

Přehled operačních systémů reálného času (1)

Vilém Srovnal

Úvod

Aplikace reálného času jsou zpravidla řešeny metodou souběžného zpracování jednotlivých úloh (multitasking). Jde například o aplikace zpracování dat v reálném čase, kdy je nutno zajistit kontinuální měření a zpracování dat, dodržet definovanou délku odezvy na měřené signály a současně zajistit komunikaci s nadřazenými systémy. Zde je paralelní zpracování jednotlivých úloh nezbytné. Při návrhu víceúlohové aplikace je nutno počítat s řadou problémů, které jednoúlohové programování vůbec nezná [1].

Systém pracující v reálném čase je takový systém, který vždy musí dát odpověď na vnější signál v konečné, předem stanovené lhůtě. To znamená, že jsou předem zajištěny maximální doby zpoždění reakce na dané podněty [1].

U takovýchto systémů bývá potřebná doba odezvy i řádově kratší než 10 μ s a tolerance v ukončení procesu ve stanovené lhůtě je v těchto případech nepřijatelná. V některých systémech může způsobit překročení časového limitu odezvy stejnou chybu jako nedodání odezvy vůbec [2].

Většina těchto RT operačních systémů (RTOS) používá standardní synchronizační a komunikační nástroje, které jsou definované v normě POSIX. Norma POSIX je velmi důležitá pro moderní RTOS, proto budou zde uvedeny základní informace důležité pro reálný čas [6].

POSIX (Portable Operating System Interface) definuje rozhraní přenositelnosti mezi operačními systémy. Tento standard byl vyvinut mezinárodní organizací IEEE a skupinou Open Group. Je zde definováno standardní rozhraní a prostředí operačního systému včetně interpretu příkazů a běžných utilit pro podporu přenositelnosti aplikací na úrovni zdrojového kódu. V současnosti se jedná o šestou revizi POSIXu, která odpovídá standardu IEEE Std 1003.1-2001. Pro reálný čas jsou důležité dodatky IEEE Std 1003.1d-1999 (rozšíření pro reálný čas) a IEEE Std 1003.1j-2000 (zdokonalené rozšíření pro reálný čas). Tento standard zahrnuje podporu pro přenositelnost zdrojového kódu pro aplikace reálného času.

Možnosti standardu IEEE Std 1003.1-2001 pro reálný čas jsou následující: asynchronní, synchronní a prioritní vstupy a výstupy, synchronizace souborů, mapování souborů v paměti, sdílení objektů v paměti, zamykání procesů a oblastí paměti, ochrana paměti, semafore, časovače, rozšíření signálů pro reálný čas, posílání zpráv a plánování procesů (prioritní plánování).

Zdokonalené rozšíření pro reálný čas má následující možnosti: výběr hodin, CPU hodiny procesu, monotónní hodiny, časové limity, popis paměťových objektů, obsluha příčinných závislostí a ojedinelých procesů.

Jinou zajímavou volitelnou podporou pro vývoj vestavěných aplikací je podpora vláken. Toto rozšíření normy POSIX definuje možnost vícenásobného toku řízení uvnitř procesu. Tyto toky řízení jsou nazývány vlákna a sdílejí svůj adresový prostor a většinu prostředků a atributů definovaných v operačním systému pro použití vlastníkem procesu. Tato specifická funkční oblast za-

hrnuje následující podporu obsluhy vláken: vytvoření, řízení a ukončení vícenásobných toků řízení, které sdílí společný adresový prostor.

V základní skupině je zahrnuta: dědičnost priorit vláken, ochrana a plánování zpracování vláken. Vlákna v zdokonaleném rozšíření zahrnují: CPU hodiny vláken, obsluhu ojedinelých vláken, kruhové blokování a ohraničení.

Standard IEEE Std 1003.1-2001 také nabízí sadu možností pro trasování, které může být velmi užitečné ve fázi vývoje vestavěné aplikace v reálném čase. Standardně pro trasování je nabízen výběr ze sady typů událost, které se použijí pro aktivaci trasovacího toku podle vybraných trasovacích událostí, které se vyskytují v toku zpracovávaných instrukcí a pro záznam trasovaných událostí. Trasované události mohou být získány později z trasovacího toku, což umožňuje analýzu chování systému. Tato skupina zahrnuje následující možnosti trasování: trasování, trasování pomocí událostního filtru, dědičnost trasování a uložení trasování.

Norma IEEE Std 1003.1-2001 definuje několik rozšíření XSI, které mohou být zajímavé pro vestavěné aplikace reálného času. Jsou to následující možnosti definované v tomto standardu: synchronizace souborů, mapování souborů v paměti, ochrana paměti, sdílená synchronizace vláken procesu, atributy adresace zásobníku vláken a velikost adresového prostoru zásobníku vláken. Systém může podporovat také jednu nebo více z následujících možností XSI Option Groups: šifrování, zdokonalený reálný čas, vlákna pro reálný čas, zdokonalená vlákna pro reálný čas, toky XSI a dědičnost. Všechny volitelné skupiny pro reálný čas společně se skupinou pro trasování představují vyhledávaný prostředek pro vývoj vestavěných aplikací v reálném čase.

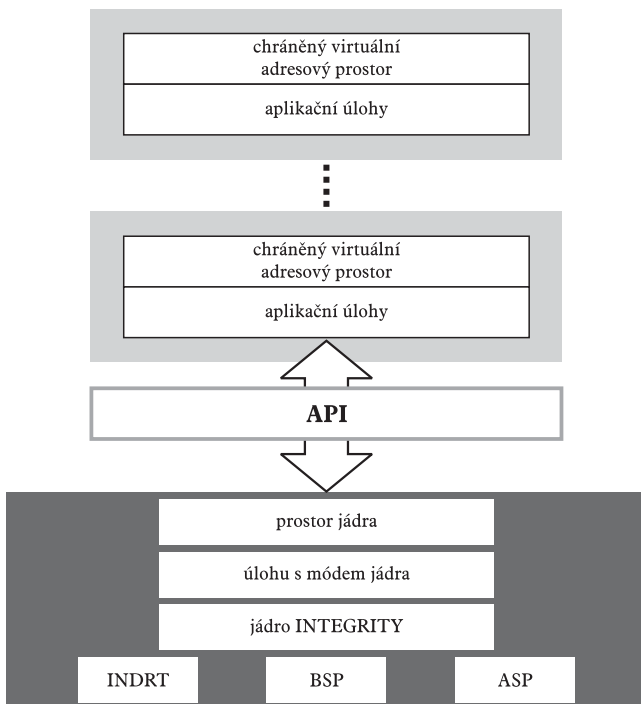
Přehled operačních systémů

Operační systémy pro řízení v reálném čase RTOS (Real Time Operating System) lze rozdělit do několika skupin podle aplikační oblasti, uplatnění ve vestavěných systémech a využití standardních operačních systémů osobních počítačů.

Jako **první skupinu** lze uvést RT operační systémy s obecnějším uplatněním. Zde především patří tradiční oblast řízení technologických procesů, kde se jedná často o kritické úlohy reálného času, ale také o informační systémy pracující v reálném čase, zde se jedná i o nekritické systémy reálného času. Mezi nejrozšířenější systémy této skupiny, zejména také i z historického počtu řídicích aplikací, patří např. OS9, QNX, a VxWorks a později RT Linux [6]. Většina těchto systémů má i verzi (embedded) pro použití ve vestavěných systémech [5]. Do této skupiny patří především následující RTOS:

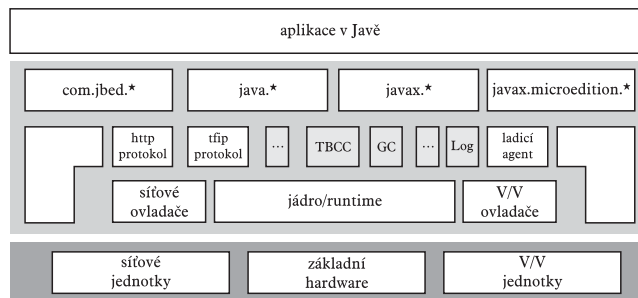
- **AMX RTOS** je kompaktní a modulární OS podle použité velikosti paměti, poprvé byl použit v roce 1980. Má většinu standardních funkcí RTOS s velice dobrými parametry rychlosti zpracování, odezvy na přerušení a času přepínání procesů u všech podporovaných procesorů (80386, 80486 a Pentium, 680x0 a rodinu 683xx, ColdFire, PowerPC, StrongARM a XScale, ARM7TDMI a MIPS32 procesory).

- **C EXECUTIVE a PSX.** C EXECUTIVE je víceúlohové RT jádro, které lze využít v paměti ROM pro vestavěné systémy a je použitelné pro 8-, 16- a 32-bitové CISC procesory, široký rozsah RISC procesorů a DSP. Umožňuje rychlé přepínání kontextu, má velice malou latenci času přerušeni a malou velikost jádra s podporou více než 20 procesorů. K dispozici je volitelný systém souborů kompatibilní s DOS, TCP/IP a SNMP. PSX RTOS poskytuje volitelnou podmnožinu systémových volání podle POSIX 1, tvoří však pouze desetinu velikosti Linuxu.
- **Eyrx** je RTOS navržený pro inženýrské systémy, které vyžadují spolehlivý a deterministický výkon v reálném čase s modulární, víceúrovňovou a velice výkonnou architekturou s volitelným intervalem tiku od 10 mikrosekund. Minimální velikost jádra je 25 KB, má neomezený počet procesů a prioritních úrovní (omezeno pouze systémovými prostředky), průměrný čas přepnutí procesu 0,7 μ s (P5 200 MHz), průměrný čas latence procesu 0,9 μ s (P5 200 MHz) a přesnost hodin 31,6 μ s/rok. Používá ochranu paměti.
- **INTEGRITY** je RT operační systém, který byl jako první certifikován IEEE a otevřenou skupinou POSIX. Je navržen pro 32-bitové a 64-bitové vestavěné procesory a zahrnuje nejnovější technologie RTOS: omezené jádro, rychlá a deterministická odezva v reálném čase, ochrana adresového prostoru, garantovaná dostupnost prostředků, dynamické zavádění úloh, provozní aktualizace a ladění, POSIX API a komplexní komunikační protokoly (obr. 1).



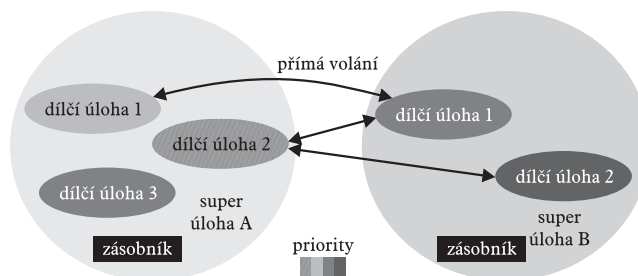
Obr.1 Blokové schéma RTOS INTEGRITY

- **JBed** má plně komponentně orientovanou architekturu. Jádro operačního systému se skládá z oddělených prvků – komponent, jádro není monolitické. To je důvod, proč neobsahuje žádná speciální volání jádra nebo virtuálního stroje. Operační systém se skládá ze souboru komponent, které provádějí potřebné služby. Cílem návrhu JBedu byla potřeba návrhu aplikací s malou velikostí, vysokou rychlostí a zároveň s dobrou bezpečností. Bezpečnost v komponentním softwarovém systému může být dosažena pouze použitím moderního programovacího jazyku, který je bezpečný. Takovým příkladem bezpečného programovacího jazyku je Java nebo Component Pascal. Součástí jádra JBedu je plánovač souběžného zpracování úloh, systém alokace a úklidu paměti a systém ošetření výjimek. Runtim systému JBed zahrnuje pouze uvedené jádro plus komponenty aplikace (obr. 2).



Obr.2 Struktura OS JBed

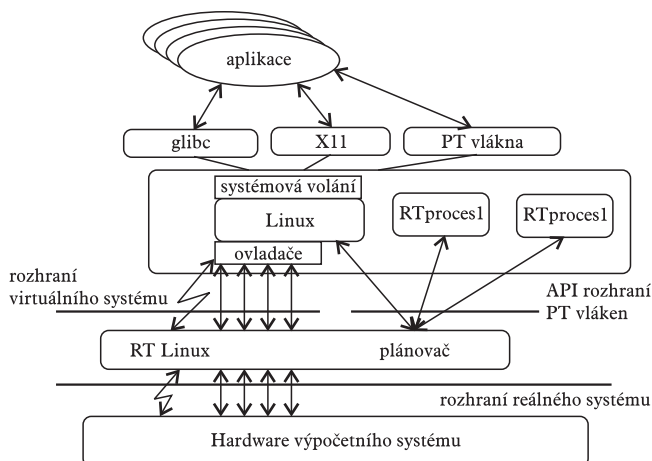
- **MQX RTOS** lze konfigurovat tak, aby zabíral méně než 6K byte paměti ROM, včetně jádra, přerušeni, semaforů, front a řízením paměti. MQX RTOS poskytuje i vysoký výkon pro aplikace RTOS s procesorem 200 MHz PowerPC MPC8248, s latencím času přerušeni 0,331 μ s a časem přepnutí kontextu 0,558 μ s podle benchmarku.
- **Microware OS-9** je systémově bezpečný, spolehlivý a odolný chybám. Je to RTOS s vysokou použitelností a spolehlivostí. Uživatelé mohou dynamicky přidávat a vyměňovat moduly během běhu systému. Architektura modelu procesu a vláken OS-9 umožňuje vynikající výběr pro použití v kritických aplikacích. Rozšiřitelné možnosti V/V jsou snadno proveditelné a unifikované se stylem V/V v Unixu, síťovým a grafickým řešením, řízením souborů MPEG, aj.
- **OSE RTOS** je RTOS s přímým zasíláním zpráv a lze jej použít pro aplikace s RISC, DSP, a CISC procesory. Plně preemptivní jádro s prioritním plánováním bylo optimalizováno tak, aby poskytovalo vysokou propustnost dat, je dosti kompaktní pro použití většiny vestavěných systémů. Architektura OSE je založena na zprávách s jednoduchým a výkonným systémem volání. Předávání zpráv v OSE je jednoduché a intuitivní a navíc slouží jako konceptní brána k víceúrovňovým a distribuovaným multiprocessorovým vestavěným systémům.
- **PDOS** je velice výkonný RT operační systém pro architekturu rodiny procesorů 68K (např. Motorola, FORCE, aj.). PDOS je spolehlivý, deterministický a osvědčený. Je vybaven robustním vývojovým prostředím a lze jej pružně rozšiřovat o další možnosti. Operační systém PDOS je implementován dvěma způsoby buď jako vlastní PDOS spustitelný na cílovém systému nebo jako PDOS v EPROM (VMEPROM). Vlastní PDOS je použitelný na široké škále systémů s VME sběrnici.
- **Portos** je novou technologií RTOS. Nejdůležitější vlastnosti jsou snadné programování, vysoký výkon, malá paměť, rozšiřitelnost na OS Windows a Linux, rozšiřitelnost o programovací jazyky: Java, Python a Matlab. Nová koncepce jsou super úlohy, prioritní funkce, prioritní objekty, automatické předávání zpráv mezi úlohami (obr. 3) a nepoužívají se poštovní schránky.
- **QNX** je navržen pro aplikace reálného času. Zajišťuje víceúlohové, víceuživatelské a síťové zpracování s prioritně řízeným preemptivním plánováním a rychlým přepínáním kontextu a všemi nezbytnými prostředky pro systémy reálného času. QNX patří mezi pružné operační systémy s dodržením standardu POSIX a aplikačního rozhraní API. Vysoká modularita OS umožňuje vývojovým pracovníkům jednoduše modifikovat



Obr.3 Volání v super úlohách v RTOS Portos

operační systém podle požadavků příslušné aplikace. Od minimální konfigurace jádra s několika moduly u vestavěných řídicích systémů až po plnou konfiguraci síťového systému pro desítky uživatelů. QNX umožňuje modifikovat systém tak, aby využíval jen ty prostředky, které jsou nezbytně potřeba. QNX dosahuje dobré spolehlivosti, modularity a jednoduchosti využitím dvou základních technik: architektura mikrojádra a meziprocesová komunikace založená na zprávách. Mikrojádro zahrnuje pouze malou množinu služeb jádra, mezi které patří: služby vláken, zasílání zpráv, mutexy, podmíněné proměnné, semaforey, signály a plánovač. Jádro může být dynamicky rozšiřováno připojením procesů poskytujících služby, jako je systém souborů, práce se sítí, fronty zpráv a ovladače zařízení. Každý proces běží ve svém vlastním adresovém prostoru v chráněné paměti, což zvyšuje spolehlivost systému. QNX používá prioritní preemptivní plánovací algoritmus se třemi metodami plánování.

- **RTLinux** je malý a rychlý operační systém, který je v souladu s normou POSIX 1003.13, což je standard pro minimální operační systém reálného času. Architektura RTLinuxu obsahuje navíc vrstvu vytvářející prostředí virtuálního hardwaru mezi standardním jádrem Linuxu a skutečným hardwarem počítače (obr. 4). RTLinux lze považovat za úplný operační systém s předvídatelnou činností v reálném čase, bez rozhraní pro standardní Linux bez reálného času. Vlákna RTLinuxu jsou zpracovávána přímo plánovacím algoritmem s pevnou prioritou. Jádro a všechny procesy standardního Linuxu jsou řízeny plánovačem RTLinuxu jako úlohy v pozadí. RTLinux vytváří úplný obecný operační systém, který běží nad malým předem definovaným jádrem RTOS.



Obr.4 Architektura RT Linuxu

- **SMX** je modulární RTOS. SMX má mnoho funkcí je vysoce výkonný a má preemptivní multitaskingové jádro. Je účinný a s velmi krátkou latencí přerušení. Podporuje procesory ARM, ColdFire, PowerPC, SH3/4, a x86. Přenositelnou verzi je možné přenést na libovolnou 32-bitovou CPU.
- **VxWorks** je operační systém pro řízení v reálném čase, který se vyznačuje RT mikro-jádrem wind, progresivní síťovou podporou, výkonným systémem souborů, řízením V/V jednotek, podporou C++ a runtimu. Základní funkce jádra mohou být doplněny připojením dalších komponent. Mikro-jádro zahrnuje většinu nástrojů pro podporu reálného času, mezi které patří především rychlý multitasking, podpora přerušení, preemptivní a cyklické plánování úloh. Mikro-jádro je navrženo s minimální režii systému, což umožňuje rychlou a deterministickou odezvu na externí události. Patří k nejrozšířenějším operačním systémům reálného času zejména v oblasti průmyslových aplikací vestavěných systémů. Je příznivě hodnocen pro svou výkonnost, flexibilitu, kompatibilitu a rozšiřitelnost. Poskytuje velmi výhodné runtime prostředí pro vývoj vestavěných apli-

káci. Je bezpečný i při použití v kritických aplikačních úlohách, například od antiblokovacího brzdového systému ABS u vozidel až po aplikace v meziplanetárním výzkumu. Systém je kompatibilní s řadou průmyslových standardů a lze jej provozovat na většině nepoužívanějších základních jednotkách – CPU. Vývojové prostředí Tornado II (obr.8) umožňuje snadný návrh vestavěných systémů a zahrnuje komplexní sadu křížových vývojových nástrojů a další vybavení včetně komunikačních nástrojů pro spojení vývojového hostitelského systému s cílovým systémem pro aplikace.

Literatura

- [1] ČERNOHORSKÝ, J., SROVNAL, V.: Systémy reálného času, AT&P journal, Issue 6-8, Volume 12, June-August 2005, ISSN 1335-2237, p. 20 – 21, p. 93, p. 57 – 59
- [2] ČERNOHORSKÝ, J., SROVNAL, V.: Základní koncepce operačních systémů pracujících v reálném čase (RTOS), AT&P journal, Issue 9-, Volume 12, September 2005, ISSN 1335-2237
- [3] BARR, M.: Programming Embedded Systems in C and C++, O'REILLY 1999. ISBN 1-56592-354-5.
- [4] EDWARDS, L.: Embedded System Design on Shoestring: Achieving High Performance with a Limited Budget. LLH Technology Publishing 2001. ISBN 0-7506-7609-4
- [5] 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
- [6] SROVNAL, V.: Operační systémy pro řízení v reálném čase, VŠB Technická universita Ostrava 2003, ISBN 80-248-0503-0

Pokračovanie v budúcom čísle.

prof. Ing. Vilém Srovnal, CSc.

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

26