Open software, successo in crescita

Linux e il codice aperto fanno sempre più parte dei nuovi progetti embedded. Attenzione però: per evitare fallimenti occorre conoscerne bene le peculiarità

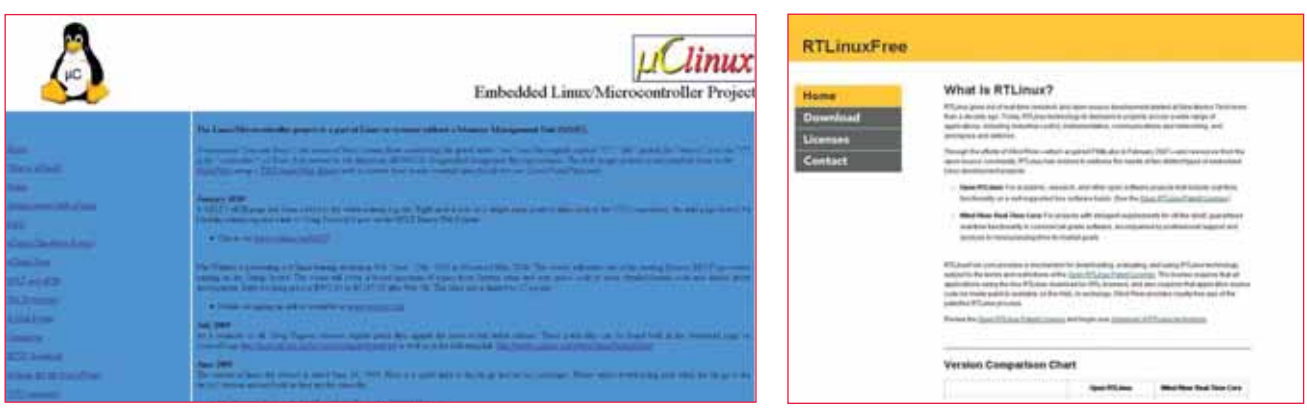
Giorgio Fusari

Ne è stata fatta di strada dal 1991, quando Linux, cioè il kernel creato da Linus Torvalds, cominciò a essere integrato con altro software (librerie, componenti) del progetto Gnu per dare origine a quelli che sarebbero diventati i primi sistemi operativi open source e le prime distribuzioni. Col tempo, sebbene fosse stato in principio concepito per un utilizzo soprattutto su macchine server, gradualmente il Pinguino ha cominciato a colonizzare anche le postazioni desktop e, negli ultimi anni, la sua adozione è cresciuta anche all'interno di molti progetti, sistemi e dispositivi embedded.

Nel mondo embedded, i vantaggi alla base di questo successo di Linux sono normalmente percepiti dagli utenti soprattutto come benefici in termini di costi delle licenze, o flessibilità di azione derivante dalla possibilità per lo sviluppatore di accedere liberamente al codice sorgente. Insieme, questi fattori rendono la soluzione open source di grande interesse per i costruttori di sistemi embedded.

Gli utenti però allo stesso tempo, all'avvio di un progetto, si interrogano su vari punti critici: quali sono i possibili problemi di compatibilità hardware; per quali settori o applicazioni potrebbe essere più vantaggioso usare un sistema operativo open source come Linux; quale eventuale percorso di migrazione occorrerebbe seguire, o quali sistemi sarebbe meglio utilizzare per soddisfare i requisiti di progetto. Gli utenti si chiedono anche a quali rischi e inconvenienti progettuali possono andare incontro nel lungo periodo scegliendo questo modello di sviluppo.

In effetti, nonostante gli analisti di mercato dicano da più parti che il sistema operativo Linux è utilizzato in modo crescente nei progetti di software embedded e sta diventando la scelta predominante per i nuovi progetti di sviluppo rispetto a ogni altro sistema operativo embedded, permangono ancora diversi ostacoli che rendono difficile agli ingegneri del software realizzare con la necessaria rapidità prodotti embedded da introdurre nel mercato rispettando i vincoli di time-to-market. Questi ostacoli sono ad esempio connessi alla necessità di creare un ambiente di sviluppo, di eseguire un'integrazione delle applicazioni custom, di ottimizzare l'hardware target o di ottenere assistenza e supporto tecnico per il sistema operativo in questione.



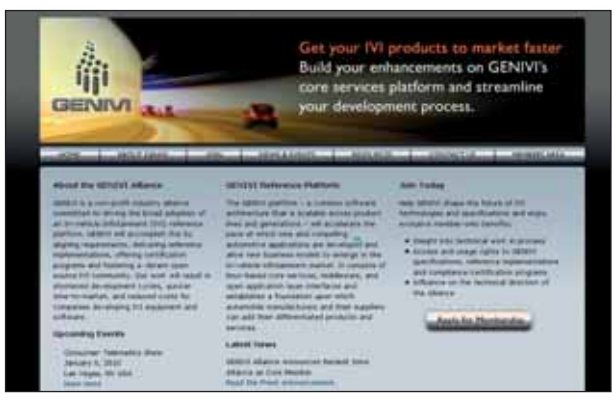
Figg. 1 - 2 - uClinux e Rtlinux, due sistemi operativi open source per i sistemi embedded

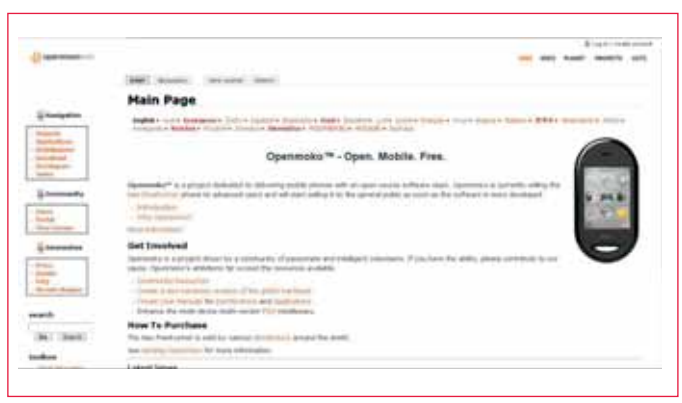
Open source: offerta ampia

Molte sono le tipologie di sistemi operativi open source disponibili sul mercato, sia in versione commerciale che scaricabili liberamente da Internet. Vari di questi sono ottimizzati per l'utilizzo in applicazioni con specifici requisiti di funzionamento. Ad esempio, Clinux è scaricabile da repository e indicato per microcontrollori e sistemi non dotati di Memory management unit (Mmu). RTLinux è invece un Rtos (Real-time operating system) con microkernel di tipo 'hard real-time' e reperibile sia come prodotto commerciale (attraverso la casa creatrice Fsmlabs), sia in versione 'free' (http://www.rtlinuxfree.com/) tramite Wind River, che ha acquisito la società nel 2007. Sempre nel dominio dei prodotti commerciali, Montavista Linux viene rilasciato in varie versioni: dalla Professional Edition, adatta per applicazioni in networking e comunicazioni, strumentazione e controllo, aerospazio e difesa, dispositivi Soho ed elettronica medicale, a quella Carrier Grade, concepita per le telecomunicazioni; a quella 'mobile' (Mobilinux), indirizzata ai telefoni cellulari e ad altri dispositivi wireless. BlueCat Linux è un'implementazione indirizzata a sistemi embedded che vanno dai piccoli dispositivi di fascia consumer ai sistemi multi-cpu di grande potenza.
KaeilOS è una distribuzione Linux embedded, Gpl open source e 'royalty free' per applicazioni industriali, con strumenti di sviluppo e debugging specifici.

Confronto fra sistemi

Un confronto tra i progetti eseguiti con Linux embedded e quelli basati sui maggiori Rtos commerciali (Integrity, VxWorks, LynxOS) è stato condotto dalla società di ricerche Embedded Market Forecasters (Emf), considerando il periodo 2007-2009. In due dei tre anni analizzati, i progetti basati su Linux embedded risultano contenere più linee di codice. Ma, commenta la società, questo non stupisce particolarmente data la natura dei progetti Linux e la disponibilità di codice riutilizzabile e software open source.
Un importante fattore analizzato dalla ricerca per la valutazione del ritorno dell'investimento (Roi) è il tempo impiegato dagli sviluppatori embedded dal momento dell'avvio del progetto alla consegna del prodotto. Su questo punto emerge un fatto abbastanza interessante: se in un precedente studio di Emf (2003) i progetti Linux richiedevano un più lungo periodo di tempo per il completamento in paragone con quelli in cui si usavano Rtos commerciali, questo scenario cambia prendendo in considerazione il periodo 2007-2009. Infatti nel triennio il tempo di progettazione impiegato con gli Rtos commerciali continua a essere in media più lungo (circa 15-16 mesi) rispetto ai tempi di sviluppo ottenuti usando Linux embedded (14 mesi). E, sottolinea Emf, il fatto che i risultati si confermino simili nell'arco dei tre anni porta a considerare la validità di questo trend con un elevato livello di affidabilità.
Sempre al fine di determinare il ritorno dell'investimento, per un'azienda è anche importante avere un indicatore che mostri in che misura i progetti vengono completati in tempo, secondo il piano previsto. Anche qui, nei tre anni, la percentuale di progetti completati in ritardo, quindi dopo la scadenza pianificata, è più alta nel caso degli Rtos commerciali (46% nel 2007-2008; 40,5% nel 2009) che non nei sistemi basati su Linux embedded (38,6% nel 2007-2008; 35,8% nel 2009).
Guardando alla percentuale di progetti che, come risultato finale, hanno rispettato le aspettative iniziali degli sviluppatori nella fase di pre-design in termini di funzionalità, prestazioni e rispetto dei tempi, le due categorie di sistemi operativi (Rtos commerciali e basati su Linux) mostrano invece risultati abbastanza simili.
Un altro fatto da sottolineare è la distinzione nel confronto fra





Figg. 3 - 4 - Le home page dei progetti open source per le piattaforme Genivi e Openmoko

le differenti versioni embedded del Pinguino: in particolare, fra le distribuzioni Linux commerciali (a pagamento) e quelle non commerciali, ossia liberamente scaricabili dal Web. Qui, ricordando che la diffusa e sbagliata convinzione di molti utenti sulla gratuità di Linux ha vissuto per almeno gli ultimi nove anni, lo studio sottolinea come emerga con chiarezza dai dati anno su anno che, per le caratteristiche e funzionalità esaminate, tutti i prodotti Linux commerciali superano le versioni del Pinguino disponibili in 'free download' e basate sull'uso di tool non commerciali.
In ogni caso, in sintesi il rapporto conclude che Linux sta emergendo sul mercato come sistema operativo adottato per molte applicazioni embedded. Inoltre, i miglioramenti apportati da alcune società (Green Hills, Wind River, LynuxWorks) ai propri sistemi operativi 'mission critical' e 'safety critical' hanno reso possibile l'introduzione dei progetti Linux anche in tali applicazioni, caratterizzate da vincoli molto stringenti a livello di affidabilità e sicurezza.

Fig. 5 - Stéphane Deruelle, senior director di Wind River per l'Europa sud-occidentale

'Free' non vuol dire gratis

Oggi tecnologie emergenti come le architetture multicore e le tecniche di virtualizzazione – che consentono su una sola piattaforma hardware la convivenza di più sistemi operativi – stanno fornendo agli sviluppatori embedded ulteriori motivazioni per utilizzare Linux e i sistemi operativi open source, commerciali e non commerciali, nei loro progetti. Ma quali sono in concreto le principali valenze del Pinguino? "Linux embedded – risponde Stéphane Deruelle, senior director di Wind River per l'Europa sud-occidentale – è già vincente in diversi mercati. I due principali valori sono la ricchezza di funzionalità che aiuta i costruttori di dispositivi a fronteggiare la rapida crescita della richiesta di feature per le loro attrezzature e la sua apertura, che guida molti sviluppatori, comunità ed ecosistemi a complementare le reciproche esigenze".
Purtroppo però non sempre gli utenti si rendono conto in tempo di cosa sia realmente necessario, in termini di tempo e risorse, per applicare con profitto i sistemi operativi open source alla progettazione embedded. "La percezione del fatto che il software open source sia 'free' sta portando ai maggiori fallimenti di progetto – chiarisce Deruelle – poiché c'è bisogno di tempo, sforzi e risorse dedicate per trasformare in prodotto, mantenere e supportare questo software nel tempo con una visibilità di lungo termine e processi industriali".
Vi sono poi ulteriori valutazioni da fare quando si considera l'uso di sistemi operativi open source in alcune applicazioni, come quelle di automazione industriale, in cui possono essere richieste caratteristiche di funzionamento real-time. Alla domanda se Linux e tali sistemi siano oggi sufficientemente maturi per questo tipo di applicazioni e per competere con gli Rtos proprietari, Deruelle risponde con chiarezza. La parola 'real-time' si può considerare come un termine relativo, che può significare requisiti diversi per molte differenti necessità di varie tipologie di applicazioni embedded. Tanto per fare un esempio, dice Deruelle, la perdita di un evento real-time in un computer di un aereo o all'interno di un router di rete non produce evidentemente lo stesso impatto in termini di danni. Vi è dunque spazio per entrambi i sistemi operativi e vi sono molte complementarietà che ne guidano l'utilizzo nel settore.

Progetti nel mondo mobile e automotive

Molti sono i progetti open source nel mondo embedded. Vi sono le piattaforme come Android, Moblin e Openmoko, che appartengono al settore dei sistemi operativi dedicati specificamente ai dispositivi di mobile computing, come ad esempio gli smartphone. Progetti simili esistono però anche nel settore automotive, come nel caso della Genivi Alliance, fondata l'anno scorso con l'obiettivo di creare middleware e una piattaforma software open source basati su Linux e utilizzabili da tutta l'industria automobilistica per sviluppare applicazioni di infotainment a bordo delle vetture.

Android	http://developer.android.com
EMF	http://embeddedforecast.com
Genivi	www.genivi.org
Moblin	http://moblin.org
Montavista (Italsoft)	n.53
Openmoko	http://wiki.openmoko.org
Rtlinux	www.rtlinuxfree.com
uClinux	www.uclinux.org
Wind River	n.54