

PROJEKTOVÁNÍ A PROGRAMOVÁNÍ SYSTÉMŮ PRACUJÍCÍCH V REÁLNÉM ČASE

Ing. Antonín Kubalec

Kancelářské stroje, k.ú.o. závod Ostrava, Řídící systémy

1. Způsob definování algoritmů pomocí formulačního jazyka

Na našem pracovišti se používá pro definování algoritmů formulační jazyk jako jazyk analytika. V současné době slouží tento zápis jako podklad pro programování v jazyce assembler, nebo PL/M (nahrazuje tímto vývojové diagramy). Vývoj tohoto jazyka směřuje k tvorbě překladače, takže zápis v tomto jazyce by byl výsledný přeložitelný produkt analytika (programátora).

Systém MODUS (mikropočítačový univerzální distribuovaný systém pro řízení a sledování technologických procesů), který byl vytvořen na našem pracovišti je nyní realizovaný na 8-bitovém mikropočítači a všechny jeho rezidentní úlohy jsou zapsány v assembleru (což je nutnost z pamětového i časového hlediska). Přeložený program zapsaný ve formulačním jazyce by jistě nebyl neúspěšnější, takže překlad formulačního jazyka není řešením pro MODUS na 8-bitovém mikropočítači.

Formulační jazyk používaný u nás má v sobě kromě běžných instrukčních a definičních příkazů speciální real-time konstrukty (odesílání zpráv, čekání na zprávy, aktivace procesů atd). Je tedy určen pro systémy pracujících v reálném čase.

2. Stručná charakteristika systému MODUS

Systém MODUS je realizován na 8-bitovém mikropočítači SM 53/10, což je síť terminálů (až 64 v jedné sestavě). SM 53/10 je sestaven ze 2 typů terminálů - jsou to TSP - terminál styku s procesem v tomto hardwarovém osazení:

- základní modul (procesor)
- paměťový modul (64 kB)
- autodiagnostický modul
- komunikační modul
- volitelné V/V moduly (analogové, číslicové převodníky)
- matematický procesor (nepovinný)

potom TOP - terminál operátora s jednotkou pružných disků, semi-grafickým displejem, tiskárnou.

TOP má toto hardwarové osazení:

- základní modul (procesor)
- paměťový modul (pro operační systém RAPOS i více modulů)
- autodiagnostický modul
- komunikační modul
- matematický procesor
- obsluha disků (2 moduly)
- modul obsluhy semigrafického displeje

Systém MODUS je programovým nástrojem pro řídicí a automatizační techniku. Pomocí něho mohu sestavovat jednoduché elementární moduly (ovládání regulátorů, přepínače, moduly převodu analogového signálu na číslicový atd), které provádějí řízení nebo sledování procesu. Tyto moduly se sestavují do sítí, které pak řídí celé technologické procesy. K tomu nám slouží část MODUSu nazvaná překladač MODUS/P. Pomocí něho interaktivním způsobem zadávám, které moduly se budou řízení účastnit a jak budou v síti zapojeny.

Dále si mohu pomocí překladače MODUS/O vytvářet sledovací obrazy operátora. Skládají se ze statického pozadí (vytvořeného pomocí kreslicího programu) a dynamického popředí. Takto připravované sítě a sledovací obrazy se pak spouštějí interpretační částí MODUSu na žádost operátora.

Překladače MODUS/P, MODUS/O pracují v ON-LINE systému, to znamená, že mohu sestavovat nové sítě a vytvářet nové sledovací obrazy, aniž by se přerušil řídicí proces. Moduly sestavené do sítě pracují na terminálu typu TSP, sledovací obrazy běží na terminálu typu TCP. Sítě pracující v TSP pracují v autonomním režimu (není nutné, aby pracoval TOP).

3. Přenositelnost real-time systémů na jiné technické prostředky

Při přenosu systému na jiné technické prostředky záleží především na tom, jak se liší původní technické prostředky od těch, na které systém přenášíme.

Z hlediska projektování pak vzniká problém

- navrhnout systém obecně pojatý, který je snadněji přenositelný na jiný počítač, na druhé straně je jeho realizace o to

obtížnější, rozsah větší

- navrhnout systém specifický pro určitý problém, pro dané technické prostředky a na určitě základní programové vybavení realizace snadnější, rozsah menší

Z toho hlediska by měl projektant zvážit své cíle a možnosti.

V současné době se přenáší MODUS na systém CIRIS 8000, což je 8-bitový mikropočítač postavený na bázi MFB 8080. Má stejnou instrukční množinu jako SM 53/10, dá se na něm provozovat stejný operační systém, obaluka disků je stejná. Liší se v displeji - systém CIRIS má alfanumerický displej s vlastním firm-ware, jiný druh komunikace. Převod MODUSu na systém CIRIS spočívá hlavně v úpravě obaluky odlišných periférií. Základ systému může zůstat beze změny.

V budoucnu se předpokládá převod MODUSu na některý 16-bitový mikropočítač. Bude to počítač s jinou instrukční množinou, jiným základním programovým vybavením. Zde se projeví i to, jak nezávisle na tech. prostředcích je systém navržen a zapsán. Pokud existuje zápis jen v assembleru, pak převod z 8-bitového počítače na 16. bitový znamená vše přeprogramovat. Pokud je zapsán ve vyšším programovacím jazyku (např. ve formulačním jazyku), pak existují i jiné možnosti převodu.

Při převodu tohoto typu vyvstává ještě otázka pro projektanta, zda nebude vhodnější řešit některé problémy jiným způsobem na počítači, který nám dává větší možnosti (větší rychlost, kapacita paměti).

4. Stručná charakteristika operačních systémů používaných MODUSEm

Systém MODUS pracuje pod operačním systémem RAPOS (operační systém pracující s rozšířenou pamětí) a MAGDOS (malý generovatelný distribuovaný operační systém). Oba jsou postaveny na původním operačním systému ERČ (exekutiva reálného času). Pro potřeby SM 53/10 byl tento operační systém rozšířen o některé funkce a nazván MAGDOS. Tento rozšiřuje ERČ o

- obaluku komunikačního modulu
- obaluku nemigrafického displeje
- časové úlohy (zadávání času, animání času, udržování centrálního

času)

- obsluha rozšířeného přerušovacího systému
- program TRANSIT pro zavádění systémů do terminálů, které nemají připojeny diskové jednotky (typ terminálů TSP)

Průběh řešení si vynutil vývoj operačního systému RAPOS, který je nadstavbou oper. systému MAGDOS, má všechny jeho funkce a navíc nám umožňuje práci s rozšířenou pamětí. Tento operační systém vyžaduje úpravu technických prostředků (realizována na auto-diagnostickém modulu).

Paměť u SM 53/10 byla rozdělena následující způsobem.

- a) společná paměť ROM - 4 kB
- b) stránkovaná paměť ROM - 3 stránek po 28 k Bytech
- c) společná paměť RAM - 32 kB

Toto představuje využitelnou paměť v rozsahu

$$4 + 8 \quad 28 + 32 = 260 \text{ kB}$$

Fyzicky nám manipulaci s pamětovými stránkami zabezpečuje registr aktuální stránky a registr předcházející stránky. Stránky jsou stínové paměti, pokud adresujeme do stránkové oblasti, konkrétní pamětové místo závisí na hodnotě registru aktuální stránky.

5. Zkušenosti získané při navrhování a realizaci systému MODUS

MODUS lze charakterizovat jako komfortní systém, který si tím klade na 8-bitový mikropočítač extrémní požadavky. Proto by měl projektant systému zvážit nejen softwarová hlediska (široké možnosti systému, přenositelnost, bezpečnost), ale i omezení vyplývající z typu počítače. Omezující faktory jsou kapacita paměti a výpočetní rychlost.

Při řešení systému MODUS nám problém nedostačující paměti vyřešil RAPOS.

Časová omezení nám částečně řeší matematický procesor. Při zařazení do systému zvýšil při interpretaci celkovou výpočetní rychlost zhruba o 1/5 až 1/4, záleží na tom, jaké sledovací obrazy běžely (např. zobrazování grafů vyžaduje mnoho matemat. operací) a jaké moduly byly zařazeny do sítě (jak využívají mat. procesor). Nepodařilo se však při přechodu na matematický

procesor zkrátit kód. Dále matematický procesor dokáže provádět operace sčítání, odečítání, násobení, dělení, umocňování ..., ale neumí provádět čísla z ASCII-tvaru do vnitřního tvaru v pohyblivé řádové čarce a naopak. Těmto operacím jsou nejnáročnější a nejčastěji používané a bohužel musí být řešeny programově.

Další cesta pro zlepšení časových poměrů je optimalizace (zkracování) kódu např. přeprogramováním částí zapsaných v jazyce PLM 80 do ASM 80.

Diskové operace 8-bitových mikropočítačů jsou velmi pomalé a náročné na využívání procesoru. Existuje způsob (zatím nerealizovaný), jak tyto operace zrychlit. Při těchto úpravách vyjde ze současného stavu. Při konfigurování systému pracujícím pod ERČ si mohou zvolit počet bufferů přidělených diskovým operacím. V MODUSu byl zvolen jeden, a to z toho důvodu, aby nedocházelo k rozpracovanosti souborů na disku (způsobovalo to vyčerpání volné paměti). V takovém případě nepotřebují pracovat s obsluhou diskových souborů původního ERČ, ale mohou pracovat přímo s fyzickým ovládním disků.

Časové poměry v systému mohou ovlivnit i při vlastním generování systému MODUS. Jako systém pracující v reálném čase obsahuje množství úloh, které se cyklicky aktivují. Těmto úlohám při generování nastavují periody spuštění, čímž mohou výrazně ovlivnit celkovou výpočetní rychlost systému. Při špatném nastavení period mohou systém zatížit na nevhodném místě.

Terminál (TOP) systému SM 53/10 má 2 diskové jednotky, v nichž jsou použity disky jednoduché hustoty s kapacitou 256 kB. Na nich jsou uchovány datové soubory operátorských obrazů (protokolů), obrazy RAM paměti do TOP a TSP, archivní údaje a celá nerezidentní část MODUSu. Zde se nabízí cesta ke zvýšení rychlosti systému v umístění často používaných datových souborů do rozšířeného pamětového prostoru.

6. Způsob a techniky ladění systému pracujícího v reálném čase

Ladění systému pracujícího v reálném čase patří k nejnáročnějším etapám budování systému.

Při generování systému se vychází z konfiguračního modulu,

kteřím jsou definovány úlohy (tasky) zařazené do systému. Konfigurační modul vytváří tabulky, se kterými pak pracuje jádro operačního systému. Do konfiguračního modulu zadávám údaje délky zásobníku, priority úloh, ochránky. Už zde se mohou dopustit chyby např. zadáním malé délky zásobníku nebo špatné priority - priority udává taky pořadí, v jakém budou úlohy při inicializaci spuštěny.

Při budování nového systému často dělá potíže odladění této inicializační fáze (někdy právě z neznalosti základního programového vybavení - problém černé ochránky). Proto je vhodné pracovat na konfigurování systému a na statickém ladění jednotlivých aplikačních úloh souběžně.

Po fázi statického odladění úlohy (systém vstupy - výstupy), zařazují úlohu do celého systému a ladím ji dynamicky. Úloha musí být navržena a realizována tak, aby nebránila ohodu celého systému. Za příčiny dynamických chyb mohu považovat:

- a) nedovolený přepis paměti (např. kódu jiné úlohy) - chybnou programovou realizací
- b) špatné využívání základního programového vybavení - chybnou programovou realizací nebo analytickým návrhem
- c) špatnou návazností na ostatní úlohy - chybným analytickým návrhem
- d) zahlcení systému nastavením vysoké frekvence spuštění úloze s vysokou prioritou - chybným analytickým návrhem a generováním systému
- e) špatné využívání paměti (volná paměť = dynamicky využívaná RAM paměť - při potřebě vyžádán - po skončení potřeby vrátím) - chybným analytickým návrhem

Ze dynamickou chybu nebo nedostatek se považuje vše, co zbytečně časově zatěžuje systém. Příčiny dynamických chyb se mohou v jednotlivých systémech lišit. Těžko je lze definovat obecně.

Na operátorském terminálu (TOP) nemá např. smysl úloze, která aktualizuje živé hodnoty do displejových obrazů, dávat periodu spuštění desetiny vteřiny, když přenosová úloha, která tyto hodnoty přenáší z TSP má periodu spuštění nastavenou na 3 - 4 vteřiny.

Není vhodné při práci s diskem číst soubor po částech. ERČ má velkou časovou režii na otevírání, zavírání souborů. (Toto platí především pro 8-bitové mikropočítače).

Na terminálu styku s procesem mohou rovněž nevhodně zatížit systém periodou spouštění sítí. Toto je však již v rukou uživatele systému MODUS, kterému jsou periody spouštění sítí uživatelsky přístupné.

Součástí základního programového vybavení ERČ je ladící program. Tento však vyžaduje 11 - 12 kB rezidentní paměti, kterou mu při ladění MODUSu nebylo možno přidělit. Proto byl MODUS odladěn pomocí emulátoru, který je silným nástrojem pro ladění, ale je obecný. Nemohu si pomocí něj jedním příkazem vypisovat speciální tabulky ERČ, způsob krokování programů je náročnější.

Pomocí emulátoru mohu vypsat posledních 44 instrukčních cyklů zpět od mé zářezky, což je pro ladění rozsáhlého systému málo. Mnohdy se podaří systém zastavit až v okamžiku, kdy je už příliš daleko od místa vzniku havárie, je často přepsaná paměť, takže nejsem schopen ani určit úlohu, která havárii způsobila. Nalezení místa vzniku havárie pak záleží na zkušenostech a intuici programátora.

Při ladění systému v těchto podmínkách je třeba postupovat opatrně. Některé zásady pro práci se systémem:

- a) do systému zařazovat úlohy postupně (nejlépe po jedné)
- b) nedělat v hotovém systému mnoho oprav najednou
(nejsem pak schopen zjistit, která oprava havárii způsobila)
- c) nechat si při časovém rozvrhu ladění asi 1/3 času na ladění systému jako celku
- d) vyhýbat se zbytečným rekonfiguracím systému
(týká se to úprav konfiguračního modulu)

8. Zkušenosti s technickými prostředky 8-bitových mikropočítačů

Při ladění systému je často obtížné určit, zda systém havaruje z důvodu chyby v programovém vybavení nebo vinou technických prostředků (zvláště pokud se jedná o prototypy, kde se chyby dají předpokládat). Zde nemá programátor lehkou úlohu, protože pokud

označí chybu jako programovou, může hledání této chyby trvat dlouhou dobu (i několik týdnů) a pak zjištění, že to byla chyba v technice, může být t rdým probuzením. Proto by si měl programátor umět vyloučit chybu v technických prostředcích. Může mu k tomu posloužit již hotový odladěný systém, přibližně stejné složitosti. Nemí dobré se v těchto situacích spoléhat na technické testy. Ty nám prověří určité specifické zařízení, ale neotestují počítač jako celek v dynamických podmínkách.

Při ladění systému MODUS jsme naráželi nejčastěji na tyto závady v technických prostředcích:

a) chyby disků

- nekompatibilita diskových jednotek
- vada řadiče disků
- vadná diskety

b) chyby v komunikaci

- vadným zapojením mikropočítačů na magistrale (koaxiální kabel), především v konektorech - komunikační modul
- vadou na komunikačním modulu

c) chyby emulátoru

- propojovací kabely prototyp - emulátor, většinou chybnou manipulací
- emulátor může způsobovat chyby tím, že narušuje (při zadávání složitých krokovacích příkazů) časové poměry realtime systému. Pokud systém přímo nehavaruje, tak určitě pracuje pomaleji.

d) chyby paměti

- poměrně snadno zjistitelná závada. Při natahování systému emulátorem provádí se zpětná kontrola zapisovaného údaje, pokud zapsaný obsah nesouhlasí se skutečným, emulátor hlásí chybu.

Závěry

Závěrem bych chtěl říci, že téma přípravku je velice široké a nemám dosud takový přehled, abych mohl vybrat z této oblasti právě to, co je v současné době projektanty (programátory) real-time systémů i jiných systémů nejaktuálnější. Proto jsem vybral ty problémy, na které jsme narazili při budování MODUSu my.

Velkým problémem, kterým se projektant nebo programátor musí zabývat, je přenositelnost vybudovaného systému na jiný počítač. Tato skutečnost nesvědčí o jedné dobré věci, a to že již dnes existuje napřeberné množství druhů mikropočítačů.

Nebylo by sice správné chtít pouze jednu unifikovanou řadu počítačů, protože existují nejrůznější oblasti aplikací mikropočítačů v praxi. Nezdá se mi však taky správné vyvíjet znovu totéž (viz příklad 8-bitového mikropočítače MUB-SM 50/40)

Přehled literatury

- Exekutiva reálného času pro SM 50/40
vydalo ZP ČSVTS při Výzkumném ústavu výpočetní techniky
v Žilině - 1982
- MAGDOS (malý generovatelný distribuovaný operační systém
(uživatelská příručka) - 1982
- ERČ - RAPOS (uživatelská příručka)
vydaly Kancelářské stroje k.ú.o. závod Ostrava,
Řídicí systémy