V Y T V Á Ř E N Í U Ž I V A T E L S K Ý C H P R O G R A M Ů V A S S E M B L E R U

J.Duda, K.Král, M.Zeman

V článku jsou diskutovány zvláštnosti v metodice projektování a realizace programů s využitím jazyka assembler na mikropočítačích. Dílčí zkušenosti jsou čerpány ze dvou odlišných aplikací, jedné v oblasti ASŘ technologických procesů, druhé v oblasti ASŘ podniku. Jsou uvedeny některé zvláštnosti vytváření programů pro každou z těchto oblastí a nejdůležitější chyby, ke kterým v obou aplikacích došlo. Hlavní pozornost je věnována zásadám, které je nutno při využívání jazyku assembler zvlášť pečlivě dodržovat, aby práce probíhaly dostatečně rychle a programování bylo možno rozdělit mezi více pracovníků.

1. Programování v assembleru

Assembler, správněji jazyk symbolických adres /dále pouze JSA/, je nejnižším prakticky používaným programovacím jazykem /3/. Většina mikropočítačů, vybudovaná na bázi 8bitových mikroprocesorů, vyráběná nebo dovážená v současnosti do ČSSR, je vybavena pouze interpreterem Basicu a překladačem JSA /vývojové systémy též překladačem PL/M/. Basic nevyhovuje pro řadu aplikací svými značnými nároky na čas zpracování a paměť mikropočítače a také tím, že obvykle neumožňuje ošetřování paralelních procesů /sdílení času/. To jsou zřejmě nejčastější důvody používání JSA pro tvorbu uživatelských programů. K přednostem JSA patří i možnost pracovat s libovolnými datovými strukturami a použít optimálního způsobu synchronizace paralelních úloh. Je-li přístupná potřebná dokumentace, umožňuje plně využívat hardware a ošetřovat přerušovací systém mikropočítače. Stejné možnosti mají ovšem i jiné jazyky, například PL/M. Za výhodu lze považovat také ladění programů krokováním po strojových instrukcích a možnost provádět menší opravy zavedeného programu přímo v paměti bez nutnosti opravovat zdrojový modul a provádět znovu překlad, sestavení atd.

Největším problémem programování v JSA jsou značné nároky na kapacitu programátorů i analytiků /projektantů/. Při rozpracovávání programového zadání je nutno dbát na to, aby se opako-

vaně neprogramovaly podobné úlohy, nevytvářely funkčně téměř shodné podprogramy a macra. Zatímco ve vyšším programovacím jazyku se lze soustředit na věcnou problematiku, při využívání JSA se musí mnohem více energie věnovat technice programování. To vyžaduje dlouhodobější cvik, rozsáhlejší programy by měl dělat programátor spocialista, který není zatěžován věcnou problematikou projektu. Programování v JSA často vyžaduje také širší znalost hardware mikropočítače než většina ostatních jazyků. Čitelnost zdrojového programu je více než u jiných jazyků podmíněna srozumitelnými komentáři a vhodnou dokumentací, např. vývojovými diagramy. Překladače JSA provádějí pouze kontrolu syntaxe jednotlivých příkazů, za vše ostatní zodpovídá programátor.

2. Vytváření programů pro ASŘ TP

Úlohy řídící technologické procesy obyčejně pracují v reálném čase a musí u nich být většinou zaručena určitá doba odezvy na každý stav řízeného systému. Často se jedná o více paralelních úloh, které sdílí řadu hardwarových i softwarových prostředků. U řídícího systému bývají kladeny vysoké nároky na spolehlivost, přičemž však ztráta menšího množství dat nebývá v některých případech kritická. Zařízení by mělo být schopno ošetřit všechny nepředvídané situace, důležité jsou autodiagnostické schopnosti a automatické obnovení činnosti po výpadku napájení. Pro řízení technologických procesů jsou charakteristické menší rozsahy zpracovávaných dat, ale poměrně vysoké nároky na vlastní výpočty. Hardware řídícího systému je často jednoúčelové zařízení, vytvořené nebo přizpůsobené podle konkrétních požadavků dané aplikace. Programy bývají odlaďovány na vývojovém systému nebo alespoň s jeho pomocí.

3. Vytváření programů pro ASŘ P

Řídící systémy v oblasti ASŘ P pracují obyčejně v dávkovém nebo interaktivním režimu s nižšími nároky na rychlost odezvy a spolehlivost. Většina dat vstupuje z klávesnice a vystupuje na obrazovce nebo ve formě tiskových sestav. Pro tyto řídící systémy je charakteristické velké množství zpracovávaných dat, přičemž jsou kladeny vysoké požadavky na jejich zabezpečení proti ztrátě nebo poškození. Základem bývá vytvoření vhodného systému pro sběr

dat s účinnými formálními i logickými kontrolami. Hlavním problémem je způsob třídění, organizace a uložení dat a zabezpečení přístupu k nim podle více klíčů. Programy bývají laděny přímo na cílovém systému, vzhledem k rozsahu projektů bývá nezbytná spolupráce více programátorů.

4. Některé chyby při programování v JSA

Níže uvedené zkušenosti jsou čerpány ze dvou odlišných aplikací. První byl experimentální řídící systém cukrovarnické odparky /1/, realizovaný na rozšířeném mikropočítači SDK-85 a SAPI80. Nejzávažnější chybou zde byly příliš vysoké funkční požadavky na činnost systému v počátečních fázích projektu, především nároky na rozšiřování a úpravy systémového software a obsluhu nestandardních technických prostředků. Problémem byla také omezená dostupnost a tím i možnost využívat standardní programové moduly a ne vždy správné rozhodnutí, zda upravit již hotový modul nebo vytvořit nový. Realizaci zpomalovala i snaha po maximální otevřenosti systému, která by umožňovala na něm experimentovat a rozšiřovat jej.

Druhou aplikací je systém pro odbytové činnosti v k.p.Spolana Neratovice /2/, realizovaný na mikropočítači VT-20 a počítači EC1032. Nedostatky v dokumentaci standardního software VT-20 byly tak značné, že bylo nutno všechny funkce prověřovat. Standardní programové vybavení neposkytovalo řadu nezbytných funkcí, chyběla například aritmetická knihovna, třídící program, systém pro práci s obrazovkami, vybavení pro práci s magnetickou páskou a řada dalších. Největším nedostatkem byla ovšem nízká projekční připravenost, porušení zásady provádění analýzy problémů zhora dolů a vytváření programů zdola nahoru mělo za následek časté a časově náročné úpravy a opravy již hotových programových modulů ještě před zahájením zkušebního provozu. Některé datové struktury nebyly vhodně definovány s ohledem na možnost náhodných výpadků mikropočítače, snaha zabezpečit data proti těmto výpadkům pak ztěžovala tvorbu algoritmů zpracování dat. V době zahájení prací byla podceněna časová náročnost vlastního programování v JSA. Situace pak byla ještě zhoršena stanovením pevných dílčích termínů pro vytvoření některých programových modulů, což vedlo ke snaze dokončit tyto moduly bez ohledu na následující části systému.

5. Osvědčený postup prací

Vytváření programů v JSA by mělo vždy předcházet detailní projekční příprava. Základem algoritmizace je definování datových struktur, přičemž je nutno brát ohled na zabezpečení dat proti nespolehlivosti hardware. Vlastnímu zakódování programů by mělo dále předcházet rozpracování algoritmů, například ve formě vývojových diagramů a v případě paralelních procesů též důkladné promyšlení všech časových vazeb mezi těmito procesy. Ve všech fázích projektování i realizace programů je nutno dbát na zásadu provádění analýzy problémů zhora dolů a programování zdola nahoru, při využívání JSA má nedodržení této zásady zvlášť nepříjemné následky. Uvedený postup obyčejně umožní rozdělení celého systému na poměrně nezávislé moduly,o jejichž tvorbu se podělí více programátorů, doba realizace se tím podstatně zkrátí. Důležitá je přesná definice rozhraní a dohoda na způsobech značení. Při tvorbě uživatelského systému musí být maximálně využito všech dostupných odladěných programových modulů a systémového software. Časová náročnost rozsáhlejších úprav systémového vybavení není pro jedinou aplikaci nikdy přijatelná, již ve fázi projektování by měly být funkční požadavky na systém přizpůsobeny technickým i programovým možnostem dostupné výpočetní techniky. Klíčové úseky projektu by měly být realizovány v předstihu. Při řízení projekčních a hlavně programátorských prací by se nemělo zapomínat na potřebu poskytnout analytikům a programátorům dostatek klidu k tvořivé technické práci, vyvolávání stresových situací obyčejně nic nevyřeší.

6. Závěr

Podkladem pro vytváření programových systémů musí být reálné plány. JSA by měl být používán pouze ve vhodných situacích, především pro tvorbu obecněji použitelných programových modulů ve vyšších programovacích jazycích. Čím blíže je programová vrstva vázána na uživatelskou problematiku, tím více podléhá změnám, proto by vrstva,se kterou uživatel pracuje,měla být napsána ve vyšším programovacím jazyku s maximálním množstvím volitelných parametrů. Při větším počtu aplikací je výhodné vytvořit systém pro generování uživatelských programových modulů. Optimalizace programů v JSA je velice náročná a vyplatí se pouze v případě,

že modul bude aplikován alespoň v několika desítkách případů. Do odladěných programů je vhodné promítat pouze nejnutnější změny, vše ostatní by mělo být ponecháno na další etapy ASŘ, v opačném případě se údržba programů stane brzdou dalšího rozvoje. Životnost uživatelského programového vybavení nebývá vyšší než 5 let, funkční úpravy bývají nezbytně častěji než jednou za 2 roky. Je-li doba tvorby uživatelského software srovnatelná s těmito časy, hrozí nebezpečí, že bude tato činnost ztrátová, zejména nejde-li o realizaci typového projektu.

7. Literatura

/1/ Duda,J.,Kmínek,M.,Michal,J.:Řízení cukrovarnické odparky pomocí mikropočítače,Průmysl potravin,1986,č.1,s.26-29

/2/ Duda,J.,Šrámková,I.,Král,K.:Využití mikropočítače VT-20 v odbytových činnostech,příspěvek na semináři experimentální sekce VT20A pořádaného sdružením uživatelů JSEP a SMEP pro Zpč.kraj,hotel"Doly" Řehenice,1985

/3/ Starý,J.:Mikropočítač a jeho programování,SNTL Praha,1984