

Pillole di Linux nell'automazione

Il nuovo kernel 2.6 offre agli utenti industriali un sistema operativo Linux ancora più efficiente

MASSIMO GIUSSANI

E' tutt'altro che infrequente, oggi, imbattersi in SBC che adottino Linux come sistema operativo. Altre applicazioni di controllo, ad esempio per la gestione di centrali telefoniche o apparecchiature per la domotica e la building automation traggono vantaggio dalle intrinseche capacità di comunicazione di questo OS. Si possono trovare in commercio robot gestiti da Linux e la crescente diffusione di Ethernet come mezzo di comunicazione sul campo rappresenta un ulteriore motivo per appoggiarsi a un sistema operativo con un'eccellente gestione dello stack TCP/IP. Ora, con i miglioramenti introdotti nel kernel 2.6 si prefigura un salto di qualità anche delle applicazioni industriali meno spinte.

Linux dispone di una struttura modulare che ben si presta alla realizzazione di sistemi minimali su misura di applicazione. Il cuore del sistema operativo è rappresentato da un microkernel (contenente i servizi di temporizzazione e di gestione dei processi e della memoria) al di sopra del quale vengono a trovarsi i moduli per le comunicazioni di rete e per la gestione dei file-system.

La possibilità di includere i driver di periferica, sia durante la compilazione del kernel, sia in seguito sotto forma di moduli aggiuntivi, permette la realizzazione di sistemi con un ridotto impiego di risorse, comunque aperti a un'espansione futura in base alle esigenze dell'utente.

La nuova versione del kernel rappresenta una delle evoluzioni più interessanti dal punto di vista dell'utilizzatore industriale.

Quattro novità che meritano di essere menzionate in questa sede sono: l'estensione dell'hardware supportato e l'introduzione del concetto di sub-architettura; la pre-empibilità del kernel; la riscrittura dello scheduler; la migliorata gestione delle sincronizzazioni.

Quattro punti vincenti

Linux potrà girare in multitasking su microcontrollori privi di unità di gestione della memoria, aprendo così la strada all'impiego in sistemi caratterizzati da maggior semplicità ed economicità. Tra i microcontrollori dotati di MMU, invece, va citato il supporto ai core Cris di Axim Communications, rivolti alle applicazioni di rete. La lista dei microprocessori coperti è stata ampliata con le CPU Motorola di classe 68k (Dragonball e ColdFire) e i sistemi Hitachi H8/300 e Nec v850. Il supporto alle architetture x86 di Intel e AMD, nonché PowerPC, prosegue con le architetture a 64 bit (Itanium64, Opteron e Ppc64). E' stato introdotto anche il supporto per i processori Crosue, di Transmeta, e per sistemi multiprocessore x86 con architettura Voyager di NCR. Considerando che molti degli SBC utilizzati nel mondo dell'automazione si rifanno ad architetture simili a quelle di un PC tradizionale, ambiente ideale per la distribuzione mainstream di Linux, non vi è da stupirsi della diffusione del supporto per questo sistema operativo fra i principali produttori. Linux può essere trovato su piattaforme PC104, compactPCI e VME, oltre che su diversi formati proprietari. Tra le novità introdotte con la nuova versione spicca anche il concetto di sub-architettura: i diversi componenti del sistema Linux sono stati disaccoppiati in modo da poterne consentire la modifica o la sostituzione individuale riducendo al minimo le alterazioni da apportare agli altri moduli. Gli utilizzatori industriali potranno sfruttare questa separazione per realizzare kernel che si adattino all'hardware utilizzato nelle applicazioni di automazione e controllo. E' inoltre possibile eliminare completamente dal sistema il codice relativo al supporto di video, tastiera e dispositivi di puntamento, riducendo l'occupazione di memoria dei sistemi 'headless'.

Sotto il profilo del kernel, la novità di maggior rilievo è

indubbiamente la sua pre-empibilità: l'esecuzione può essere interrotta mentre uno dei task programmati dallo scheduler viene elaborato dalla CPU. Questo permette a processi a priorità più elevata di chiedere la sospensione di un determinato task a priorità inferiore, in modo da venire serviti prima del completamento del task. Anche lo scheduler degli I/O è stato completamente riscritto per ridurre i tempi richiesti a un dato processo per accedere alle periferiche d'ingresso e uscita. Non si può poi non citare la significativa estensione del supporto nativo alle funzioni Posix per thread, segnali e timer di alta precisione.

Alternative per tutti i gusti

Le distribuzioni commerciali di Linux sono molteplici e ciascuna di esse si contraddistingue per il diverso taglio applicativo e per l'offerta degli strumenti di sviluppo e configurazione, nonché delle condizioni di assistenza e di supporto tecnico che costituiscono il vero valore aggiunto in grado di influenzare le scelte degli utenti.

Le diverse varianti open source richiedono investimenti iniziali maggiori in termini di formazione e conoscenza ma, una volta 'incamerato' il know-how necessario allo sviluppo delle applicazioni, presentano l'indubbio vantaggio di affrancare il produttore dalle royalty sul prodotto finito.

Da una recente ricerca effettuata da LinuxDevices.com sulle distribuzioni più utilizzate nel segmento embedded emerge un trend interessante, ossia la crescente tendenza degli sviluppatori a creare un sistema operativo su misura basandosi sul codice sorgente messo a disposizione dalla comunità open source. Una selezione delle distribuzioni di Linux è riportata in tabella. Gli utenti che necessitano un kernel hard real-time possono appoggiarsi a una delle distribuzioni commerciali o gratuite appositamente modificate per fornire prestazioni deterministiche (anche se, va detto, difficilmente si vedrà usare il termine 'hard' per descrivere le prestazioni in tempo reale di Linux). Sono essenzialmente due i modi di modificare il sistema. Il primo è storicamente legato a KURT Linux (Kansas University Real-Time Linux): una patch altera il meccanismo di temporizzazione all'interno del kernel per garantire risoluzioni temporali dell'ordine di pochi microsecondi. La distribuzione commerciale Hard Hat Linux di Montavista, che ha ispirato alcune delle

modifiche al kernel ufficiale 2.6, adotta questa metodologia. Il secondo approccio, invece, utilizza un secondo kernel, il quale controlla la gestione degli interrupt e assegna una priorità superiore ai task in tempo reale. Le distribuzioni Open Source RTAI (messo a punto dal dipartimento d'ingegneria aerospaziale del Politecnico di Milano) e RT-

Distribuzioni Linux

Produttore	Distribuzione
Commerciali	
FsmLabs	RT-Linux Pro www.fsmlabs.com
Koan	K-Linux www.koansoftware.com/it/prd_embe_klinux.htm
Metrowerks	Embedix e LXRT www.metrowerks.com/MW/Develop/Embedded/Linux/default.htm
LinuxWorks	BlueCat e BlueCatRT www.bluecat.com
Montavista	Hard Hat Linux, RT-Linux www.mvista.com
Red Hat	Embedded Linux www.redhat.com
RidgeRun	DSP Linux www.ridgerun.com
TymeSys	LinuxRT www.timesys.com
Vital Systems	V-Linux www.armlinux.net
Open Source	
Debian	Embedded Debian Project www.emdebian.org
DiaPM	RTAI www.aero.polimi.it/~rtai
Kansas University	KURT www.ittc.ku.edu/kurt
New Mexico University	RT-Linux www.fsmlabs.com
Prosa	ETLinux www.prosa.it/etlinux
uCLinux	uCLinux www.uclinux.org

Linux (inizialmente sviluppato dall'università del New Mexico) sono due importanti esempi di applicazione di questa tecnica. Entrambe le distribuzioni sono disponibili gratuitamente in rete e presentano delle controparti commerciali che offrono, con il supporto e gli strumenti di sviluppo, anche delle evoluzioni di questi sistemi operativi in grado di operare direttamente dallo spazio utente. RTAI con LXRT offerto da Lineo ed RT-Linux come distribuzione commerciale messa a punto da FsmLabs. ■