

Sistemi operativi real-time: nuova evoluzione in corso

Architetture multi-core e virtualizzazione, elaborazione di contenuti multimediali e nuovi requisiti di sicurezza sono fra le sfide tecnologiche da affrontare per i sistemi operativi real-time embedded di ultima generazione

Giorgio Fusari

Un hardware sempre più complesso e distribuito in contesti tecnologici che spesso oggi vanno oltre gli ambienti classici (ufficio, fabbrica) sta portando a un nuovo processo evolutivo nello sviluppo dei sistemi operativi in tempo reale, conosciuti anche come RTOS (Real-Time Operating Systems). In sostanza, i requisiti che questi sistemi, sia di tipo proprietario sia open source, devono soddisfare per controllare in modo adeguato le varie applicazioni si innalzano e diversificano ulteriormente.

“Se solo pochi anni fa – spiega Davide Ricci, technical account manager di Wind River - caratteristiche come scalabilità, determinismo, hard real-time, safety e protezione della memoria sembravano essere un ambizioso traguardo per i sistemi operativi real-time. Oggi la maggior parte degli utilizzatori le considerano ormai scontate. Una delle sfide alle quali gli sviluppatori devono oggi rispondere passa per la crescente complessità dell'architettura dei sistemi, spesso equipaggiati con più core per soddisfare ad esempio la necessità di elaborare contenuti multimediali. Questa necessità conduce alla seconda sfida, ossia quella di affiancare a sempre più affidabili meccanismi di IPC (Inter-Process Communication, ndr.), delle media-library in grado di gestire schermi multipli e applicazioni di grafica 3D”. Senza poi considerare la possibilità di avere più sistemi operativi in esecuzione su diversi core (virtualizzazione), utilizzandone uno per le problematiche più strettamente real-time (OS master) e l'altro per la gestione dei contenuti multimediali o delle applicazioni non critiche (OS guest). “Tutto questo avviene in un contesto di alta competitività dei mercati, che



Fig. 1 - Cesare De Siena, embedded manager per l'Italia di Abacus ECC

accorcia di gran lunga il time-to-market dei prodotti e rende necessaria l'adozione di standard e il supporto attraverso COTS (Commercial Off-The-Shelf, ndr.)”.

Sulla nuova fisionomia degli RTOS di ultima generazione influisce anche, secondo Filippo Ferrarese, responsabile tecnico di Microtask Embedded, la diffusione sempre più ‘pervasiva’ di funzioni intelligenti nei dispositivi elettronici, che si fonde oggi con la necessità di dover accedere alle loro interfacce tramite la Rete.

L'esplosione del numero di nodi in linea che sta portando alla migrazione verso architetture a più ampia ‘numerabilità’ come IPV6 è, spiega, sostanzialmente pilotata dalla necessità di collegare in automatico sensori dei tipi più diversi, o ‘agenti’ intelligenti con compiti di supervisione, raccolta dati o monitoraggio. Per contro però, proprio una distribuzione così spinta richiede di ripensare gli aspetti legati alla sicurezza nei sistemi embedded. “Mai come oggi, un dispositivo intelligente che produce o consuma dati scambiati sulla Rete deve essere protetto contro accessi non controllati o contro i tentativi di sostituire flussi di dati corretti con informazioni truffaldine.

La strategia del double licensing

In quali casi conviene scegliere un RTOS proprietario o uno open source? Paolo Gai di Evidence vede l'uso di sistemi commerciali 'closed source' come soluzione adatta in applicazioni 'safety critical' di dimensioni medio/piccole con particolari esigenze di certificazione e robustezza: "Per tutte le altre applicazioni - precisa - l'opzione open source è sicuramente la più attraente". Ricorda però di fare attenzione e non confondere open source' con 'gratuito'. "L'utilizzo di sistemi open source non è di fatto gratuito per le aziende, che devono sviluppare know-how internamente o appoggiarsi ad aziende qualificate per la 'customizzazione' del prodotto e il supporto tecnico alla risoluzione dei problemi. Per questo motivo, esistono varie aziende sul mercato che propongono soluzioni basate su versioni personalizzate di Linux, ben collaudate e irrobustite per l'uso industriale". Inoltre i due mondi, dice Gai, di fatto sono destinati a convivere. "Lato sistemi 'grandi', vediamo senza dubbio come trend crescente l'utilizzo di piattaforme Linux che permettono di padroneggiare il sistema e avere a disposizione una quantità enorme di software senza dover pagare royalty per pezzo. Lato sistemi minimali a microcontrollore, vediamo la coesistenza di sistemi proprietari e open source". In questo ambito la strategia di Evidence è fare double licensing, fornendo il sistema operativo Erika Enterprise sia con licenza open source che commerciale e lasciando all'utente la decisione sulla tipologia di licenza per il proprio software. "Inoltre il double licensing agevola la ricerca universitaria, permettendo di accedere a tecnologie innovative con un costo molto basso".

Fig. 2 - Paolo Gai, amministratore delegato di Evidence

Fig. 3 - Marco Cavallini, socio fondatore di Koan Software

Credo che il dominio della sicurezza, cioè la security, sia quello su cui occorra lavorare di più, soprattutto se l'accesso illegale a un dispositivo può provocare anomalie in grado di mettere a repentaglio l'incolumità fisica delle persone, ossia la safety. Basti pensare alle reti di distribuzione di elettricità, gas, acqua, o a quelle di alcuni sistemi di supervisione, ad esempio del traffico aereo, stradale, ferroviario; o alle reti presenti nei cosiddetti 'edifici intelligenti' o, più prosaicamente, alle reti di fabbrica. In sintesi condenserei nel termine 'sicurezza', inteso come

 **Fenway**
Embedded Systems

HW&SW development products

**Build Your
Next Project
with Confidence**

**Small, Efficient,
Integrated, Supported**



Quadros[™]
Systems Inc.
<http://www.quadros.com>



 **Fenway**
Embedded Systems

Via Don Giovanni Minzoni, 31 • 20010 Arluno (Mi) • Italy
tel. +39 02 97310120 • <http://www.fenwayembedded.com>
readerservice.it n.17694



Fig. 4 - Filippo Ferrarese, responsabile tecnico di Microtask Embedded

‘safety e security’, i trend di mercato e le richieste di nuovi contenuti che stiamo osservando attualmente. Dal punto di vista dei sistemi operativi real-time questo vuol dire fornire sempre di più ‘componenti’ sicure

come IPsec, SSL, SSH, IKE, Radius e al contempo indirizzare gli aspetti di safety con tecniche come il partizionamento della memoria, la segregazione dei device driver e dei protocolli, e la certificazione in accordo con standard industriali riconosciuti, come ad esempio IEC61508.

A che punto è l'open source

E in merito agli RTOS basati su codice aperto cosa va detto? Marco Cavallini, socio fondatore di Koan Software, nonché consulente e collaboratore in diversi progetti open source, ricorda che nell'ambito di questi RTOS il primo traguardo da raggiungere è l'affidabilità, da cui poi dipende la garanzia sulla fiducia degli utenti: “Nella pratica ci si scontra ancora con la diffidenza tipica di chi si è avventurato in passato verso problematiche tipiche dei sistemi real-time, improvvisando soluzioni basate su sistemi operativi non adatti a questo scopo

L'hard real-time di Linux? “Non è deterministico”

Per Davide Ricci di Wind River il mercato industriale dell'automazione e del controllo fa più fatica ad avvicinarsi al Pinguino proprio a causa dei requisiti real-time che il kernel Linux non soddisfa pienamente. “Il real-time di Linux non è infatti un hard real-time deterministico e per quanto il modello di preemption RT, ossia PREEMPT_RT, abbatta notevolmente i tempi di risposta - si parla di 25 microsecondi come latenza massima - queste stime rimangono pur sempre non deterministiche e pertanto inadatte ad applicazioni in cui i tempi di risposta sono critici.

Per questo motivo la scelta del sistema operativo nelle applicazioni del mercato industriale, e in particolare per il controllo, viene valutata caso per caso”.



Fig. 5 - Davide Ricci, technical account manager di Wind River

Fig. 6 - Il sistema open source KaeiLOS, fornito da Koan Software



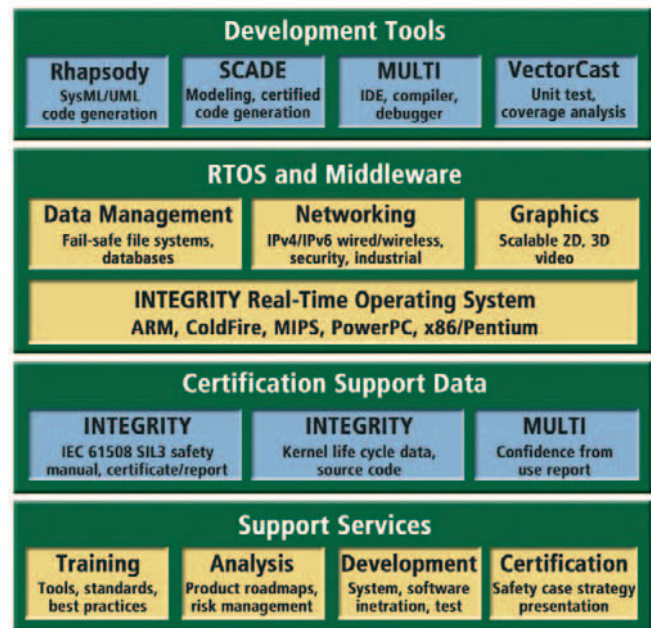
come DOS o Windows. I sistemi RTOS open source già disponibili forniscono l'affidabilità necessaria a un impiego ‘mission-critical’, a patto di rispettare certe condizioni, quali l'utilizzo degli strumenti giusti e delle estensioni real-time atte ad assicurare tale robustezza”.

Inoltre Cavallini sottolinea un altro punto: “Le richieste più frequenti dei nostri utenti hanno come obiettivo la gestione di sistemi d'automazione attraverso un unico elemento, in cui siano concentrate tutte le funzionalità del sistema: cioè in sostanza ottenere un PC che permetta di far girare un RTOS, l'interfaccia utente (HMI - Human-Machine Interface, ndr.) e la logica di controllo senza bisogno di ulteriori componenti esterni”.



Fig. 7 - Microsoft Windows Embedded CE 6.0

Fig. 8 - Le funzionalità dell'Rtos Integrity di Microtask



Anche Paolo Gai, amministratore delegato di Evidence, sottolinea che l'utilizzo di varianti real-time di Linux permette d'integrare interfaccia operatore e controllo industriale all'interno dello stesso sistema, fornendo interessanti riduzioni di costo per la produzione di sistemi di controllo industriale. E, per quanto riguarda i dispositivi embedded 'più grandi', come i sistemi di controllo industriale, supervisione e visualizzazione (anche multimediale) che tipicamente utilizzano un sistema operativo Linux per ridurre i propri costi di produzione, auspica sicuramente in termini di traguardo da raggiungere una maggiore penetrazione di Linux come sistema operativo di riferimento: "Tale scelta - spiega Gai - porta come vantaggio immediato l'utilizzo di un sistema robusto e sicuro, libero da royalty, ma soprattutto permette la creazione di competenze tecnologiche che possano essere interamente dominate all'interno del territorio italiano; competenze che con il tempo vanno ad arricchire la competitività del sistema Italia".

Un diverso ambito è poi quello dei sistemi operativi minimali, concepiti per i dispositivi a microcontrollore: "Per quanto riguarda questi ultimi - aggiunge - i traguardi da raggiungere riguardano soprattutto il supporto efficiente per le nuove architetture multi-core, il supporto per i sistemi basati su hardware riconfigurabile come gli FPGA, e l'utilizzo di generatori automatici di codice per permettere di ridurre i tempi di sviluppo e testing del software, grazie a un passaggio indolore dalla simulazione degli algoritmi di controllo al prodotto finito".

Accennando in generale al mondo degli RTOS, Cesare De

Una soluzione "adattabile"

Tutte le soluzioni software sviluppate da Quadros Systems hanno un comune denominatore nel sistema operativo real-time RTXC. Sviluppato verso la fine degli anni '70, RTXC è stato uno dei primi RTOS a fare la sua comparsa sul mercato. Verso la fine degli anni '90 è stato introdotto RTXC Quadros, un sistema operativo di nuova generazione che ha contribuito a ridefinire il panorama del settore. Il RTOS RTXC Quadros è una soluzione caratterizzata da un elevato grado di affidabilità che mette a disposizione quattro varianti che garantiscono significativi vantaggi per ciascuno dei principali modelli di elaborazione:

- RTXC/ss (a stack singolo): si tratta di un executive specializzato che impiega "threads" a commutazione veloce per il supporto di applicazioni di elaborazione del segnale e caratterizzate da elevata velocità di trasferimento dati;
- RTXC/ms (multi-stack): ottimizzato per l'elaborazione del controllo utilizzando uno scheduled di tipo pre-empitivo, di tipo a priorità e basato su eventi;
- RTXC/dm (dual mode) che abbina RTXC/ms e RTXC/ss ed è ideale per applicazioni di elaborazione convergente;
- RTXC/mp (multiprocessing) che abbina RTXC/ms e RTXC/dm in differenti combinazioni per il supporto di elaborazione multicore e multiprocessore.

Siena, embedded manager per l'Italia di Abacus ECC, più che di traguardi da raggiungere, preferisce parlare di un processo di miglioramento continuo, basato sulle richieste degli utenti. "Attualmente le principali esigenze che essi manifestano in tutti i settori applicativi sono un'ulteriore riduzione del time-to-market, una maggiore integrazione tra le diverse tecnologie, un migliore sfruttamento degli investimenti hardware e software, attraverso il loro riutilizzo su più progetti, e un ulteriore miglioramento della stabilità degli RTOS".

Ultime funzionalità

Nel mondo degli RTOS Windows CE 6.0, afferma De Siena, è un prodotto che permette di avere costi certi fin dall'inizio del progetto, di ridurre i costi di manutenzione, difendere la proprietà intellettuale dell'utente e avere maggiore libertà nella scelta dei consulenti. Le innovazioni introdotte nella versione 6.0 sono numerose: prima di tutto la comodità per lo sviluppatore di avere un sistema integrato in cui sono unificati l'ambiente di sviluppo del sistema operativo e l'ambiente di sviluppo degli applicativi. Sono stati introdotti anche un nuovo file system, più robusto rispetto al precedente, e due modalità per i driver: ora c'è anche il livello kernel (blindato) in aggiunta a quello user, al fine di aumentare tanto le prestazioni quanto l'affidabilità del sistema. Anche l'integrazione della funzione

Digital Video Recording punta a fornire un sistema operativo completo, che il programmatore può usare, ad esempio, sia per creare macchine industriali a controllo numerico real-time che utilizzano l'ispezione ottica nelle linee di produzione, sia per progettare set-top-box destinati all'utenza consumer. In effetti, sottolinea De Siena, la caratteristica chiave della versione 6.0 è proprio la sua possibilità di utilizzo 'trasversale' in qualunque tipo di settore.

Anche KaeilOS, il sistema Linux open source interamente realizzato in Italia dalla software house Koan Software, ha avuto alcuni aggiornamenti riguardanti le estensioni real-time RTAI e Xenomai, su cui è basato. Nella prossima versione di KaeilOS verrà integrato il supporto real-time anche per alcuni processori della famiglia ARM, in particolare il modello AT91RM9200.

Nell'RTOS Integrity, Ferrarese segnala, a partire dalla versione 10 'Anniversary edition', il supporto per architetture SMP (Symmetric Multiprocessing), disponibile per le nuove piattaforme multi-core embedded come Freescale Power MPC8641D. In questa versione, il supporto Smp indirizza anche il bilanciamento di carico tra le CPU controllate, consentendo la 'migrazione' di task tra processori se il tempo di Cpu non è equamente distribuito. Viene inoltre sottolineato il supporto per architetture NUMA (Non-Uniform Memory Architecture) in cui, diversamente da una struttura Smp 'pura', i processori hanno proprie aree di memoria ad alta velocità in cui le applicazioni possono essere eseguite in modo segregato. Infine la versione 2.2 di u-

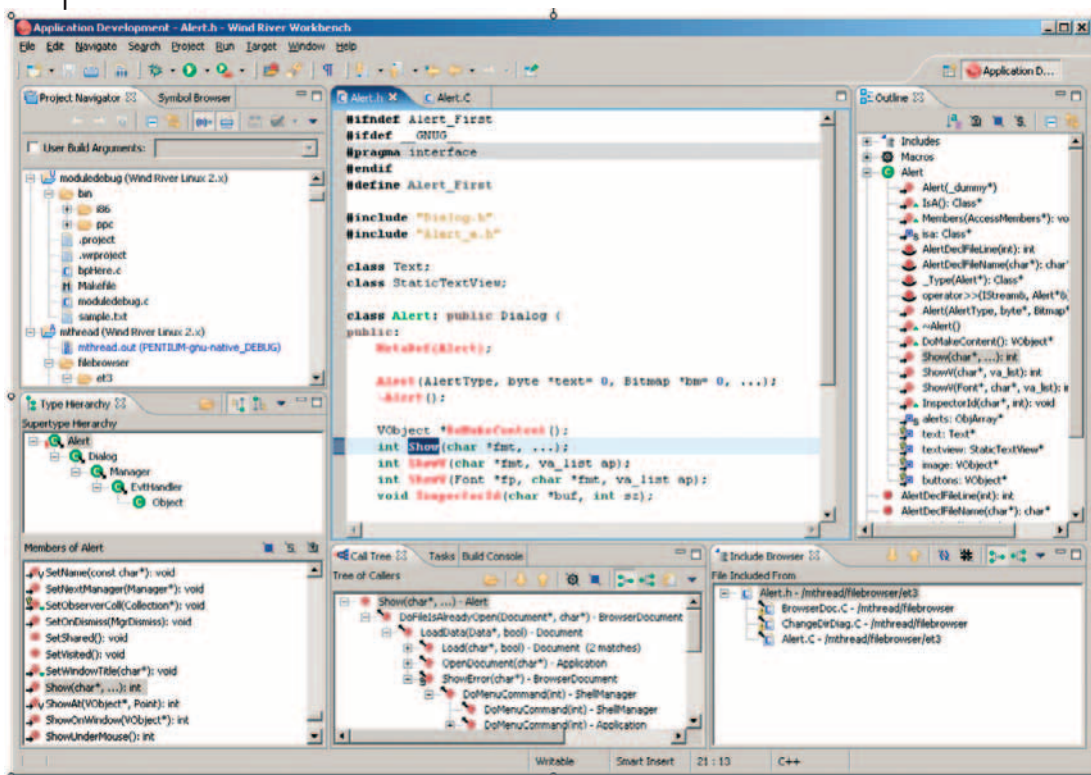


Fig. 9 - Una schermata della suite di sviluppo Wind River Workbench



Fig. 10 - Le schede Flex prodotte da Evidence ed Embedded Solutions

velOSity, un RTOS a consumo di risorse molto contenuto, fornisce ora un file system con formato 'FAT-compatibile', un flash device manager con wear-leveling e caratteristiche fault-tolerant; uno stack USB 1.1 e 2.0, uno stack TCP/IP e il supporto per architetture MIPS.

Per quanto riguarda Wind River Linux, arrivato alla release 1.4, ad oggi, spiega Ricci, le patch che abilitano il modello di preemption RT (PREEMPT_RT) e gli High Resolution Timer sono incluse nelle distribuzioni e validate per tutte le board di riferimento Intel 32 e alcune PowerPC e ARM. Obiettivi in roadmap sono la validazione di queste patch per tutte le board di riferimento e anche il pieno supporto per i sistemi multi-core e le architetture a 64 bit.

In VxWorks è stata invece di recente raggiunta la conformità 100% POSIX (Portable Operating System Interface), nonché la certificabilità DO-178B e IEC 61508. Il numero di architetture pienamente supportate incorpora ora tra le altre ARM946, Freescale PowerPC e MIPS BroadCom BCM91480. Da citare anche il supporto per display multipli e l'integrazione di una media-library OpenGL. Wind River ha inoltre in fase di testing (attualmente per architetture PowerPC) la cosiddetta 'system virtualization', ossia la possibilità di avere Linux in esecuzione come sistema operativo 'guest' su master VxWorks.

Per quanto riguarda i sistemi basati su Linux, spiega Gai, Evidence sta lavorando per integrare all'interno delle proprie distribuzioni il supporto per una compilazione ed emulazione del codice lato host. In particolare, la società sostiene l'utilizzo di Scratchbox, un progetto open source finanziato da Nokia che semplifica lo sviluppo di piattaforme Linux per sistemi embedded. Supporta inoltre sistemi di pacchettizzazione degli applicativi basati su standard Debian, permettendo ai propri clienti di accedere ad una vasta libreria di software disponibili nel mondo Linux. Infine fornisce supporto diretto all'inte-

grazione e al debugging applicativo tramite i tool di sviluppo Trace32 di Lauterbach, per sistemi embedded.

“Per quanto riguarda i sistemi minimali basati su Erika Enterprise – aggiunge Gai – le novità vanno dal rila-

scio di Erika Enterprise in double licensing, cioè licenza GPL e licenza commerciale, all'integrazione di questo sistema operativo in un ambiente di sviluppo semplificato basato su Eclipse, con la possibilità di generare automaticamente il codice di configurazione del sistema operativo, utilizzando standard industriali proposti dal consorzio OSEK/VDX in ambito automotive. Infine, Erika Enterprise supporta direttamente architetture multi-core basate su Altera Nios II, permettendo di partizionare il software in modo semplice su diversi microcontrollori ospitati nello stesso chip”.

Nei prossimi mesi Evidence intende poi realizzare numerose innovazioni all'interno della propria offerta di sistemi operativi real-time: “Primo tra tutti, vogliamo citare il supporto per le nuove schede Flex prodotte da Evidence ed Embedded Solutions. Flex è una scheda elettronica basata su microcontrollori dsPIC(R) di Microchip, che permette di realizzare prototipazione elettronica in modo semplice e a basso costo. Infine, stiamo collaborando con l'università SUPSI di Lugano, in Svizzera, all'integrazione e al supporto per la generazione automatica di codice per sistemi di controllo industriale, utilizzando tool open source come Scilab e Scicos”.

readerservice.it

Abacus ECC	n. 25
Evidence	n. 26
Koan Software	n. 27
Microtask Embedded	n. 28
Quadros Systems (Fenway Embedded Systems)	n. 29
Wind River Systems	n. 30