

Operačné systémy reálneho času, prehľad možností

Viszus Eugen · Informačné technológie

16.06.2014



Príspevok je zameraný na prehľad možností operačných systémov reálneho času, ktoré sú v súčasnosti dostupné na trhu. V príspevku sú opísané riešenia určené pre operačný systém Windows, operačný systém Linux a tiež vložené operačné systémy (Embedded OS). Pri tradičných operačných systémoch Windows a Linux sa z pravidla jedná o tzv. rozšírenia reálneho času, ktoré upravujú vlastnosti už existujúceho systému. Dôležitým faktorom pri operačných systémoch reálneho času je okrem výkonu aj samotná cena riešenia. V tomto príspevku predstavíme voľne dostupné produkty ako aj komerčné riešenia.

Úvod

Operačný systém je softvér, ktorý spravuje hardvérové prostriedky počítača a vytvára rozhranie medzi užívateľom a hardvérom. Operačný systém vykonáva základné operácie ako sú správa pamäti, pridelenie času procesora jednotlivým systémovým požiadavkám podľa určitých priorít alebo správa vstupných a výstupných zariadení. Základným prvkom každého riadiaceho systému je riadiaci počítač. Každý riadiaci počítač obsahuje nejaký druh operačného systému. Pri takomto systéme je dôležité, aby na riadiacom počítači nedochádzalo k zbytočným oneskoreniam, teda aby riadiaci algoritmus zbehol bez prerušení v čo najkratšom možnom čase a aby bol výsledok s presnou periódou odovzdaný ďalej bez oneskorení. Za týmto účelom sa v riadiacich počítačoch používajú operačné systémy reálneho času. Existujú dve základné kategórie takýchto operačných systémov. Sú to:

- Soft real-time OS
- Hard real-time OS

Soft real-time operačný systém môže byť použitý v aplikáciách, kde je prijateľné určité oneskorenie. Toto oneskorenie spôsobí iba zníženie kvality riadenia a nemá kritický dopad na samotný proces. Hard real-time operačný systém sa používa pri aplikáciách, kde aj minimálne oneskorenie nad povolenú hranicu môže spôsobiť zlyhanie celého procesu. Hard real-time garantuje určitú maximálnu dobu odozvy systému [1]. Operačný systém reálneho času môžeme získať úpravou jadra už existujúceho systému. Takýmto spôsobom väčšinou získame systém na univerzálne použitie. Pri systémoch rodiny Windows sa zväčša jedná o úpravu jadra na báze komerčného softvéru. Pri systémoch Linux ide naopak zväčša o voľne dostupné riešenie. Pre špecifické systémy používané napríklad v leteckom priemysle alebo vo vesmírnom

programe sú využívané operačné systémy vytvorené na mieru, tzv. vložené operačné systémy. Sú to napríklad VxWorks alebo Windows CE.

RT OS na báze Windows

Operačné systémy z rodiny Windows sú už dlhé roky jednými z najpoužívanejších operačných systémov na svetovom trhu. V roku 1985 bol vydaný operačný systém Windows 1.0. Nejednalo sa o plnohodnotný systém ale išlo o grafickú nadstavbu systému MS DOS. Bol to jednoduchý operačný systém, ktorý nedokázal spracovávať viac úloh súčasne, teda pracoval iba s jedným vláknom. V roku 1998 boli zdrojové kódy kompletne prepísané a vznikol systém s názvom Windows NT. Jednou z najväčších výhod bola omnoho vyššia stabilita systému oproti jeho predchodcom. Práve stabilita operačného systému je jedným z hlavných predpokladov pre riadiaci počítač používaný v aplikáciách reálneho času. Na základoch Windows NT fungujú aj dnešné systémy Windows. Na začiatku treba povedať, že systém Windows nie je sám o sebe operačný systém reálneho času. Obsahuje však niekoľko funkcií, ktoré sú vhodné pre aplikácie v reálnom čase. Sú to nasledovné:

- práca s viacerými vláknami súčasne (multithreading)
- úrovne priority pre reálny čas
- systémové časovače a hodiny s vysokým rozlíšením

Windows NT podporuje celkovo 32 úrovni priority (0 je najnižšia, 31 najvyššia) rozdelených do 6 tried. Každý proces patrí do jednej triedy priorít. Rozdeľujeme ich ako idle, below normal, normal, above normal, high a real-time. Každá trieda priorít obsahuje nasledujúce úrovne priority pre vlákna daného procesu:

- thread_priority_idle
- thread_priority_lowest
- thread_priority_below_normal
- thread_priority_normal
- thread_priority_above_normal
- thread_priority_highest
- thread_priority_time_critical

Predvolená prioritná trieda pri vzniku procesu je normal. Jeho vlákna dostanú pri vzniku priority thread_priority_normal. Úrovne priority tried normal a high sú periodicky prepočítavané systémom. Po vyčerpaní času procesora, ktorý bol vláknom pridelený sa prioritna vlákna zníži. Ak je vlákno blokováné na vstupno-výstupnom zariadení a jeho pridelený čas procesora nebol vyčerpaný, prioritna vlákna sa zvýši. Úroveň priority vlákna sa môže od základnej hodnoty líšiť maximálne o dva stupne vyššie alebo nižšie. Keďže Windows NT používa preemptívny plánovač na základe priorít, procesy s prioritou real-time sú nadradené všetkým ostatným [2].

Windows NT však obsahuje aj funkcie, ktoré sú proti filozofii systémov reálneho času. Jednou z nich je obsluha prerušení. Problémom je, že úroveň priority prerušenia je vždy vyššia ako prioritna užívateľského vlákna, vrátane vlákien zaradených do triedy real-time. Pri výskyte prerušenia sa vykoná obslužná procedúra priradená k prerušeniu (Interrupt Service Routine), ktorá sa spracuje rýchlejšie ako ľubovoľné

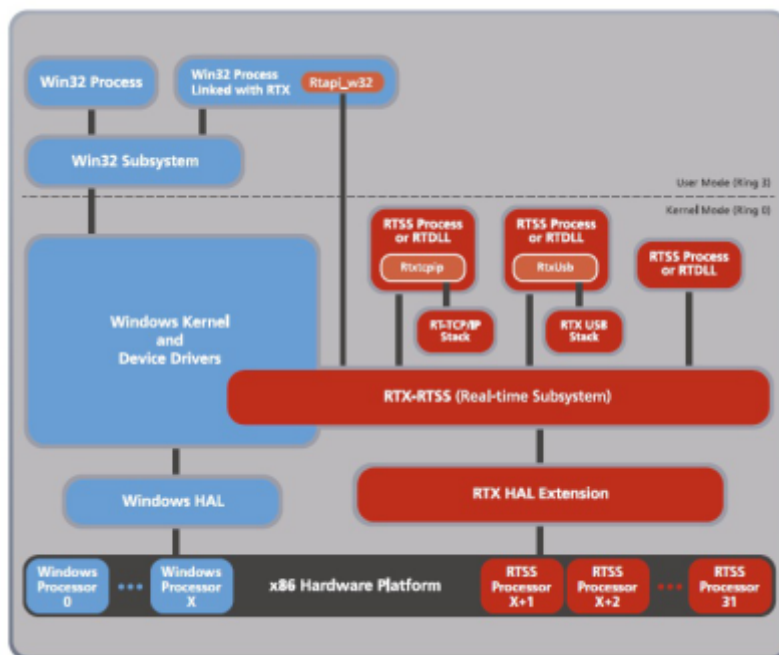
vlákno patriace do triedy real-time. Systém Windows CE (embedded) túto vlastnosť neobsahuje.

Ďalším problémom je protokol pre podporu zdieľania systémových prostriedkov. Windows NT neobsahujú žiadnu podporu pre zdieľanie kritických systémových zdrojov pre úlohy reálneho času. Vďaka tomu sa môže vyskytnúť problém nazývaný „priority inversion“, ktorý môže spôsobiť oneskorenie vlákna alebo dokonca zlyhanie celého systému. Tieto problémy môžu byť vyriešené upravením, prípadne nahradením jadra systému softvérom tretích strán. Pri systémoch Windows sa jedná o komerčné riešenia.

RTX RTOS rozšírenie pre Windows

RTX od výrobcu IntervalZero je rozšírenie reálneho času pre operačný systém Windows. Nejedná sa o samostatný operačný systém. Použitím tohto rozšírenia rozšírime Windows na systém, ktorý spĺňa požiadavky pre hard real-time aplikácie. Výhodou je, že užívateľ stále pracuje v známom grafickom rozhraní. RTX funguje na princípe rozšírenia jadra systému o oddelení subsystém, ktorý plánuje a riadi všetky procesy reálneho času. RTX podporuje symetrický multiprocessing. Pri viac procesorovom systéme môže plánovať procesy a vlákna pre ktorýkoľvek procesor. Pri prevádzke RTX na viac procesorovom systéme treba určiť, koľko procesorov bude využívať systém Windows a koľko sa vyhradí pre subsystém reálneho času. RTX podporuje systémy s maximálne 32 procesormi. Prostredie RTX môže byť nakonfigurované tak, že pri zlyhaní systému Windows prevezme kontrolu a bezpečne ukončí všetky procesy reálneho času.

RTX taktiež obsahuje špeciálne ovládače pre veľké množstvo bežne dostupných sieťových kariet od rôznych výrobcov ako napríklad Intel alebo Realtek. Vďaka tomu môže užívateľ využiť svoju sieťovú kartu na komunikáciu prostredníctvom niektorých priemyselných komunikačných sietí. Pri programovaní aplikácii reálneho času sa využíva štandardné Win32 API, teda rovnaký kód môže bežať ako Windows proces alebo ako proces reálneho času. Ako programovacie jazyky sa používajú C a C++. Výhodou je taktiež možnosť využiť štandardné vývojové prostredia ako napr. Visual Studio. Na komunikáciu medzi procesmi reálneho času a Windows procesmi sa využívajú klasické prvky medzi procesovej komunikácie IPC, teda zdieľaná pamäť, semaforey a mutexové zámky [3].



Obr. 1: Architektúra RTX

Zhrnutie:

- jednoduchší vývoj softvéru, použitie štandardných programovacích jazykov vývojových prostredí, strmá krivka učenia,
- zníženie nákladov jednoduchším vývojom a použitím bežnej platformy x86, lacnejší hardware,
- zvýšenie výkonu systému priradením viacerých jadier/procesorov subsystému reálneho času,
- využitie bežných sieťových rozhraní pre komunikáciu v reálnom čase.

RT OS na báze Linux

Operačný systém Linux sa postupne vyvinul z tzv. jadra(Kernel), ktoré vytvoril študent univerzity v Helsinkách Linus Torvalds. Prvá verzia tohto systému s číslom 0.01 bola vydaná v septembri 1991. Linux Kernel je voľne šíriteľný operačný systém pod licenciou GPL (General Public Licence), ktorá zaručuje slobodu použitia, ale zároveň chráni Linux pred zneužitím. Jeho zdrojové kódy sú verejne prístupné a voľne modifikovateľné. V priebehu niekoľkých rokov vznikla okolo Linuxu obrovská komunita programátorov, ktorí sa podieľajú na vývoji nových jadier. Vďaka tomu sú všetky objavené chyby rýchlo opravené a opravy sú hneď dostupné na internete. Pôvodne bol Linux určený len pre architektúry i386. Postupom času sa však Linux rozšíril aj na iné architektúry ako ARM alebo PowerPC. Linux sa rozšíril z bežných PC a serverov aj do mobilných telefónov (Android), do špecializovaných zariadení ako sú napríklad sieťové smerovače alebo do bežných domácich spotrebičov ako sú smart televízory alebo chladničky.

Pri množstve vývojárov s rôznymi predstavami o dokonalom operačnom systéme začali postupom času vznikať rôzne verzie OS založených na jadre Linux. Tieto rôzne verzie sa nazývajú distribúcie. Najznámejšími sú Ubuntu, SuSE alebo Fedora, ktoré obsahujú aj grafické užívateľské rozhranie a teda sú užívateľsky veľmi prívetivé. Existujú však aj extrémnejšie distribúcie v ktorých sa pracuje prostredníctvom príkazového riadku ako

napríklad Slackware alebo Arch Linux. Vo všeobecnosti, či už s grafickým rozhraním alebo bez, má Linux v porovnaní s inými operačnými systémami veľmi nízke systémové nároky [4]. Z hľadiska real-time aplikácií spĺňa súčasný Linux požiadavky soft real-time operačného systému. Medzi vlastnosti vhodné pre aplikácie reálneho času patria:

- deterministický plánovač,
- podpora preemptivity,
- časovač s vysokým rozlíšením,
- PI(Priority Inheritance) mutexové zámky, ktorá zabraňuje zablokovaniu systému s dôvodu problému „priority inversion“,
- obsluha prerušenia vláknami.

Aby sme mohli používať Linux ako hard real-time operačný systém, musíme použiť jedno z dostupných rozšírení jadra operačného systému. S pravidla sa jedná o open-source projekty, ktoré sú voľne dostupné na internete. Väčšinou sa takéto rozšírenie aplikuje na zdrojový kód čistého jadra záplatou, ktorá pozmení v zdrojovom kóde jadra niekoľko stoviek až tisícok riadkov. Následne sa pozmenené jadro nainštaluje. Spomenieme tri najpoužívanejšie rozšírenia reálneho času pre systém Linux a to Real-Time Application Interface (RTAI), PREEMPT RT Patch a Xenomai.

Real-Time Application Interface

Real-Time Application Interface(RTAI) predstavuje nadstavbu jadra Linux, ktorá do bežného operačného systému vloží prostriedky potrebné pre aplikácie reálneho času. Vývoj RTAI začal v roku 1998 na polytechnickej univerzite v Miláne. Jeho inštalácia prebieha formou aplikácie záplaty. Hlavnou zmenou oproti bežnému jadru je zoskupenie všetkých časovo kritických smerníkov interných dát jadra, ktoré sú dôležité pre aplikácie reálneho času do jednej štruktúry, aby ich mohol RTAI dynamicky nahradiť v prípade vzniku požiadavky na odozvu v reálnom čase. Vo všeobecnosti ide o funkcie a dátové štruktúry potrebné na manipuláciu s hardvérovými prerušeniami, softvérovými obslužnými rutinami prerušenia, systémovými volaniami a časovačmi. Smerníky na tieto objekty sa môžu meniť dynamicky.

Keďže jadro Linux je modulárne a umožňuje vyberanie a vkladanie modulov jadra za behu systému, RTAI túto vlastnosť využíva. Po zavedení špecifických modulov do jadra preberá obsluhu všetkých potrebných funkcií RTAI (dynamickou zmenou smerníkov), avšak softvérovou emuláciou zabezpečuje nerušený chod zvyšku Linuxu vrátane používateľských aplikácií. Výsledkom je, že RTAI na najnižšej úrovni odchyťava všetky prerušenia a zároveň riadi činnosť všetkých používateľských procesov. V prípade, že nastane prerušenie, RTAI ho okamžite prevezme a rozhodne čo vykoná. Ak je zadaná obslužná rutina pre toto prerušenie, je v čo najkratšom čase vyvolaná, inak je prerušenie uvedené do stavu čakania. V prípade, že sa vykonajú všetky časovo kritické úlohy, môže dostať Linux povolenie na vykonanie všetkých prerušení, ktoré sú v stave čakania [1] [5].

Ako základ pri inštalácii je možné použiť akúkoľvek distribúciu Linux OS avšak odporúčajú sa konzervatívne distribúcie ako napríklad openSuSE. Všetky RTAI patche sú určené pre špecifické čisté(vanilla) Linux jadro. Inštalácia na jadro danej distribúcie je síce možná ale napriek tomu sa neodporúča.

PREEMPT RT Patch

Hlavnou filozofiou a princípom fungovania PREEMPT RT záplaty je minimalizácia kódu jadra, ktorý nespĺňa preemptívne požiadavky tak, že takýto kód je nahradený preemptívnym kódom s podobnou funkcionalitou. Taktiež sa kladie dôraz na minimalizáciu množstva kódu, ktorý musí byť pozmenený. Výhodou je, že na rozdiel od iných rozšírení reálneho času vývoj PREEMPT RT záplaty je veľmi flexibilný a drží krok s vývojom nových verzii jadra. Hlavné zmeny v jadre [6]:

- tvorba jednoduchých zámkov v jadre (pomocou spinlocks) bola nahradená rtmutex zámkami,
- kritické sekcie chránené pomocou spinlock_t a rwlock_t sú teraz preemptívne, vytváranie nepreemptívnych sekcii je stále možné pomocou raw_spinlock_t,
- implementácia dedenia priorít pre spinlocks a semaforey v jadre,
- konverzia obsluhy prerušenia na preemptívne vlákna jadra,
- konverzia starých Linux Timer API na samostatné štruktúry využívajúce High Resolution Kernel časovače a jeden samostatný časovač pre timeouty. Táto úprava vedie k možnosti použiť userspace POSIX časovače s vysokým rozlíšením.

Vďaka neustálemu vývoju a podpore najnovších jadier Linux je toto riešenie vhodné aj pre mini počítače založené na architektúre ARM. Tieto platformy typu SoC (System on a Chip) využívajú najnovšie technológie a preto je vhodné použiť čo najnovšiu verziu jadra, ktorá obsahuje potrebné ovládače. Vývoj PREEMPT RT záplaty pravdepodobne zanedlho skončí, keďže všetky jeho funkcionality už budú implementované v hlavnej vývojovej vetve jadra Linux.

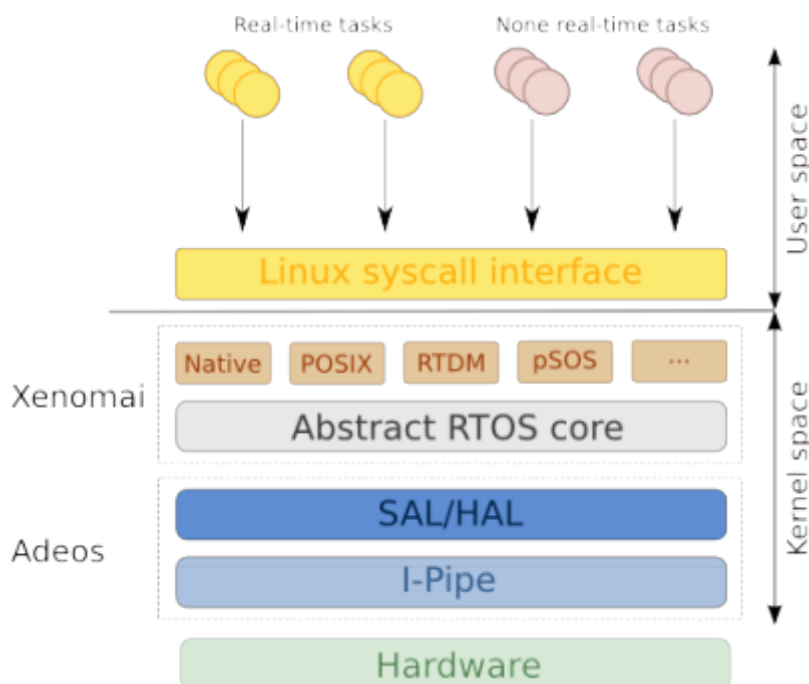
Xenomai

Xenomai je framework pre vývoj aplikácií reálneho času, ktorý spolupracuje s jadrom Linuxu. V systéme implementuje plánovač reálneho času a poskytuje podporu pre hard real-time aplikácie bežiacie v užívateľskom priestore (userspace). Tento framework je plne integrovaný do prostredia Linux tak ako predchádzajúce riešenia. Xenomai projekt bol spustený v auguste 2001. V roku 2003 sa spojil s projektom RTAI, ale vďaka rozdielnym pohľadom na ďalšie smerovanie sa v roku 2005 opäť rozdelili a vznikol samostatný Xenomai.

Na to aby mohli byť do Linuxu vložené vlastnosti reálneho času, musí byť medzi hardwarom a jadrom Linuxu implementované mikro jadro. Toto mikro jadro je zodpovedné za vykonávanie úloh v reálnom čase a odchyťvanie prerušení, ktoré blokuje aby sa nedostali k jadru Linux. Týmto bráni jadru Linuxu aby sa pri vykonávaní prerušenia nepresadilo pred mikro jadrom. Úlohy Linux jadra sú vykonávané na pozadí s nízkou prioritou. Zaujímavosťou frameworku Xenomai je, že poskytuje rôzne rozhrania API pre tvorbu RT úloh, časovačov, IPC a synchronizačných nástrojov. Tieto rôzne API sa nazývajú Xenomai skiny (skins) a sú dostupné pre POSIX rozhranie, RTAI, VxWorks a iné. Pri prenose aplikácie reálneho času z inej platformy na platformu s nainštalovaným Xenomai nie je nutné prepisovať zdrojový kód podľa nového API rozhrania.

Na spracovanie a správu prerušení v reálnom čase používa Xenomai technológiu

ADEOS(Adaptive Domain Environment for Operating System), ktorá používa nano jadro fungujúce v reálnom čase(ešte nižšia úroveň ako mikro jadro). ADEOS slúži na zdieľanie hardwarových zdrojov medzi viacerými operačnými systémami alebo medzi viacerými časťami jedného operačného systému [7]. Architektúra systému s nainštalovaným Xenomai je na obrázku 2.



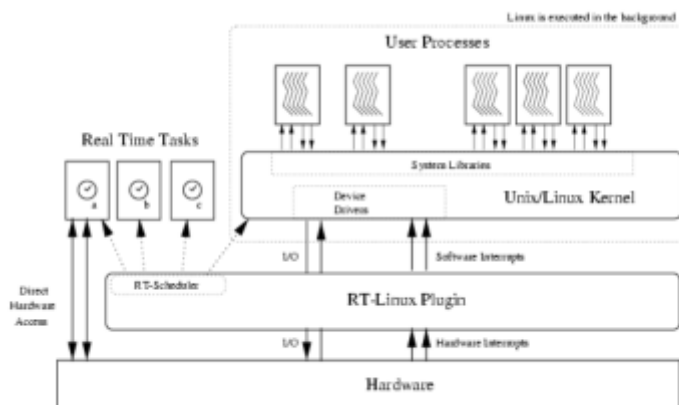
Obr. 2: Architektúra Xenomai

RTLinux (Wind River Linux)

RTLinux je hard real-time operačný systém založený na platforme Linux. Jeho vývoj začal na technologickom inštitúte v Novom Mexiku. Autormi tohto operačného systému sú Victor Yodaiken, Michael Barabanov a Cort Dougan. Následne sa z tohto projektu stal komerčný produkt, ktorý bol ďalej vyvíjaný pod záštitou spoločnosti FSMLabs. V roku 2007 však divíziu pre embedded technológie získala spoločnosť Wind River. V súčasnosti RTLinux vystupuje pod názvom Wind River Linux. Aktuálna verzia je Wind River Linux 6.

Bežný plánovač používaný v systémoch Linux nepodporuje preemptívne správanie, ale je zameraný hlavne na viditeľný výkon, teda pružnosť systému. RTLinux zavádza medzi hardvér počítača a štandardné jadro systému akúsi abstraktnú vrstvu nazývanú „virtual machine“. Pre jadro systému sa táto vrstva tvári ako hardvér. V skutočnosti je to však hard real-time mikro jadro, ktoré používa vlastný preemptívny plánovač s fixnou prioritou. Štandardné jadro teraz beží ako nezávislá úloha nad RT mikro jadrom, ktorej plánovač priradil najnižšiu prioritu. Štandardné jadro teda ostane nedotknuté.

Hardvérové prerušenia sú odchyťované mikro jadrom. Ak sa prerušenie netýka procesu reálneho času, je odložené dovtedy, kým RT mikro jadro nebude nečinné. Následne sa toto prerušenie pošle štandardnému Linux jadru ako softvérové prerušenie. Ak sa prerušenie týka procesu reálneho času, zavolá sa prislúchajúca real-time obslužná funkcia [8]. Architektúra systému RTLinux je zobrazená na obrázku 3.



Obr. 3: Architektúra RTLinux systému.

RTLinux je modúlárne orientovaný operačný systém. Do systému je možné doplniť vlastné funkcionality vložení nových modulov jadra. Základom RT jadra sú dva moduly. Prvým je plánovač a druhým modul, ktorý implementuje RT FIFO zásobník pre úlohy. Vložení nového modulu môže užívateľ nahradiť existujúci plánovač alternatívnym riešením. Je to napríklad „Early Deadline First“ plánovač, ktorý pre RTLinux implementoval Ismael Rippol alebo „Rate Monotonic“ plánovač, ktorý implementoval Oleg Subbotin. Real-time užívateľské aplikácie sú taktiež programované ako moduly jadra, takže môžu byť dynamicky vkladané do pamäti [9].

V súčasnosti RTLinux podporuje štyri procesorové architektúry a to ARM, x86, MIPS a PowerPC. RTLinux je dodávaný aj s vývojárskym prostredím Wind River Workbench, ktoré užívateľovi uľahčuje programovanie aplikácii reálneho času. Natívnym programovacím jazykom je kombinácia C/C++. Keďže sa v súčasnosti jedná o plne komerčný projekt, je možné k systému zakúpiť aj technickú podporu.

Vložené operačné systémy

Vložené(embedded) operačné systémy sú operačné systémy určené pre embedded systémy. Embedded systém je počítačový systém vytvorený pre vykonávanie špecifickej funkcie. Často takýto systém musí spĺňať požiadavky pre aplikácie reálneho času. Príkladom môže byť televízny Set-Top-Box prijímač, ktorý pri spracovaní obrazu a zvuku musí dodržiavať určité požiadavky na reálny čas, aby nedochádzalo k strate kvality výstupného signálu. Takýto prijímač obsahuje vložený operačný systém, ktorý zabezpečuje jeho správnu funkcionality a taktiež interakciu s užívateľom. Tento operačný systém funguje spoľahlivo pri takomto type zariadení, ale pre iné zariadenia by bol absolútne nevhodný. V tomto príspevku predstavíme najznámejší vložený operačný systém a to VxWorks.

VxWorks

VxWorks je operačný systém reálneho času určený pre embedded zariadenia. Bol vytvorený spoločnosťou Wind River Systems(spoločnosť v roku 2009 odkúpil Intel). Prvá verzia VxWorks systému vznikla v roku 1987. Aktuálna verzia je VxWorks 7. Produkty spoločnosti Wind River využíva aj NASA vo svojom vesmírnom programe. VxWorks má všetky vlastnosti moderného operačného systému. Je natívnym 64-bitovým operačným systémom a podporujú multitasking. Podporuje symetrický aj

asymetrický multiprocessing. Na zachytenie a spracovanie chýb využíva vlastný framework. Využíva semafore s technikou dedenia priorít čím zabraňuje vzniku známeho problému s inverziou priorít. Taktiež podporuje medzi procesovú komunikáciu. VxWorks poskytuje aj POSIX rozhranie. Na prácu so súbormi môže používať viac typov súborových systémov.

VxWorks plánovače

Operačný systém VxWorks podporuje viacero druhov plánovačov:

- VxWorks natívny plánovač, ktorý podporuje dva typy plánovania. Prvým je preemptívne plánovanie na základe priorít a druhým round-robin plánovanie.
- POSIX plánovač, ktorý podporuje všetky funkcie VxWorks natívneho plánovača a navyše pridáva funkcionality plánovania vlákien v užívateľskom priestore.
- framework pre nahradenie plánovača umožňuje užívateľovi implementovať vlastný plánovač.

VxWorks natívny plánovač sa používa ako štandardný plánovač pre operačný systém. Plánovač môže byť nastavený pre použitie preemptívneho plánovania na základe priorít alebo pre round-robin plánovanie. Jadro rozlišuje 256 úrovni priorít. Najvyššiu prioritu má úroveň 0, najnižšiu 255. POSIX plánovač rozširuje funkcie natívneho plánovača o vlastnosti potrebné pre plánovanie vlákien v užívateľskom priestore. Natívny plánovač vykonáva svoju činnosť na úrovni celého systému. Na rozdiel od neho, POSIX plánovač dokáže pracovať na dvoch úrovniach, teda vlákna dokáže plánovať aj na úrovni samotného procesu. Ďalším rozdielom oproti natívnemu plánovaču je obrátená stupnica úrovni priorít. Úroveň 0 má najnižšiu prioritu a úroveň 255 najvyššiu [10].

Framework pre nahradenie plánovača poskytuje užívateľovi možnosť nahradiť používaný plánovač vlastným riešením. Keďže plánovač tvorí kritickú časť operačného systému, pri použití vlastného plánovača treba podrobne otestovať správanie celého systému. Príklad takejto operácie môžeme nájsť v práci, ktorá bola vykonaná na technickej univerzite v Eindhovene. Cieľom práce bolo nahradiť natívny plánovač VxWorks vlastným riešením využívajúcim hierarchické plánovanie [11]. VxWorks je poskytovaný v rôznych obmenách pre rôzne skupiny zariadení. Tieto špecifické skupiny sa nazývajú VxWorks platformy. Sú to tieto:

- univerzálna,
- pre priemyselné zariadenia,
- medicínske zariadenia,
- sieťové vybavenie,
- automobilový priemysel,
- spotrebné zariadenia.

VxWorks ponúka aj certifikované platformy, ktoré sú určené pre zariadenia, ktoré musia spĺňať rôzne bezpečnostné certifikáty. Tento operačný systém podporuje procesorové architektúry typu x86, ARM, MIPS a PowerPC. V minulosti bol systém VxWorks implementovaný aj na iné architektúry. Na vývoj aplikácií sa využíva VxWorks Workbench. Jedná sa o vývojové prostredie založené na Eclipse

technológiách. Podporuje všetky bežné projektové funkcionality ako sú napríklad vytváranie a buildovanie projektov, vytvorenie komunikácie s cieľovým systémom, konfigurácia, debugging a monitoring samotného operačného systému ako aj aplikácii bežiacich na cieľovom systéme. Na vytváranie aplikácii je možné použiť jazyky C/C++. Taktiež je možné použiť rôzne druhy kompilátorov.

Záver

Cieľom tohto príspevku bolo urobiť prehľad dostupných riešení v oblasti operačných systémov reálneho času. Takýto operačný systém je možné vytvoriť už z existujúceho systému pomocou nadstavby reálneho času. Takéto riešenie nám zväčša poskytuje výhodu známeho užívateľského prostredia ako aj možnosť použiť už overené vývojárske nástroje. Pre systémy typu Windows je to RTX. Keďže sa jedná o profesionálny komerčný produkt, užívateľ má pri inštalácii a používaní produktu prístup k rozsiahlej podpore. Pre systémy Linux sme spomenuli RTAI, PREEMPT RT a Xenomai. Tieto riešenia sú voľne dostupné avšak inštalácia býva často zložitejšia a vyžaduje veľmi skúseného užívateľa. Podpora takýchto produktov je vytvorená pomocou tzv. mailing listov, kde si užívatelia vymieňajú skúsenosti. Druhou možnosťou je využiť kompletný operačný systém ako napríklad Wind River Linux alebo VxWorks. Užívateľ dostane kompletný produkt bez nutnosti zložitejšej inštalácie, s rozsiahlou podporou a nakonfigurovaný priamo na cieľový hardvér. Cena takýchto riešení je samozrejme veľmi vysoká.

Podakovanie

Tento príspevok bol podporený Agentúrou na podporu výskumu a vývoja projektom APVV-0211-10.

Referencie

1. Murgaš, T., Fodrek, P. a Farkas, L. Networked control system using Linux real time application interface. Bratislava : WSEAS, 2011. ISBN 978-1-61804-057-2.
2. Microsoft. Scheduling Priorities (Windows). MSDN Microsoft. [Online] [Dátum: 14. December 2012.]
[http://msdn.microsoft.com/en-us/library/windows/desktop/ms685100\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms685100(v=vs.85).aspx)
3. IntervalZero. RTX 2012 Product Brief. Waltham : IntervalZero, 2012.
4. Linux Foundation. What Is Linux: An Overview of the Linux Operating System. linux.com. [Online] [Dátum: 17. December 2012.]
<https://www.linux.com/learn/new-user-guides/376-linux-is-everywhere-an-overview-of-the-linux-operating-system?showall=1>
5. Mantegazza, P., Dozio, E.L. a Papacharalambous, S. RTAI: Real Time Application Interface. Houston : Beltown Media, 2000. Zv. 72, 10.
6. Fu, L. a Schwebel, R. RT PREEMPT HOWTO. Real-Time Linux Wiki. [Online] [Dátum: 18. December 2012.]
https://rt.wiki.kernel.org/index.php/RT_PREEMPT_HOWTO
7. Ugal, A.S. Hard Real Time Linux using Xenomai on Intel Multi-Core Processors. s.l. : Intel Corporation, 2009
8. FSM Labs, Inc. Getting Started with RTLinux. s.l. : FSM Labs, Inc, 2001
9. Yodaiken, V. The RTLinux Manifesto. Socorro : New Mexico Institute of Technology

-
10. Wind River. VxWorks Kernel Programmer's Guide. Alameda : Wind River Systems, Inc., 2005
 11. Behman, M., a iní, a iní. Towards Hierarchical Scheduling in VxWorks. Vasters, Sweden : IEEE, 2009
-

Spoluautorom článku je prof. Ing. Ján Murgaš, PhD.
