

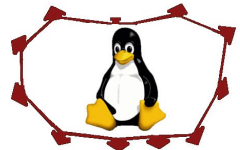
L'automazione Industriale e GNU/Linux

Acquisire knowhow e non licenze

di

Rodolfo Giometti

<giometti@enneenne.com>

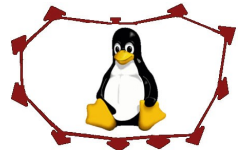


Il sistema GNU/Linux

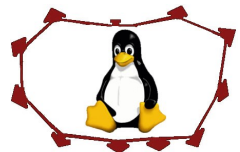
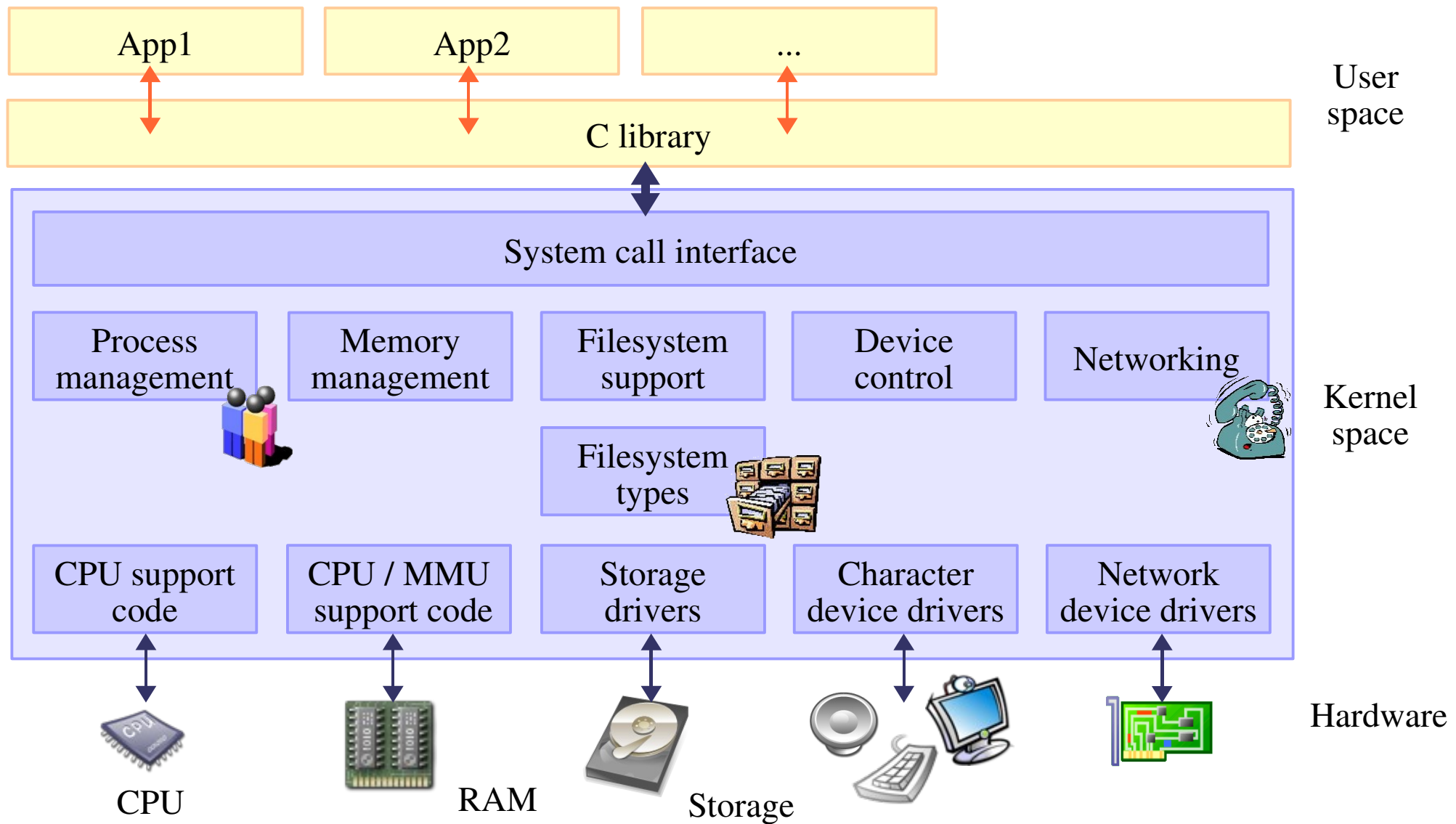
Parliamo di *GNU/Linux* e non solo di *Linux* perché *Linux* di per se è solo il nucleo del sistema.



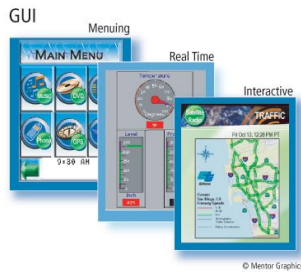
Quando parliamo di «sistema operativo» intendiamo quindi l'insieme del nucleo (*Linux*) più tutta una serie di applicazioni di gestione, utilità, ecc. (*GNU*) del sistema stesso.



Il nucleo (o kernel): Linux



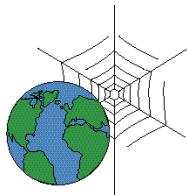
Distro & applicativi: GNU



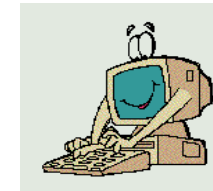
GUI



Tool di comunicazione



Servizi Web



Tool di calcolo



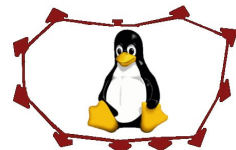
Servizi di
posta



Tool di sviluppo



*Database o
monitoraggio*



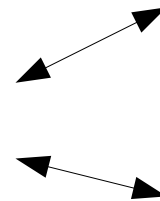
Scambiarsi dati con o senza lo GNU

Molti protocolli di comunicazione utilizzati in ambito industriale sono già implementati come software libero, sia nel kernel sia in userland, altri (purtroppo) no!

WWW, TFTP, ecc.



serial, storage, ether, ecc.



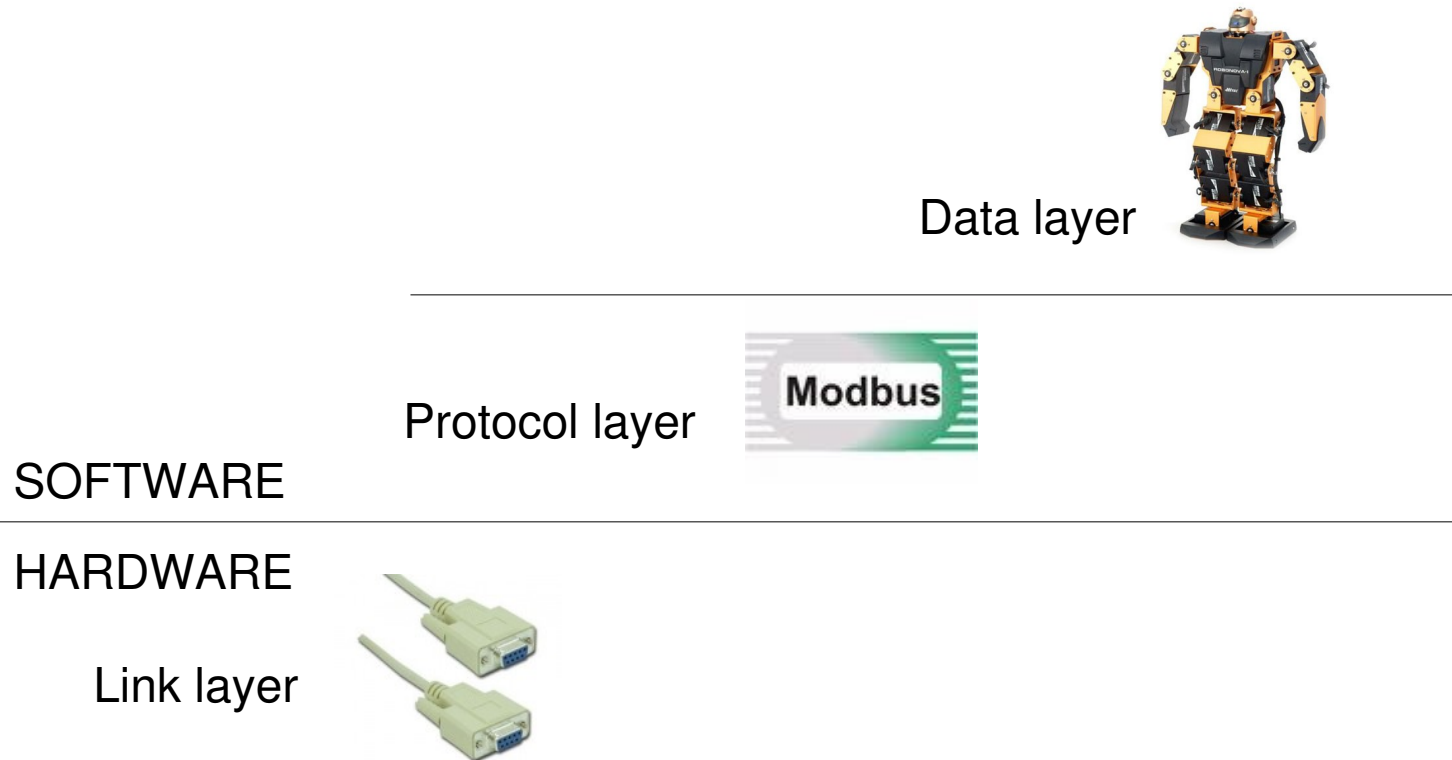
ModBus...

La causa: mancanza di specifiche **libere** o comunque **accessibili**.

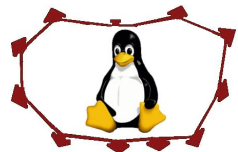
L'effetto: o si compra software **non libero** o lo si **riscrive da zero**.

I dati «sensibili»

In moltissime applicazioni poi non è importante il protocollo di scambio dati, ma il dato scambiato in se!



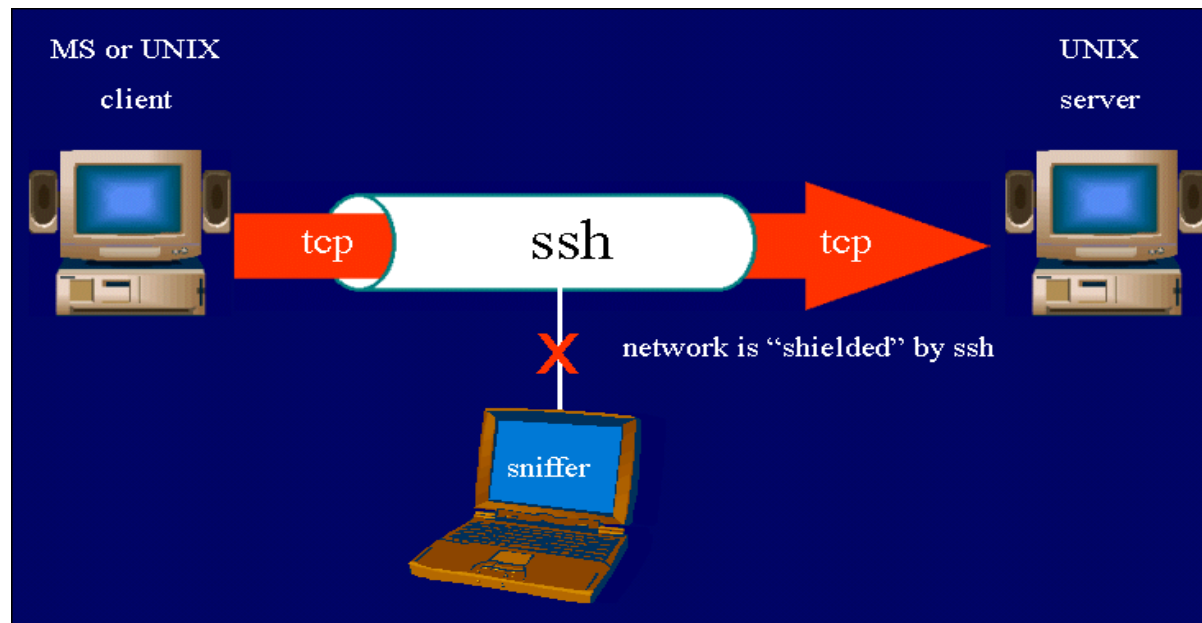
Molti produttori di hardware implementano protocolli (e hardware) non liberi nelle loro macchine per «imprigionare» i clienti e rendere difficile l'adozione di macchine della concorrenza.



Segretezza: «sss» o «ssh»?

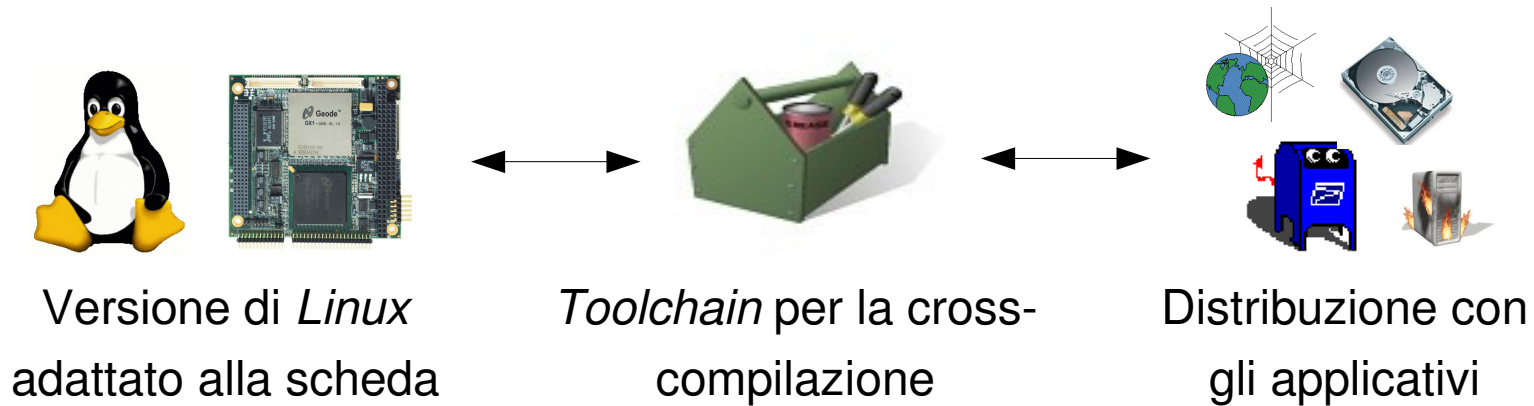
Alcuni pensano che se non si conosce come funziona una serratura, allora questa sia sicura. Niente di più sbagliato! La serratura è sicura se pur conoscendo come funziona, si dimostra che non è violabile.

Ad esempio il protocollo SSH permette di colloquiare via rete con una macchina remota in maniera sicura (dimostrato) e tutta la catena di trasmissione è ben conosciuta!



Sistema di sviluppo embedded

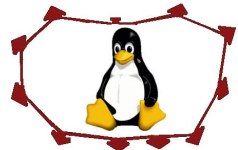
Quando scarichiamo o acquistiamo una distribuzione *embedded GNU/Linux* solitamente ci vengono fornite diverse componenti.



Se questo non avviene allora occorre *arrangiarsi*:

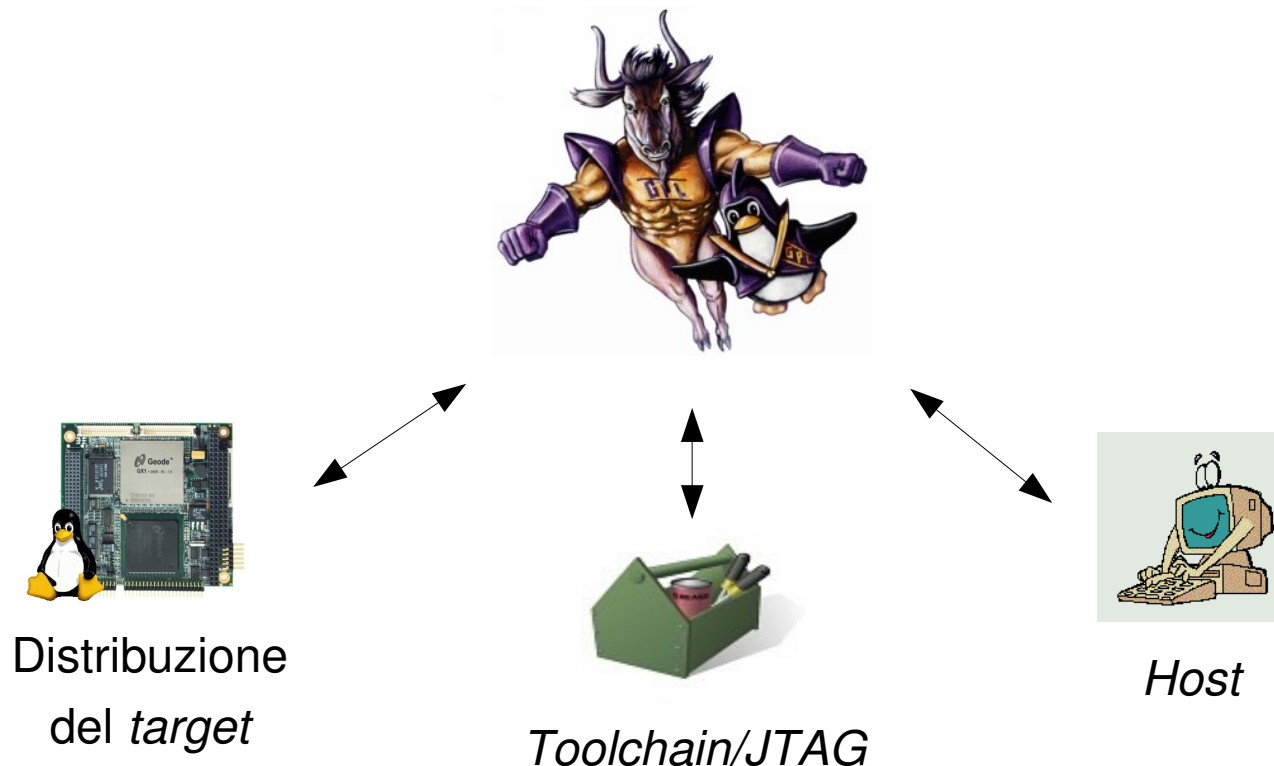


- Prendere i sorgenti
- Applicare le *patch*
- Trovare una *toolchain* & distribuzione
- Compilare, compilare, compilare...



Distribuzioni GNU/Linux

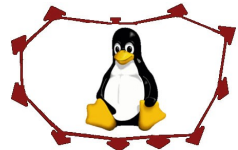
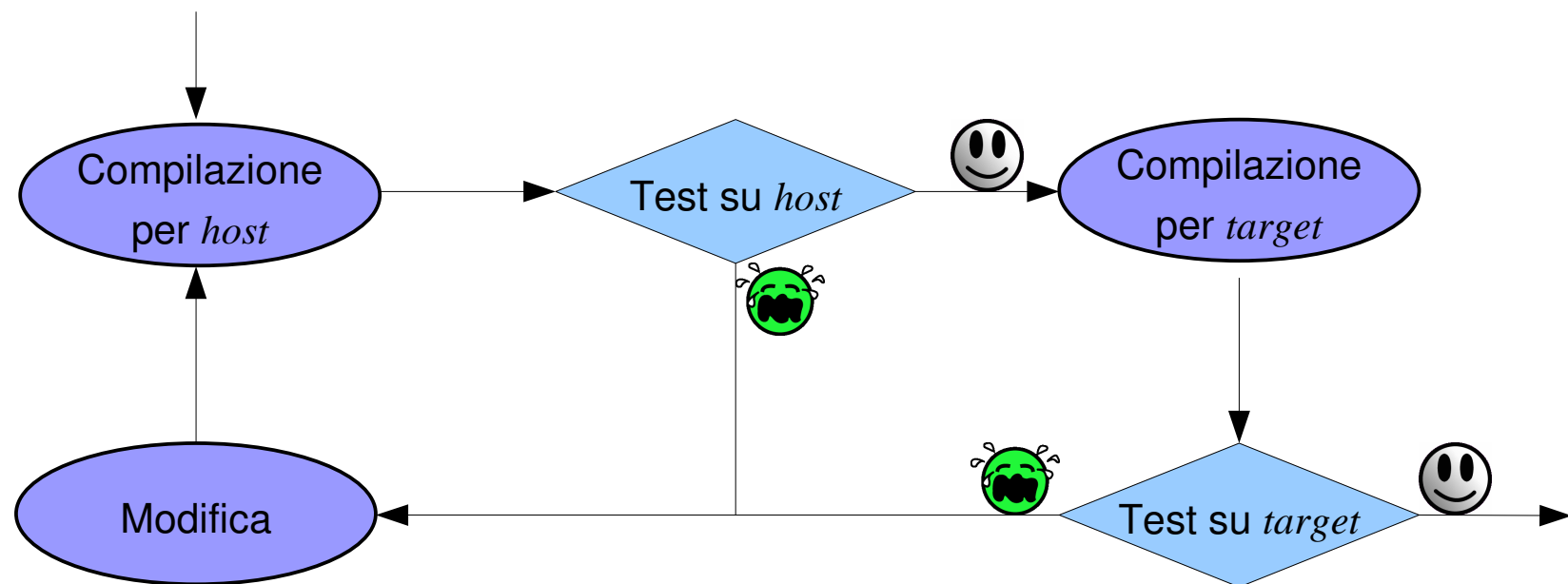
Nel caso dei sistemi *embedded* generalmente possiamo parlare di «distribuzione *embedded GNU/Linux*» poiché abbiamo non solo il sistema *target* di quel tipo ma anche i *tool di sviluppo* appartengono al progetto *GNU* (e molte volte anche il sistema *host*).



Lo sviluppo passo passo

Ogni applicativo è una parte a se stante e quindi è facilmente rimpiazzabile sia in fase di sviluppo che in una successiva fase di messa in opera.

addirittura la fase di test e validazione potrebbe essere effettuata sul sistema *host* e poi, semplicemente ricompilando il tutto trasferita sul sistema *target*!



Dove trovare aiuto!!!

Use the source, Luke!

Thanks to LucasArts



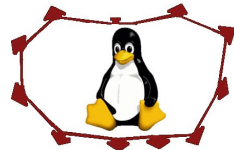
Leggere il codice è importantissimo, ma le fonti di informazione possono essere molte a partire dalla rete.



Ci sono poi i libri specifici per la programmazione in user o kernel space.

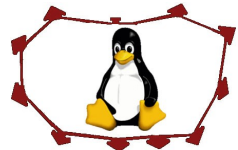


E in fine ci sono anche i consulenti...



Conclusioni

- Il software libero lo si può riutilizzare ad libitum (royalties free).
- Il software libero può essere studiato.
- È importante che le specifiche del link layer e del protocol layer di una apparecchiatura siano libere (o almeno accessibili) onde evitare il monopolio della fornitura.
- Utilizzare software libero non impedisce di avere alcune parti proprietarie.
- Utilizzare software libero permette di utilizzare gli strumenti di base come si vuole.
- Alcune parti dello sviluppo possono essere date esternamente senza vincoli di licenza.



Riferimenti

http://ftp.enneenne.com/pub/docs/talks/acquisire_knowhow

<http://www.kernel.org>

<http://www.gnu.org>

<http://www.linux.it>

