

Pro e contro del software Open Source

Non ha tregua la competizione fra software proprietari e liberi, ed è per questo motivo che Tom Barrett, CEO della società Quadros Systems, ritiene utile dare alcune informazioni sull'utilizzo del software Open Source nella progettazione di sistemi embedded, mettendo in guardia dai rischi tipici di questi prodotti

Tom Barrett
President
Quadros Systems



Tom Barrett, CEO
di Quadros Systems

Il termine Open Source lascia pensare che si tratti di software liberamente utilizzabile: lo si scarica dal Web, lo si compila e lo si utilizza senza costi né preoccupazioni di sorta, nemmeno per l'impiego nei prodotti commerciali.

Purtroppo non è sempre così e dopo un'iniziale illusione di qualità e sicurezza non raramente ci si imbatte in sorprese e imprevisti di ogni tipo.

Quadros Systems sviluppa e commercializza sistemi operativi real time e software per le applicazioni embedded. I tecnici di Quadros si sono imbattuti spesso in sistemi operativi Open Source e hanno potuto così maturare nel tempo un'abbondante esperienza nell'imparare a valutarne vantaggi e svantaggi applicativi.

Tecnologia, dimensioni del codice e costi

La prima doverosa considerazione va fatta per cancellare la fama di cattiva qualità che talvolta viene attribuita ai software Open Source, mentre invece, alcuni di questi possono essere molto utili agli sviluppatori di sistemi embedded. Ciò non toglie che non sono certamente una soluzione per tutti i problemi, poiché ci sono vantaggi ben definiti nell'utilizzo di software proprietari. Inoltre, anche se talvolta la scelta si fa considerando caratteristiche e prestazioni, capita sempre più spesso di dover decidere in base a strategie commerciali e alle previsioni di redditività dei prodotti.

Innanzitutto, non sempre i sistemi operativi Open Source che si scaricano gratuitamente sono utilizzabili per la propria applicazione; alcuni per esempio, sono tagliati su misura per applicazioni specifiche e non si adattano a null'altro, poiché

possono rivelarsi troppo grandi e non facilmente adattabili alle piccole applicazioni.

I sistemi operativi real time commerciali, invece, offrono possibilità di configurazione e scalabilità che consentono di raggiungere dimensioni variabili di 2-30 kByte, laddove questo non si può fare con i Sistemi Operativi Open Source, le cui dimensioni possono essere decisamente più grandi.

Ma quando e perché i software Open Source sono così grandi? Per rispondere a questa domanda bisogna pensare che in questo settore lavorano ingegneri software di grande talento impegnati a realizzare prodotti con prestazioni e caratteristiche che nulla hanno da invidiare ai software proprietari. Tendono, però, a riempirli di funzionalità e opzioni, generando come risultato un'elevata occupazione di memoria sul proprio target. Nonostante i bassi costi, le memorie sono un costo ricorrente in un sistema embedded, ed ha senso, sia dal punto di vista dello sviluppatore sia per l'economicità del prodotto, ridurre al minimo l'utilizzo di memoria.

Quando si parla di milioni di pezzi venduti, ecco che la maggior quantità di memoria utilizzata per un sistema operativo Open Source può incidere significativamente sui costi.

I Sistemi Operativi Real Time commerciali, per esempio, possono variare nella occupazione di memoria, da pochi kByte a poche centinaia di kByte; ciò detto, la maggioranza di questi

RTOS difficilmente supera il tetto dei 100 kByte e gran parte di essi si aggira all'incirca a metà.

In passato, un test fatto dai tecnici Quadros ha riscontrato che il solo kernel di Linux 2.4 era di dimensioni maggiori della massima configurazione del RTOS della società completo di tutti gli accessori.

Questo farebbe di Linux una scelta sbagliata? Non necessariamente. Il punto è che le dimensioni del codice possono influire notevolmente sul costo dei sistemi e, dunque, molto spesso la scelta fra i due può essere fatta solo stimando bene l'impatto sul costo del prodotto finale.

Affidabilità

Il mercato embedded non è certamente rigido e legato a voci di costo fisse e irremovibili; al contrario è riccamente variopinto con infinite possibilità di caratterizzazione applicativa. Tuttavia, i sistemi aperti più popolari, come Linux, non tengono conto di questa evidente constatazione e spesso non sono facilmente adattabili alle applicazioni a elevata specializzazione. In questi casi i prodotti proprietari sono più adatti perché consentono di definire funzionalità dedicate molto efficaci.

Nel mercato embedded ci sono applicazioni basate su DSP, su microcontrollore oppure su entrambi. Un sistema basato su DSP lavora meglio con alcuni tipi di RTOS che a loro volta non sono ideali per applicazioni di controllo e, viceversa, alcuni RTOS molto efficienti con i microcontrollori non lo sono affatto per elaborare applicazioni con un elevato flusso di dati. Non c'è dubbio che i software Open Source non siano la miglior scelta per soddisfare una così ampia gamma applicativa. Molti sistemi operativi real time sono commercializzati ostentandone la capacità di offrire prestazioni in tempo reale deterministiche e garantite nelle più severe condizioni d'impiego (hard real-time). Sebbene non tutti i sistemi embedded richiedano prestazioni di hard real-time, questi possono beneficiare di un RTOS commerciale per svariati altri motivi quali la lunghezza del codice, la modularità delle funzioni operative, l'efficienza e l'affidabilità delle prestazioni.

La scelta di un sistema operativo Open Source esiste certamente per applicazioni non-real-time, ma tenendo conto che il loro codice sarà di dimensioni maggiori.

D'altro canto, il sistema operativo deve essere la parte più affidabile di un sistema embedded. I software Open Source possono avere 'radici' solide e affidabili, ma sono continuamente modificati da parte della comunità di ingegneri che dopo averli fatti nascere non smette di aggiornarli con l'intento di migliorarne le prestazioni. Ciò significa che vi possono essere parti di codice appena inseriti che non hanno avuto sufficiente sperimentazione sul campo e possono celare rischi di malfunzionamento imprevisti. I RTOS proprietari, invece, nascono solo dopo essere stati abbondantemente sperimentati e vengono commercializzati solo con l'assoluta garanzia di funzionare perfettamente in ogni condizione d'impiego, senza rischi, né sorprese. Questo punto di vista può essere decisivo nel far preferire i RTOS proprietari tutte le volte che le aziende vogliono proteggere i propri prodotti con certezza totale da qualsivoglia eventualità di malfunzionamento.

 **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

Time-to-market

Quando si scarica gratuitamente un software Open Source dal Web c'è una ragionevole probabilità che per poter funzionare ci sia bisogno di qualche modifica sia sull'hardware sia nell'applicazione. Questa fase, tuttavia, non è priva di rischi perché ci si potrebbe imbattere in un problema inconsueto per il quale potrebbe non esserci un'unica soluzione, ma due o tre possibilità, o talvolta nessuna. In tal caso qualcuno deve assumersi la responsabilità di risolvere il problema, e non è detto che Internet possa aiutare perché non è facile trovare qualcuno che abbia avuto lo stesso identico problema o che ne conosca la natura e sappia come risolverlo. Inoltre, tutto ciò allunga il tempo di sviluppo del prodotto e, quindi, i suoi costi.

È possibile aggirare il guaio con l'acquisto di un supporto completo da società che forniscono il software Open Source, così da avere qualcuno a cui rivolgersi; situazione, questa, assai comune. Non è però insolito che tale costo superi quello dell'acquisto di un software proprietario.

Quest'ultimo, invece, viene sempre fornito col supporto tecnico del produttore ed è accompagnato da un numero di telefono e da un indirizzo email dove rivolgersi per avere consigli da qualcuno certamente competente in materia.

Oggi è molto comune per i costruttori realizzare molteplici prodotti per anno, in più versioni simili ma che si differenziano per l'adattamento alle esigenze specifiche di altrettante categorie di utilizzatori finali.

Diminuisce quindi la vita media dei prodotti e la loro redditività commerciale, vedendo però aumentare tempi e complessità dello sviluppo.

Al momento del lancio di nuovi prodotti, questi fattori influiscono nel raggiungimento di quella finestra di mercato alla quale il progettista di sistemi embedded deve fare riferimento.

Gli sviluppatori hanno un maggior ritorno quando spendono il tempo di sviluppo inserendo nei propri prodotti quanto più valore aggiunto, sulla base di conoscenze specifiche. In molti casi, questo esclude i sistemi operativi e l'integrazione di device drivers, protocolli di comunicazione e middleware. Lo sviluppatore deve semplicemente conoscere come usarli al meglio. Invero, la qualità dei prodotti è l'unica variabile capace di creare ritorno di redditività sugli investimenti nel ciclo di sviluppo.

I software Open Source, tuttavia, soprattutto se devono integrarsi nei sistemi embedded grandi e complessi, impongono un adattamento non sempre agevole che può allungare i tempi di sviluppo e quindi anche i costi, oltre che comportare rischi di affidabilità, due rischi che ogni sviluppatore tende assiduamente ad evitare.

Molte società produttrici di Sistemi Operativi Real Time commerciali, inclusa Quadros Systems, sono in grado di fornire il

valore fondamentale di un RTOS, protocollo di comunicazione e middleware, sono tutti integrati, portati e testati per uno specifico target e per uno specifico compilatore. Questo significa rapidità di sviluppo e minimo rischio. La differenza è abbastanza netta.

Licenze

Tipicamente i prodotti Open Source usano licenze note come GPL e LGPL e altre forme di licenza open source.

Queste licenze possono imporre severe restrizioni su ciò che può essere fatto col codice Open Source e come le applicazioni possono interagire con esso. Alcune di queste restrizioni possono legalmente obbligare a mostrare la proprietà intellettuale, ad esempio l'applicativo, a chiunque lo richieda, sia pure un concorrente.

Si può certamente ricorrere a uno studio legale specializzato in proprietà intellettuale software per proteggere i propri interessi, ma anche questo porta ad alzare prepotentemente i costi di sviluppo.

Utilizzando un software commerciale si paga sicuramente un diritto di licenza che generalmente viene offerto nelle due forme "shrinkwrap" o "negotiated". Nel primo caso la licenza serve a limitare il numero dei prodotti o le modalità d'utilizzo in vario modo. Generalmente non occorre essere esperti per interpretarne il contenuto e le clausole sono facilmente comprensibili e preparate per adattarsi alle esigenze degli utenti. Nel caso delle licenze negoziabili i tempi si allungano un po' perché implicano il raggiungimento di un accordo fra le parti in funzione delle esigenze commerciali di entrambi. Ciò significa che crescono tempi e costi, ma migliora la redditività dei prodotti.

È chiaro, dunque, che ci sono molte occasioni tali per cui i software Open Source da "liberi e gratuiti" diventano costosi e, talvolta, persino più costosi rispetto alla licenza di un RTOS proprietario. Ciò significa che la reputazione di economicità che avvolge i software Open Source non è un parametro su cui si può fare affidamento per una corretta scelta, soprattutto nello sviluppo dei sistemi embedded di grandi dimensioni. Al crescere della complessità delle applicazioni sono, infatti, le prestazioni e l'affidabilità i parametri di merito che possono far preferire i sistemi operativi proprietari.

Quadros Systems (Fenway Embedded Systems)

readerservice.it n. 21