

NAVROVÁNÍ SYSTÉMŮ PRACUJÍCÍCH V REÁLNÉM ČASE NA MIKROPOČÍTAČÍCH

Ing. Antonín Kubalec

I. Volba stupně univerzálnosti real-time systémů

Důležitým hlediskem při návrhu řídicího systému je správná volba stupně univerzálnosti. Tato volba patří k základním koncepčním úvahám a může mít velký vliv na případný úspěch nebo neúspěch celého systému.

Při volbě stupně obecnosti máme možnost zvolit:

- a) univerzální systém
- b) systém pro konkrétní aplikaci

ad a) volbou univerzálnosti rozumíme schopnost systému pokrýt co největší množinu řídicích systémů. Jaké výhody a nevýhody nám tato volba přináší:

Výhody:

zvolím-li vysokou úroveň univerzálnosti, pak nám tento systém při opakovaném použití přináší značné úspory. Jeho implementace na daný řídicí systém je rychlejší, konkrétní aplikace může být generována pomocí parametrů, řídicí algoritmy mohou být sestavovány pomocí problémově orientovaných překladačů rovněž displejové obrazy, protokoly, alarmní výstupy (tak jak tomu je např. u univerzálního systému MODUS pro řízení a sledování technologických procesů - viz sborník PROGRAMOVÁNÍ 85). V systému tohoto typu, pokud je odladěn a správně sestaven, je mnohem méně možností dopustit se chyby při generování pro danou aplikaci. Ten, kdo systém nashazuje v praxi nemusí podrobně znát vnitřní strukturu systému, nemusí zvládat problematiku reálného času, technických prostředků, programování a problémy složitosti při sestavování takového systému. Stačí, aby pochopil funkci systému, který ovládá pomocí generačních parametrů a spouštěcích rutin a tím se mohl soustředit na řešení problémů vlastní aplikace a procesního řízení.

Nevýhody:

tento systém je mnohem obtížnější zkonstruovat. Vyžaduje to jednak mít dostatek času a finančních prostředků, ale taky tým pracovníků, kteří mají zkušenosti s podobnými systémy (třeba i s jednoduššími). Tento tým pracovníků by se měl skládat z programátorů, analytiků, koncepčních pracovníků, je potřeba, aby v takovém týmu pracoval dobrý organizátor a řídicí pracovník. Ten by odborně orientované specialisty na problematiku řízení, programování a analýzy vedl správným směrem, aby se jejich řešení nerozcházelo. Sestavit dohromady takový tým může být velký problém.

Univerzální systém je jednou odlačen s parametry definovanými v zadání. Změny v průběhu řešení s nebo po ukončení jsou náročné, ne-li nemožné. Systém univerzální (a tím i velké mohutnosti) je málo pružný na změny. Po zkušenostech se systémem MODUS se ukázalo, že i univerzální systém nepokryje veškeré požadavky kladené na řídicí systém. Pak je nutné vytvářet speciální uživatelské úlohy, které pokrývají požadavky uživatele. Tyto uživatelské úlohy nemusí do systému organicky "zapadat."

Konkrétní aplikace univerzálního systému málokdy využije všech možností, které univerzální systém nabízí. V některých případech se může stát, že se redundance stane přítěží.

a) systém pro konkrétní aplikaci

tento přístup k tvorbě řídicího systému přináší:

Výhody:

jeho složitost a náročnost nebude tak vysoká jako u univerzálního systému, nebude mít tak náročné požadavky na tým stabilizovaných, kvalifikovaných pracovníků. Jeho velkou předností bude, že se celá koncepce může plně přizpůsobit požadavkům uživatele, systém bude "šitý na míru". Tento systém bude svým rozsahem menší, což znamená, že bude snadněji laditelný, přehlednější i pro méně zasvěcené odborníky. Dále v tomto systému nebudou tak aktuální otázky kapacitní a časové.

Nevýhody:

pro každou aplikaci je třeba vždy řešit nejen problémy procesního řízení, ale i problémy analýzy, programování a ladění nových úloh, které je třeba s každou novou aplikací buď znovu napsat, nebo přizpůsobit starou úlohu novým požadavkům. V každém případě to vyžaduje účast analytiků a programátorů u každé aplikace, což u univerzálního systému není nutné. Je třeba znovu řešit problémy vzniklé novou programovou konfigurací a u každé aplikace je znovu ladit. Právě tato část, ladění systému jako celku, je nejnáročnější činností a je třeba jí u každé aplikace opakovat.

Při volbě stupně obecnosti řídicího systému hraje taky velkou roli počet nasezení na daných technických prostředcích. Toto hledisko může převážít výhody určitého typu řídicího systému třeba v jeho neprospěch. Dále to mohou být i hlediska realizační: nedostatek řešitelských kapacit, času, finančních prostředků; tyto faktory nechť má na zvažení koncepční pracovník v oblasti tvorby řídicích systémů.

II. Technické prostředky

Zda často není velká možnost volby: Situace je většinou taková, že jsou k dispozici určité prostředky a je jim třeba přizpůsobit programové vybavení. Technické prostředky dávají tato hlavní omezení:

- a) časová
- b) kapacitní

ad a) časové omezení je dané operační rychlostí počítače, rychlostí periférií. Při návrhu je třeba odhadovat, jak bude systém rychlý na daných technických prostředcích. Tento odhad se provádí obtížně, ale platí (ne sice v přímé závislosti) čím větší (rozsáhlejší) systém, tím pomalejší. Hraje zde roli hlavně množství cyklických úloh (komunikační interpretační, alarmní úlohy) a s jakou periodou jsou spouštěny. Tyto odhady hrají velkou roli u systémů implementovaných na malých počítačích (8-bitové mikropočítače)

ad b) kapacitní omezení je dané kapacitou hlevní paměti, vnějších pamětí, periferních jednotek (velikost, počet displejů, tiskáren), rozsahy vstupně-výstupních modulů. Tato omezení se dají určitými způsoby řešit (na rozdíl od časových) jako je tomu např. u 8-bitového mikropočítače SM 53/10, kde je problém kapacity operační paměti vyřešen pomocí stránkované paměti (celková kapacita 260 kB).

Kapacita vstupně-výstupních modulů se dá řešit poměrně schůdnou cestou. Při překročení kapacity modulu se do konfigurace zařadí další modul až do vyčerpání kapacity mikropočítače, pak je možnost zařadit do sítě další mikropočítač.

III. Projektant a analytik (programátor) řídicího systému

Při tvorbě řídicího systému se uplatní některé odlišné profesní činnosti. Jedno z možných dělení těchto činností je na:

- návrh a realizace řídicí části
- návrh a realizace programové části

První činnost v sobě obsahuje návrh řídicích algoritmů na základě požadavků zadavatele, návrh technických prostředků, se kterými se bude řízení realizovat. Tuto činnost by měl pokrývat projektant řídicího systému.

Druhá činnost v sobě obsahuje práce spojení s implementací řídicích algoritmů na počítač.

Protože tyto 2 činnosti na sebe navazují, je třeba definovat rozhraní mezi nimi. Toto rozhraní bude např. dáno prostředky a jakými bude projektant pracovat, a které mu analytik (programátor) vytvoří. Jak by měly tyto prostředky vypadat a na jakém stupni obecnosti budou - to závisí na požadavcích projektanta, na omezeních počítače, na řešitelských kapacitách. Mělo by se vyjít z požadavků, které by si projektant definoval sám. Měly by to být prostředky takové, které by odpovídaly jeho představám, také, aby je dokázal zvládnout, aby se mu s nimi dobře pracovalo. Tyto prostředky mu vytvoří analytik (programátor) řídicího systému.

Otázka je, jak mají tyto prostředky vypadat:

- má to být sada algoritmických modulů s mechanismem jejich spouštění, které si projektant bude spojovat do výsledného systému podle potřeb generované aplikace. To znamená dát projektantovi prostředek nízké úrovně obecnosti, který umožní výrazně zasáhnout do systému a vytvořit pro každou aplikaci poměrně optimální systém (systém pro konkrétní aplikaci).
- má to být prostředek vysoké úrovně obecnosti, univerzální systém velkého rozsahu, které bude projektant ovládat pomocí parametrů. Tím se umožní projektantovi, aby se soustředil na problémy řízení a nebyla jeho pozornost odváděna problémy související s implementací na počítač.

Můžeme říci, že nízká úroveň obecnosti prostředků pro tvorbu řídicích systémů do jisté míry usnadňuje práci analytikovi (programátorovi) a klade vyšší nároky na práci projektanta. Vyžaduje od projektanta znalosti operačního systému, problematiku reálného času, znalosti technických prostředků. Zda je sto tento celý rozsah projektant zvládnout, nebo zda je účelné, aby se touto problematikou zabýval, to by mělo být předmětem úvah před zahájením budování řídicích systémů. Snad ideální by byl takový prostředek, který nemá tuto hranici úrovně obecnosti pevně stanovenou a lze ji podle potřeby posuzovat směrem k vyšší nebo nižší úrovni.

IV. Operační systémy mikropočítačů

Operační systém vystupuje v roli prostředníka mezi uživatelskou úlohou a technickými prostředky, na kterých je uživatelská úloha realizována.

Pokud se zaměříme na mikropočítače, pak operační systémy 8-bitových mikropočítačů, především těch, které pracují v reálném čase, nejsou tak složité jako operační systémy počítačů vyššího typu.

Tyto operační systémy nemají velké možnosti ochrany proti přepisu, proti nesprávnému používání (zadávaní nesprávných parametrů při volání rutin jádra, ...). Tato jednoduchoost je však

v případě 8-bitových počítačů nutnosti, a má výhodu v tom, že se operační systém zaměřuje na vlastní vykonávání funkce se sňahou o minimální časovou a paměťovou režii.

Operační systémy 16-ti bitových mikropočítačů odpovídají svou mohutností technickým prostředkům, na kterých pracují. Pro činnost řídicích systémů je potřeba užít operační systém podporující systémy pracující v reálném čase. Takovým operačním systémem je ERČ/86. Liší se od ostatních tím, že není generovaný pevně, ale je navržen tak, aby si ho mohl uživatel generovat podle potřeby. ERČ/86 obsahuje soubor podsystémů, z kterých každý poskytuje jednu nebo více služeb. Kombinací vybraných podsystémů se potom vytvoří operační systém na míru, který odpovídá požadavkům uživatele. Cílem ERČ/86 je zkrácení času a nákladů na vývoj a řešení aplikačního programového vybavení. Vzhledem na množství služeb, které má ERČ/86 k dispozici, může být použitý v množství aplikačních systémů, tím umožňuje uživateli soustředit se na řešení svých problémů.

Další problém, který nelze přehlédnout je to, že řídicí systém bývá realizován na síti počítačů. Operační systémy, které jsou vyvinuty pro práci v reálném čase, jsou jednopočítačové.

Tento nedostatek je řešen nadstavbovými operačními systémy, např. MAGDOS pro SM 53/10, který však nepokrývá plně všechny požadavky uživatele (programátora řídicího systému). Tento nadstavbový operační systém sice dává k dispozici základní funkce mezi-terminálových vazeb, ale nemá charakter globálního operačního systému, tak jak by to síť počítačů potřebovala. Globální operační systém sítě by měl jednoduchým způsobem umožňovat synchronizaci a výměnu dat mezi úlohami umístěnými v různých terminálech. Např. některé úlohy procesního řízení (na terminálu připojeného k procesu) připravují data pro operátorský terminál. Má smysl přenášet tato data na operátorský terminál až v okamžiku, kdy jsou procesním terminálem zpracována a zpracovat je na operátorském terminálu až v okamžiku, kdy jsou přenesena. V současném systému (MODUS pro SM 53/10) jsou tyto synchronizace řešeny ručním nastavováním časových parametrů (periody spouštění cyklických úloh). Bylo by vhodné, aby pro řešení podobných synchronizačních problémů byl k dispozici operační systém.

V. Přenositelnosti řídicích systémů

Pokud se při budování řídicího systému zvolí cesta univerzálního velkého systému, pak je otázka přenositelnosti tohoto programového celku na jiné technické prostředky aktuální. Na druhé straně však přenositelnost systému pracujícího v reálném čase je považována za jednu z nejhůře přenositelnou (zejména části, které se vztahují k obaluze vnějších zařízení).

Pojmu přenositelnosti můžeme rozumět jako přizpůsobivosti programů novým technickým prostředkům a operačním systémům. Základním problémem je najít vhodný jazyk, tj. kompilátor, který bude nové vazby zabezpečovat. Měl by to být kompilátor dostatečně vysoké úrovně, aby nebyl ovlivňován změnami v architektuře technických prostředků. V současné době se jeví jako nejvýhodnější C-jazyk. Tento jazyk byl speciálně vyvinut pro přenesení programů (u Bell Laboratories) včetně systémových pro základní prostředí PDP 11 s operačním systémem UNIX. Otázkou je, jak je tento jazyk vhodný pro zápis programů pracujících v reálném čase, které vyžadují speciální real-time konstrukty, s které tento jazyk nemá.