolaný vnějším popudem (hard i soft)

Synchronní – vyvolané samotným programem

Asynchronní – vyvolané okolím nebo chybou procesoru

Mikroprogramová úroveň počítače

M1

Nejnižší úroveň z celé hierarchie - technické vybavení reálného počítače - hradla, registry, čítače, aj. Respektuje obvodové prostředky, časování.

Úroveň - hostitelská = mikroprogramová - na této běží interpreter
- cílová = konvenční strojová - programy, které jsou interpretovány

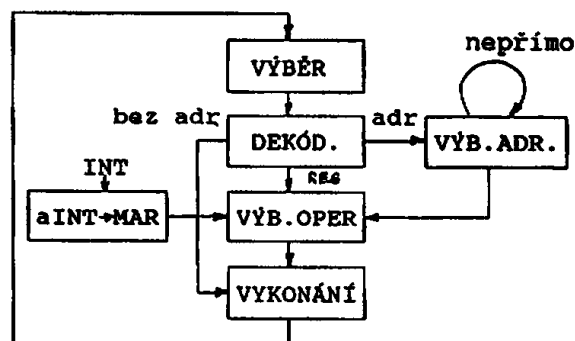
Interpretace instrukcí strojové úrovně mikroprogramovou úrovní představuje ovládání tech. prostředků řadičem - 2 složky:

- programová - představuje cykly: výběr instrukce a provedení ve funkčních blocích (syntaktická činnost) - ucelená sekvence operací pro provedení instrukce
- mikroprogramová - představuje stavy (subcykly) - provedení jednotlivých operací mikrooperacemi.

Základní cykl. instrukce

Typy instrukcí:

- přenosy OP, reg, adresní
- výkonné adresní
- výkonné bezadresní
- testy a větvení, skoky
- I/O
- strojní



Prostředky mikroprogramové úrovně:

- technické - komponenty, z nichž je počítač složen
- programové - množina mikroinstrukcí pro řízení tech. prostředků.

Technické prostředky - komponenty a bloky M1

Registry - paměť informace, délka byte nebo slovo, umístěny v procesoru => rychlá obsluha

Hradla - řídí přenosové cesty - přenos po sběrnicích, zápisy do registrů

Sběrnice - soustava vodičů pro paralelní přenos informace mezi registry. Jednosměrné, obousměrné

Dekodér - dekodování stavů, adres, tj, informací v kódu

Čítače - čítání v kódu

Multiplexery - logické přepínače: podle adresy z n vstupů na jeden výstup

Paměťové registry (porty) - adresový (MAR) a datový (MBR)

Operační jednotka (ALU) - 2 vstupní registry, jeden výstupní registr

Hodiny - pulsy konstantní frekvence synchronizující funkci procesoru na mikroprogramové úrovni

Programové prostředky - základní operace mikroprogramové úrovně

Instrukce konvenční strojové úrovně jsou sestaveny z několika základních operací - mikrooperací. Instrukce pro mikrooperace - mikroinstrukce:

Přenosy mezi registry

Přímý, s posuvem, do několika cílů, z několika zdrojů.

Posuny registrů

Přímo posuvem vlevo nebo vpravo, pomocným registrem a přesunem

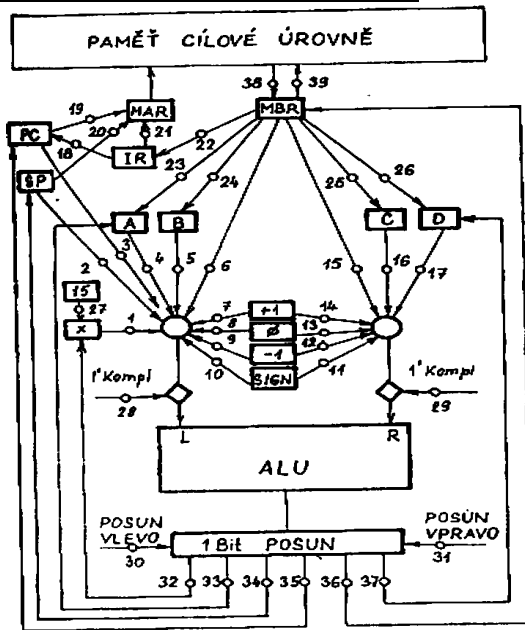
Čtení a zápis operační paměti

Čtení do MBR z adresy definované MAR (MBR, MAR - viz další scan, na který čtenář narazí), zápis na adresu paměti definovanou MAR daty z MBR

Testování bitů

Slouží jako řídicí mechanismus větvení programů. Způsoby: a) výběr mikroprogramové cesty podle testu spec. registru; b) pole mikroinstrukce s dalším registrem definuje adresu pokračování mikroprogramu.

Příklad mikroprogramové úrovně



Funkce registrů

PC - programový čítač
 SP - ukazatel zásobníku
 IR - instrukční registr
 MAR - datový registr paměti
 MBR - datový registr paměti
 A,B,C,D - registry dat pro mikroprogramy
 X - čítač v mikroprogramových smyčkách (LOOP)

Výkon.instrukce

Časové dělení na:
 cykly - ucelená operace (část instr.)
 subcykly (stavy) - otevírání cest - čas. vykonání cyklu.

Vyhrazení stavů (subcyklů)

- 1 stav - přenosy + reg do reg
- cesty 1-17,28,29,12-27
- 2 stav - sčítání, posuny, výstup ALU
- cesty 30-37
- 3 stav - R/W MBR reg
- cesty 38,39

Příklad instrukce ADD STACK

1. POP OPER1
2. POP OPER2
3. ADD OPER1,OPER2
4. PUSH výsledek

CYKL	STAV		
	1	2	3
1	2,12,20	34	38
2	20,23		38
3	4,15	36	39

Některé cesty nesmějí být otevřeny současně -> dělení na stavy.
 Cesta otevřena po celý stav.

VSTUPY SČÍTAČKY

1. X->ALU odečítání 1 od X (LOOP)
2. SP->ALU +1,-1 při operacích zásobníku
3. PC->ALU PC do PAM přes MBR
4. A->ALU aritmetika a přenosy do D,MBR
5. B->ALU " " A,D,MBR
6. MBR->ALU aritmetika
7. +1->ALU inkrement registrů
8. 0->ALU opakování obsahu reg (reg+0)
9. -1->ALU dekrement registrů
10. SIGN->ALU definice carry bitu při posunech
11. SIGN->ALU viz 10 (L/R operand)
12. -1->ALU viz 9 (L/R operand)
13. 0->ALU viz 8 (L/R operand)
14. +1->ALU viz 7 (L/R operand)
15. MBR->ALU viz 6 (L/R operand)
16. C->ALU aritmetika a přenosy do A,D,MBR
17. D->ALU " " A,MBR

PŘENOSY mezi REGISTRY

18. IR->PC JUMP instrukce
19. PC->MAR adresa vybírání instrukce (FETCH)
20. SP->MAR R/W vrcholu zásobníku
21. IR->MAR PUSH,POP operandu
22. MBR->IR čtení OPCODE do IR
23. MBR->A nahrání A reg
24. MBR->B " B "
25. MBR->C " C "
26. MBR->D " D "
27. 15->X inicializace X pro LOOP (16 bit.slovo)

VÝSTUPY SČÍTAČKY

32. ALU->X dekrement X při LOOP
33. ALU->SP výsledek do A reg
34. ALU->SP +1,-1 při PUSH,POP
35. ALU->PC inkrement PC pro další instr.
36. ALU->MBR výsledek do PAM
37. ALU->D výsledek do D reg

KOMPLEMENTY A POSUNY

28. Komplement L oper.
29. Komplement R oper.
30. Posun doleva 1b
31. Posun doprava 1b

ČTENÍ/ZÁPIS Z CÍLOVÉ PAMĚTI

38. PAM->MBR čtení z PAM (adr MAR)
39. MBR->PAM zápis do PAM(adr MAR)

Mikroprogramování

Používá se pro návrh automatů, popsaných mikroprogramem (mikroprogramové automaty).

- bohatá vstupní abeceda (10b – 1024 písmen/podmínek)
- malý počet větvení (v zásadě pouze 2 následné cesty)
- velký počet vnitřních stavů

Obecný návrh dle teorie automatů (velmi složité), neminimalizuje se. Obvykle: kombinační část –ROM – bez optimalizace, omezit větvení pouze na 1, max. 2 podmínky. Firmware: základní naprogramování řadiče + dodávané doplňky. Dynamické mikroprogramování – nejenom ROM, ale i RAM.

Hlavní výhody mikroprogramování:

- vytvoření „libovolného“ strojového kódu nad prostředky
- doplnitelnost strojového kódu – nový firmware

Počítačový systém, technické a organizační prostředky

Technické prostředky

Přerušeni

Automatická reakce technických prostředků na asynchronní událost mimo i uvnitř procesoru (žádost o přerušeni).

Příjem přerušeni

Přechod na obslužný program (obsluha přerušeni)

Příčiny přerušeni

- chyba v programu (přetečení, dělení 0, chybný operační znak, porušení ochrany paměti, ...)
- vnější příčiny (reálný čas z časovače, operační panel, komunikační linky, ...)
- přerušeni z I/O (ukončení I/O operace nebo chyba)
- technické (kontrolní obvody systému - napájení, chyba procesoru nebo paměti, chyba periferie)

Přerušovací systém

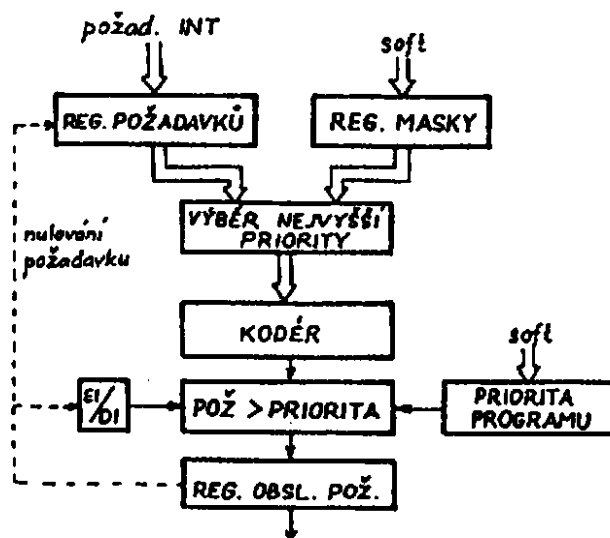
Technické i programové vybavení počítače, obsluhující žádosti o přerušeni. Zajišťuje jednotlivé fáze průběhu přerušeni. Přerušovací systém bývá vybaven možností měnit masku přerušeni. Různým událostem je přiřazena různá úroveň přerušeni. Možnost úplně vypnout/zapnout přerušovací systém – DI/EI.

Typy přerušeni

NMI – nemaskovatelné

INT – maskované (vektorové, nevektorové)

RESET – zákl. nulování a nastavení



Programové prostředky přerušovacího systému

Přijmutý požadavek způsobí přechod na obslužný program.

Nutné akce:

- detekovat zdroj přerušeni
- uchovat stav procesu pro pokračování
- předat řízení programu obsluhy
- zabezpečit možnost návratu

Organizační prostředky

V operační paměti jsou uloženy - supervizor, aktivní programy, systémové tabulky, standardní podprogramy. Počátek paměti je vyhrazen pro systémové prostředky.

Přidělování paměti uživateli

Statické

Každý uživatelský program má vyhrazenou pevnou oblast OP. Jednoduché, ale část paměti nemusí být využita.

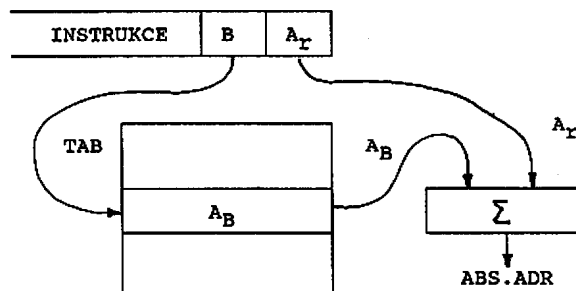
Dynamické

Přiděluje prostor podle okamžité potřeby programu.

Existují tři způsoby:

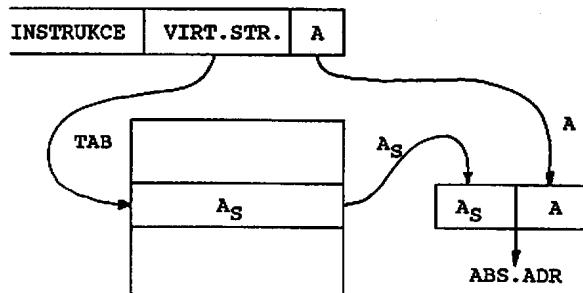
Bázové registry

Každý program má svoji básovou adresu. Básová adresa je v tabulce supervizora. Nevýhoda - spojitý blok paměti.



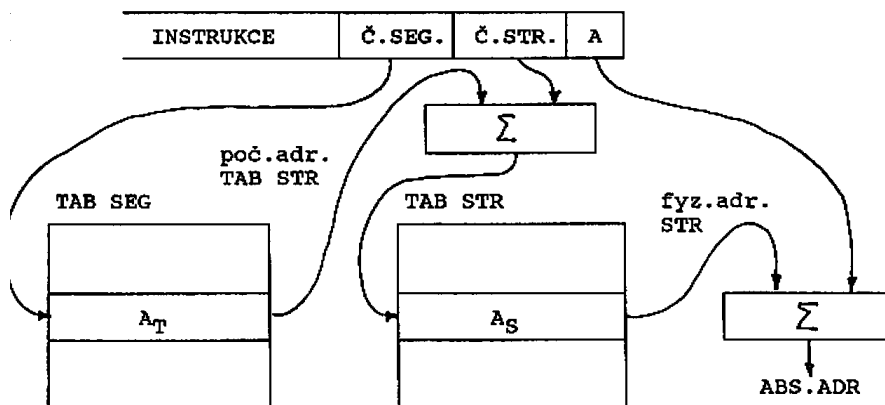
Stránková organizace

V tabulce má program vymezeny stránky, v instrukci je číslo virtuální stránky a adresa uvnitř stránky. Převod na fyzickou adresu se provádí přes tabulku.



Segmentová organizace

V instrukci 3 pole: číslo segmentu, číslo stránky, adresa uvnitř stránky. Vhodné pro složité programy.



Ochrana paměti

Zabezpečení informace v paměti proti náhodnému přepsání.

Charakter paměťových oblastí

- nepřístupné - systémové programy, řídicí programy.
- plně přístupné - i pro uživatele
- s povoleným čtením - překladače, ladící programy.

Obvody reálného času

RTC - Real Time Clock

Obvody RTC udržují v počítačovém systému aktuální datum a čas.

Sítě procesorů a počítačů

Požadavek zrychlení výpočtu při nutnosti obsluhovat rychlé procesy nebo zpracování rozsáhlých dat.

Počítačové struktury

Klasická – jednoduchý tok dat a instrukcí (SISD)

Procesorová pole – jednoduchý tok instrukcí a vícenásobný tok dat (SIMD)

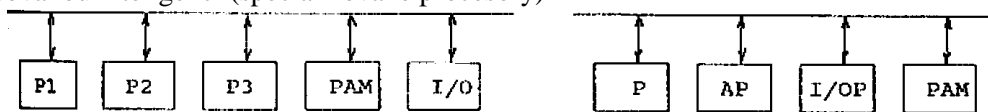
Pipeline – vícenásobný tok instrukcí, jednoduchý tok dat (MISD)

Multiprocesorové systémy – vícenásobný tok instrukcí a dat (MIMD)

Počítačové systémy

Jednopočítačový – centrální řízení

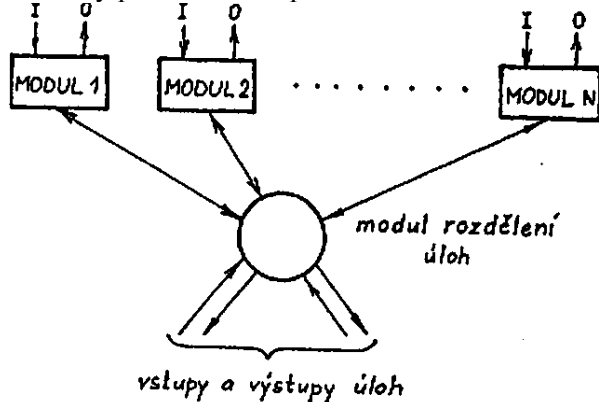
- multiprocesorový (všechny procesory identické)
- s distribuovanou inteligencí (specializované procesory)



Vícepočítačový – sítě počítačů

- multiprocesorový (všechny moduly identické)
- s distribuovanou inteligencí (specializované moduly)

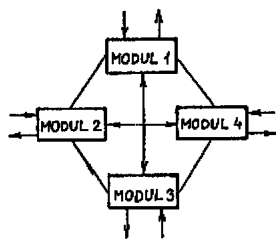
U vícepočítačových systémů má každý procesor svou paměť a I/O.



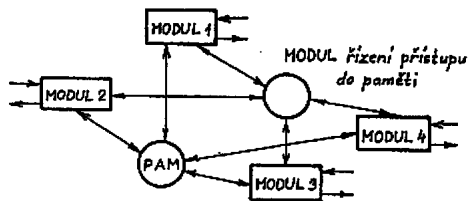
Uspořádání vícepočítačových systémů:

1. s rovnocennými moduly

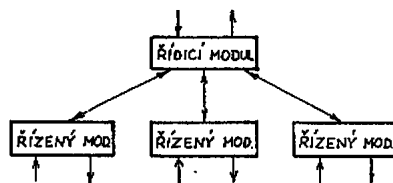
spolupráce na základě přenosu dat



spolupráce na základě společné paměti



2. s jedním řídicím a několika podřízenými

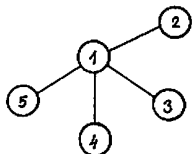


Topologie sítí, vlastnosti, použití

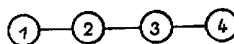
Stromové sítě – mají většinou střed sítě

Hvězdicová, lineární, vlastní stromová, větvená.

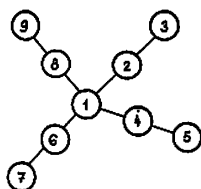
hvězdicová



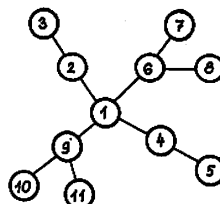
lineární



vlastní stromová



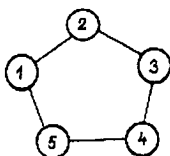
větvená



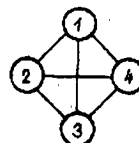
Ostatní - nemají střed sítě

Kruhová, polygonální.

kruhová



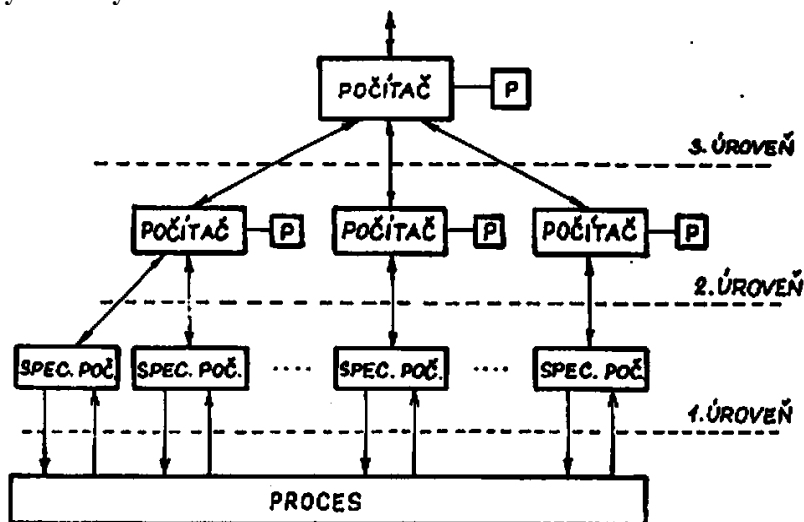
polgynální



Počítače propojené v síti dovolují vícestupňové řízení, větší počet řídicích zásahů současně. Menší zranitelnost decentralizovaného řízení, ekonomický výhodné sdílet periferie.

Použití

Př.: Hierarchický řídicí systém



Vyšší úroveň rozděljuje úlohy a hlídá vykonání spodní úrovně.

1. úroveň - spec. počítače - inteligentní řídicí moduly (reg. smyčky)
2. úroveň - vyšší řídicí funkce, změna chování 1 úrovně (konstanty regulátorů)
3. úroveň - globální funkce, kontrola spodních úrovní (optimalizace)

Literatura:

- Sylaby přednášek "Počítače pro řízení", PCX ze školní sítě, 1995/6