

Základní koncepce operačních systémů pracujících v reálném čase (RTOS) (1)

Úvod

Oblast použití operačních systémů pro řízení v reálném čase RTOS (Real Time Operating System) neustále narůstá. Kromě tradiční oblasti řízení technologických procesů, kde se jedná často o kritické úlohy reálného času např. antiblokovací brzdový systém ABS u vozidel, a informačních systémů pro práci v reálném čase např. rezervace místenek v dopravě, kde se jedná o nekritický systém reálného času. Rychle narůstá použití RTOS zejména v oblasti vestavěných systémů (Embedded Systems) [3], [6].

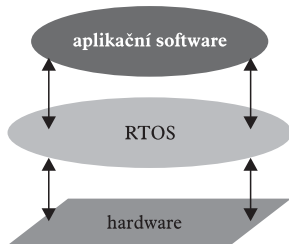
Aplikační oblasti použití vestavěných řídicích systémů s mikropočítači se neustále rozšiřuje a v současné době je lze nalézt ve všech oblastech lidské činnosti. Mezi hlavní oblasti jejich nasazení lze právě zařadit řízení v reálném čase. Tyto vestavěné systémy nás obklopují v každodenním životě od automobilů, domácích spotřebičů a mobilních telefonů v osobním životě, přes ordinace lékařů, průmyslovou automatizaci ve výrobních podnicích, až po kosmické lety. Většina uživatelů ani nevnímá vestavěné systémy jako počítače. Jsou totiž skryty uvnitř zařízení, které každodenně používají a které je obklopují a pomáhají jim v každodenním životě. Vestavěné systémy jsou charakterizovány omezenou velikostí paměti, omezeným výkonem, nestandardními interfacemi s okolním prostředím. Jejich spojení s okolním světem neodpovídá obvyklému spojení jež je používané u osobních počítačů jako je např. myš, klávesnice nebo grafický uživatelský interface. Vestavěné systémy jsou spojeny s okolním světem pomocí senzorů, akčních členů a speciálních komunikačních spojení [2].

Vestavěné systémy pracující v reálném čase v omezeném prostředí, omezení je dáno velikostí paměť a výkonem mikroprocesoru. Často je požadováno, aby vestavěné systémy poskytovaly své služby v přísně stanoveném čase vzhledem k uživateli a okolnímu světu. Je to právě paměť, rychlost a časová omezení, které určují použití RT operačního systému ve software vestavěného systému [4]. RT systémy a vestavěné systémy mají požadavky na reálný čas, zavádí striktní časové limity pro dodání výsledku zpracování. Operační systémy, které se používají u klasických a vestavěných systémů, se často liší pouze svým rozsahem nabízených služeb [6].

Služby jádra systému

Jak bude diskutováno dále, lze jádro systému považovat za část operačního systému, která poskytuje základní služby aplikacím běžícím na procesoru. Jádro RT operačního systému (RTOS) vytváří „abstraktní“ úroveň, která je skryta aplikačnímu software, jež zahrnuje např. detaily hardware procesoru (nebo sady procesorů), v době běhu aplikačních úloh. Takto vytváří interface mezi aplikačními úlohami a vlastním hardwarem počítače (obr. 1).

Jádro zahrnuje většinu nástrojů pro podporu reálného času, mezi které především patří rychlý multitasking, podpora přerušování, preemptivní a cyklické plánování úloh. Jádro je navrženo s minimální režii systému, která umožňuje rychlou a deterministickou odezvu na externí události. Výkonný mechanismus meziprocesové komunikace umožňuje nezá-

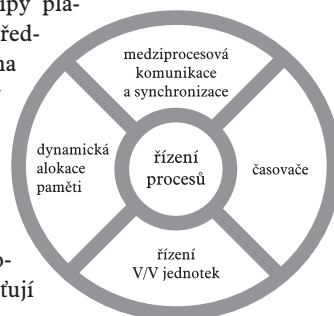


Obr.1 Jádro RTOS jako abstraktní úroveň mezi aplikačním softwarem a hardwarem vestavěného systému

vislým úlohám koordinaci svých aktivit v systémech reálného času. Jádro RTOS poskytuje pět hlavních skupin základních služeb pro aplikační software, jak je ukázáno na obr. 2. Mezi základní služby jádra patří služby pro řízení procesů, které jsou na obr. 2 znázorněny uprostřed.

Tato skupina služeb umožňuje vývojářům aplikačního software navrhout aplikační software ve tvaru několika samostatných oddělených částí, z nichž každá zajišťuje samostatnou část aplikačního software, která má svůj cíl a v řadě případů i vlastní požadavky na časový limit v reálném čase. Každá tato samostatná část softwaru se nazývá úloha, po spuštění proces nebo vlákno (task, process, thread) [6]. Služby z této skupiny umožňují spouštět procesy nebo vlákna a přiřadit jim příslušnou prioritu. Nejdůležitější službou RTOS z této skupiny je plánování procesů, která zajišťuje činnost vestavěného systému. Plánovač procesů řídí zpracování úloh aplikačního software, spouští je podle časového plánu a odpovídajícím způsobem. Principy plánování úloh byly uvedeny v předcházejícím článku. Tato skupina služeb zahrnuje také rychlý a výkonný přerušovací systém a rutiny pro obsluhu chyb.

Do druhé skupiny služeb jádra, podle obr. 2, patří meziprocesová komunikace a synchronizace [5]. Tyto služby zajišťují předávání zpráv mezi procesy, se zabezpečením jejich předání před poškozením nebo ztrátou informací. Rovněž umožňují vzájemnou koordinaci procesů tak, aby účelně spolupracovaly jeden s druhým. Bez použití těchto služeb RTOS, mohou procesy komunikovat s porušenými informacemi nebo v jiném případě vzájemně se rušit mezi sebou. I když mají často vestavěné systémy přísné časové požadavky, přesto většina jader RTOS poskytuje služby základních časovačů, jako je například možné zdržení (časový limit – deadline) nebo kontrola času prodlevy. Tyto služby jsou znázorněny na pravé straně obr. 2.



Obr.2 Základní služby jádra RT operačního systému

Jádro RTOS u některých systému poskytuje služby pro dynamickou alokací paměti, pro kritické úlohy je ovšem někdy vhodnější statické přidělení paměti. Tato skupina služeb umožňuje procesům „vypůjčit si“ blok (oblast) paměti RAM pro dočasné použití v aplikačním softwaru. Často jsou tyto bloky paměti sdíleny mezi procesy, a takto umožňují rychlou komunikaci s velkým množstvím dat mezi procesy. Některé velmi malá jádra RTOS, které jsou určeny pro malé rozsahy paměti, neposkytují služby dynamické alokace paměti.

Většina jader RTOS, ale ne všechny, poskytuje skupinu služeb pro řízení V/V jednotek. Jsou-li tyto služby k dispozici, umožňují pomocí ovladačů (driverů) standardizovaný přístup k mnoha hardwarovým jednotkám, které jsou typické pro vestavěné systémy.

Celá řada jader RTOS rovněž nabízí další volitelné komponenty operačního systému mezi které patří vyšší úroveň služeb jako je systém řízení souborů, komunikace v síti, řízení sítě, řízení databáze, grafický uživatelský interface, aj. Ačkoliv mnoho z těchto přídatných komponent je mnohem větší a mnohem komplexnější než-li je jádro RTOS, opírají se o přítomnost jádra RTOS a vy-

užívají s výhodou jeho základní služby. Každá z těchto volitelných přídatných komponent je zahrnuta ve vestavěném systému pouze tehdy, jestliže její služby jsou potřebné při implementaci vestavěné aplikace, v souladu s požadavkem minimální spotřeby paměti. Vysoká rozšiřitelnost podpůrných prostředků a přírůstkové sestavení a zavedení komponent umožňuje pružný vývoj širokého rozsahu aplikací.

V tomto článku se lze seznámit se základními službami jádra RTOS pro řízení procesů, meziprocesovou komunikaci a synchronizaci a dynamickou alokaci paměti.

Srovnání RTOS s obecným operačním systémem

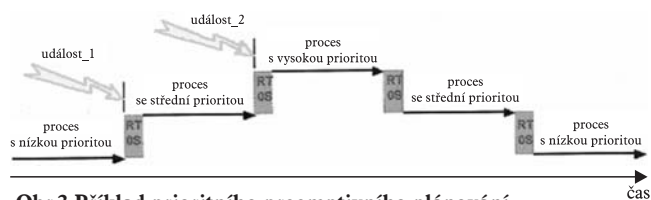
Mnoho NRT (non real-time) operačních systémů rovněž poskytuje obdobné služby jádra systému [7]. Základním rozdílem mezi obecným operačním systémem a RT operačním systémem je požadavek na „deterministické“ chování v čase u RT operačních systémů. Formálně „deterministické“ chování v čase znamená, že služby operačního systému spotřebují pouze známé a požadované množství času. Teoreticky lze čas těchto služeb vyjádřit pomocí matematických vztahů. Tyto vztahy musí být přísně algebraické a nesmí obsahovat žádné náhodné složky. Náhodné prvky v čase služeb mohou způsobit náhodná zpoždění v aplikačním software a mohou způsobit, že aplikace náhodně nestihne stanovené RT časové termíny – a tento scénář je jasně nepřijatelný pro RT vestavěné systémy.

Obecné NRT operační systémy jsou často zcela nedeterministické. Jejich služby mohou zavést náhodné zpoždění do aplikačního software a takto způsobit pomalou schopnost reagovat na podněty z aplikace a navíc v neočekávaných časových okamžicích. Jestliže je požadován po vývojáři NRTOS algebraický vztah popisující chování v čase některé ze služeb OS (například posílání zpráv mezi procesy), nikdy nedostanete požadovaný algebraický vztah. Kromě toho vývojář NRTOS (jako je Windows, Unix nebo Linux) se na vás podívá rozpačitým nebo nechápajícím pohledem. Deterministické chování v čase jednoduše není cílem těchto obecných operačních systémů. Na druhé straně operační systémy reálného času často jdou až za základní požadavek determinismu. Většina služeb jádra těchto operačních systémů nabízí konstantní časování, které je nezávislé na zavedení. Jinak řečeno, algebraický vztah je tak jednoduchý jako vztah: $T(\text{message_send}) = \text{konstanta}$, bez ohledu na délku zaslání zprávy nebo na jiných faktorech jako je počet procesů, fronta a zpráv, které řídí RTOS.

Plánování procesů

Většina RTOS provádí plánování procesů na základě prioritního preemptivního plánování. Každému procesu softwarové aplikace musí být přiřazena priorita, nejvyšší hodnota priority představuje požadavek na nejrychlejší odezvu. Rychlá odezva je přirozeně zajištěna právě preemptivním plánováním procesů. Preempece znamená, že plánovač může pozastavit kterýkoliv proces v kterémkoliv bodě zpracování, jestliže přijde požadavek od jiného procesu na své okamžité zpracování.

Základní pravidlo, které řídí prioritní preemptivní plánování, je v kterémkoliv okamžiku následující, „proces s nejvyšší prioritou, který je připraven na zpracování je procesem, jehož zpracování musí být zahájeno“. Jinými slovy, jestliže jsou dva procesy připravené na zpracování, jeden s nízkou prioritou a druhý s vysokou prioritou, pak plánovač naplánuje na zpracování proces s vysokou prioritou jako první. Proces s nižší prioritou se dostane ke zpracování až tehdy, když proces s vyšší prioritou ukončí svou aktuální činnost. Co se stane jestliže začne zpracování procesu s nižší prioritou a pak bude připraven na zpracování proces s vyšší prioritou? Toto může nastat např. při příchodu vnější události například sepnutí vypínače. Prioritní preemptivní plánovač bude postupovat následovně: Umožní procesu s nižší prioritou ukončit právě probíhající assemblerovskou instrukci. [Neumožní dokon-



Obr.3 Příklad prioritního preemptivního plánování znázorněný na časové ose

čení celého řádku v kódu vyššího jazyka; ale umožní dokončení zpracování do následujícího tiku hodin.] Zpracování procesu s nízkou prioritou pak bude okamžitě zastaveno a umožní zahájit zpracování procesu s vyšší prioritou. Po ukončení zpracování procesu s vyšší prioritou proces s nižší prioritou bude pokračovat ve zpracování. Toto je znázorněno na obr. 3, kde proces s vyšší prioritou je volán procesem se střední prioritou.

Zatímco proces se střední prioritou je zpracováván a událost aktivuje proces s vyšší prioritou je znázorněno na obr. 3 kde „spínač _2“ aktivuje proces s vyšší prioritou. V tomto případě běžící proces (se střední prioritou) měl by být přerušeno, aby proces s vyšší prioritou mohl být zpracován. Když proces s vyšší prioritou ukončí svoje zpracování, proces se střední prioritou pokračuje ve zpracování. Jakmile oba procesy se s vyšší a střední prioritou ukončí své zpracování, pak proces s nízkou prioritou pokračuje ve svém zpracování. Tato situace se nazývá „vnořená“ preempece.

Ve všech okamžicích kdy prioritní preemptivní plánovač je spuštěn na základě externí (např. sepnutí kontaktu) nebo vnitřní (např. příchod zprávy) události provede přepínání procesů, které musí proběhnout v následujících pěti krocích:

- Zjistí zda právě zpracováván (běžící) proces měl by pokračovat ve zpracování. V případě, že ne...
- Stanoví, který z procesů by měl přejít ze stavu připraven na zpracování.
- Uloží veškerá potřebná data o procesu, jehož zpracování bylo pozastaveno, ale bude pokračovat ve zpracování později.
- Nastaví potřebné data umožňující zahájení zpracování vybraného procesu.
- Zahájí zpracování vybraného procesu.

Těchto pět kroků se nazývá přepínání nebo přepnutí procesu.

Literatura

- [1] BARR, M. : Programming Embedded Systems in C and C++, O'REILLY 1999. ISBN 1-56592-354-5.
- [2] EDWARDS, L.: Embedded System Design on Shoestring: Achieving High Performance with a Limited Budget. LLH Technology Publishing 2001. ISBN 0-7506-7609-4
- [3] KOCIS, T., SROVNAL, V.: Operating Systems for Embedded Computers. In : Programmable Devices and Systems 2003-IFAC Workshop. Pergamon Press-Elsevier 2003, Great Britain, ISBN 0-08-044130-0, p. 359-364
- [4] QING, Li : Real-Time Concepts for Embedded Systems, CMP Books 2003, ISBN 1-57820-124-1
- [5] SRIDHAR, T.: Designing Embedded Communications Software. CMP Books 2003. ISBN 1-57820-125-X
- [6] SROVNAL, V.: Operační systémy pro řízení v reálném čase, VŠB Technická universita Ostrava 2003, ISBN 80-248-0503-0
- [7] TANENBAUM, A. S.: Modern Oprating Systems. Prentice Hall 1992

Pokračovanie v budúcom čísle.

**doc. RNDr. Jindřich Černožorský, CSc.
prof. Ing. Vilém Srovnal, CSc.**

**VŠB Technická univerzita Ostrava
Katedra měřicí a řídicí techniky, FEI**

35