

L'uso di Linux nei sistemi embedded

Perché usarlo e con quali tecniche di sviluppo



di

Rodolfo Giometti

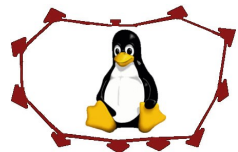
<giometti@enneenne.com>

Creative Commons - Attribuzione-Non commerciale-Condividi allo stesso modo 2.5 Italia - <http://creativecommons.org/licenses/by-nc-sa/2.5/it/>



Better Software 2010

Professionisti
Consulenti
ICT - Italia



Rodolfo Giometti <giometti@linux.it>

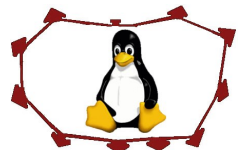


Rodolfo
Ing. Giometti

- Sono un ingegnere informatico libero professionista.
- Scrivo driver e faccio port di Linux su piattaforme x86, ARM, PPC e MIPS (Maintainer LinuxPPS + vari driver tra cui i2c-mux, c2port, pxa-keyb).
- Scrivo driver e faccio port di U-Boot su piattaforme ARM, PPC e MIPS (ex Custodian Video + vari driver).
- Progetto/realizzo apparati di controllo automatico (Real-Time) e monitoraggio remoto GNU/Linux based.
- Esperto di protocolli di comunicazione (TFTP, SNMP, LLRP + altri IP/ETH) e bus dati (USB, I2C, CAN, ecc.).
- Faccio corsi specialistici su U-Boot/Linux/UNIX.
- Scrivo su Linux&C e il Portale `consulenti-ict.it`.
- *Software Libre Evangelist*.

HOMEPAGE: <http://www.enneenne.com/staff/giometti>

Better Software 2010

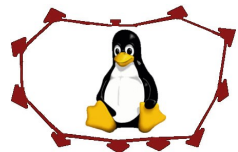


Il sistema GNU/Linux

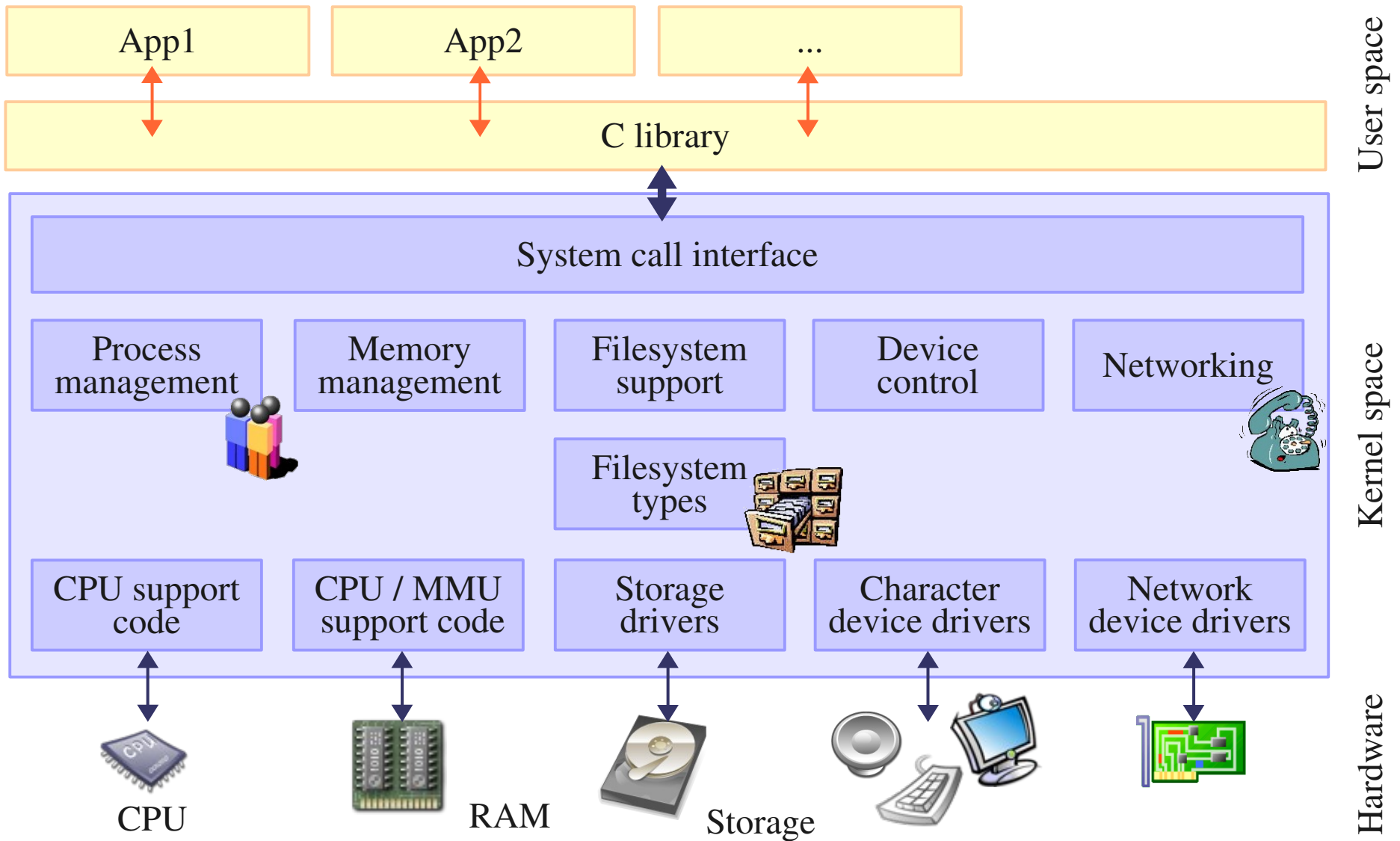
Parliamo di *GNU/Linux* e non solo di *Linux* perché *Linux* di per se è solo il nucleo del sistema.



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)

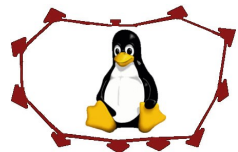


Creative Commons - Attribuzione-Non commerciale-Condividi allo stesso modo 2.5 Italia - <http://creativecommons.org/licenses/by-nc-sa/2.5/it/>



Better Software 2010

Professionisti
Consulenti
ICT - Italia



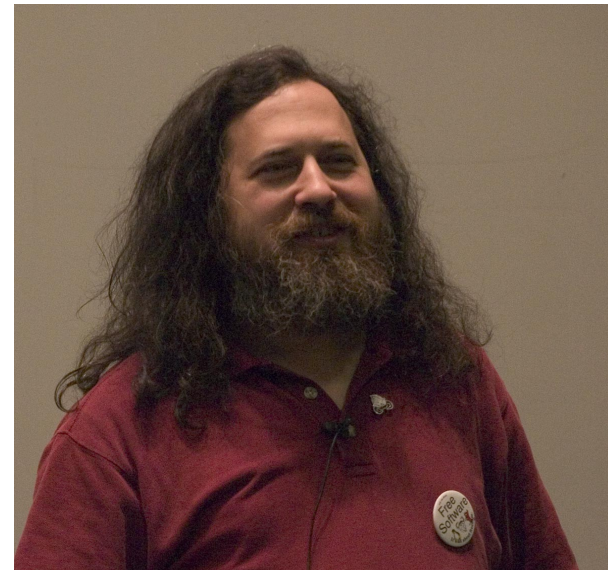
UNIX → GNU

Il progetto GNU voleva realizzare un sistema basato su UNIX ma che fosse composto solo da Software Libero.

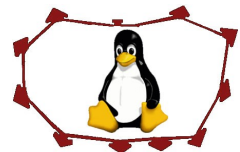


= Gnu is Not Unix

Il Software libero fu «riscoperto» da R. M. Stallman nel 1984, anno in cui il progetto GNU prese vita.



Richard Matthew Stallman

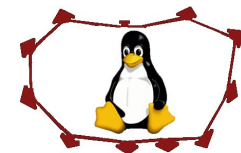


La licenza GNU/GPL

Stallman è un hacker (da non confondere con *cracker*), quindi aveva la competenza per scrivere un sistema da zero. Ma si rese conto ben presto che questo non bastava: gli serviva una *licenza*!

La licenza GNU GPL si basa su **4 libertà**:

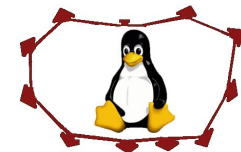
- ★ **0** (o libertà fondamentale): **la libertà di eseguire il programma per qualsiasi scopo.**
- ★ **1**: **la libertà di studiare il funzionamento del programma, e adattarlo alle proprie esigenze.**
- ★ **2**: **la libertà di copia e redistribuzione.**
- ★ **3**: **la libertà di migliorare il programma, e distribuirne i miglioramenti.**



Illibertà 0

Non si può usare il programma per qualsiasi scopo.

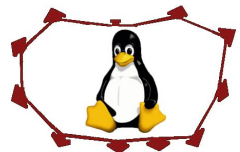
- Se il PRODOTTO SOFTWARE è Windows 2000, non potrà essere utilizzato da più di due (2) processori sul COMPUTER [...] (Windows)
- Se il PRODOTTO SOFTWARE è Windows Millennium Edition o Windows 98, è possibile connettere un massimo di cinque (5) COMPUTER al singolo COMPUTER che esegue il PRODOTTO SOFTWARE esclusivamente per accedere a Internet [...] (Windows)



Illibertà 1

Non si può studiare il funzionamento del programma.

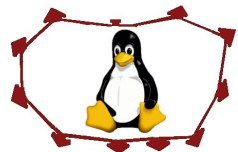
- L'utente non può decodificare, decompilare disassemblare il PRODOTTO SOFTWARE [...] (Windows ME/2000/XP/ecc.)



Illibertà 2

Non si possono redistribuire copie del programma.

- [...] l'utente è autorizzato a utilizzare una (1) copia del PRODOTTO SOFTWARE [...] (Windows ME)
- L'utente non può concedere in locazione, in leasing o in prestito il PRODOTTO SOFTWARE a un altro utente. [...] (Windows - Office)

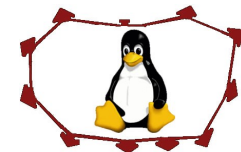


Illibertà 3

Non si può studiare il funzionamento del programma.

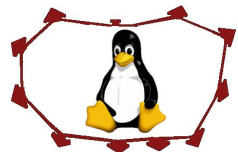
- Non si può migliorare (o correggere) il programma né distribuirne i miglioramenti.

Non esplicitata ma deriva direttamente dalle illibertà 1 e 2 che impediscono, rispettivamente, lo studio del funzionamento del programma e la possibilità di redistribuirne delle copie.



Perquisizioni...

[...] Il SOFTWARE contiene componenti che abilitano e facilitano l'utilizzo di alcuni servizi basati su Internet. **L'utente prende atto e acconsente al fatto che Microsoft, Microsoft Corporation o le rispettive filiali possano verificare automaticamente la versione del SOFTWARE e/o dei relativi componenti in uso da parte dell'utente e possano provvedere ad aggiornamenti o integrazioni al SOFTWARE che possono essere scaricate automaticamente nel COMPUTER dell'utente.** (Windows XP)



Pirateria

Si noti come queste illibertà ingigantiscano il concetto di pirateria!

Se si utilizzassero infatti Software Libero molti dei comportamenti che qualificano un utente come «pirata» sarebbero permessi dalla licenza d'uso, e quindi legali.

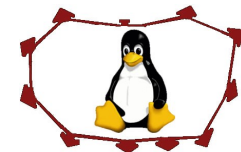
La comunità del Software Libero
Da tempo dice STOP alla pirateria:
usiamo tutti Software Libero!!!

software libero



Better Software 2010

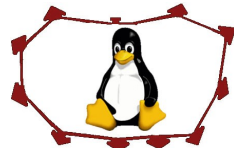
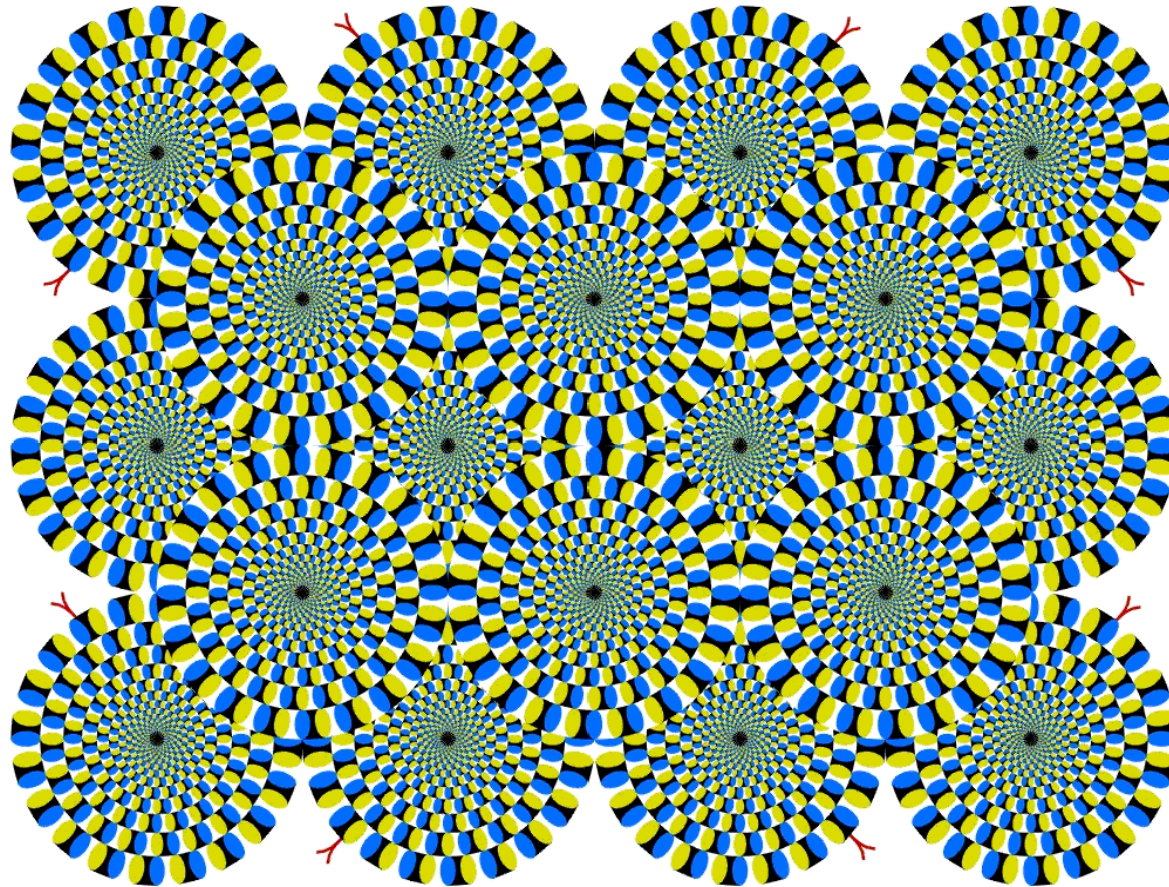
Professionisti
Consulenti
ICT - Italia



FUD sul Software Libero

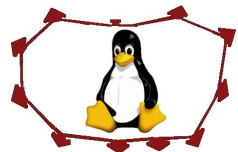
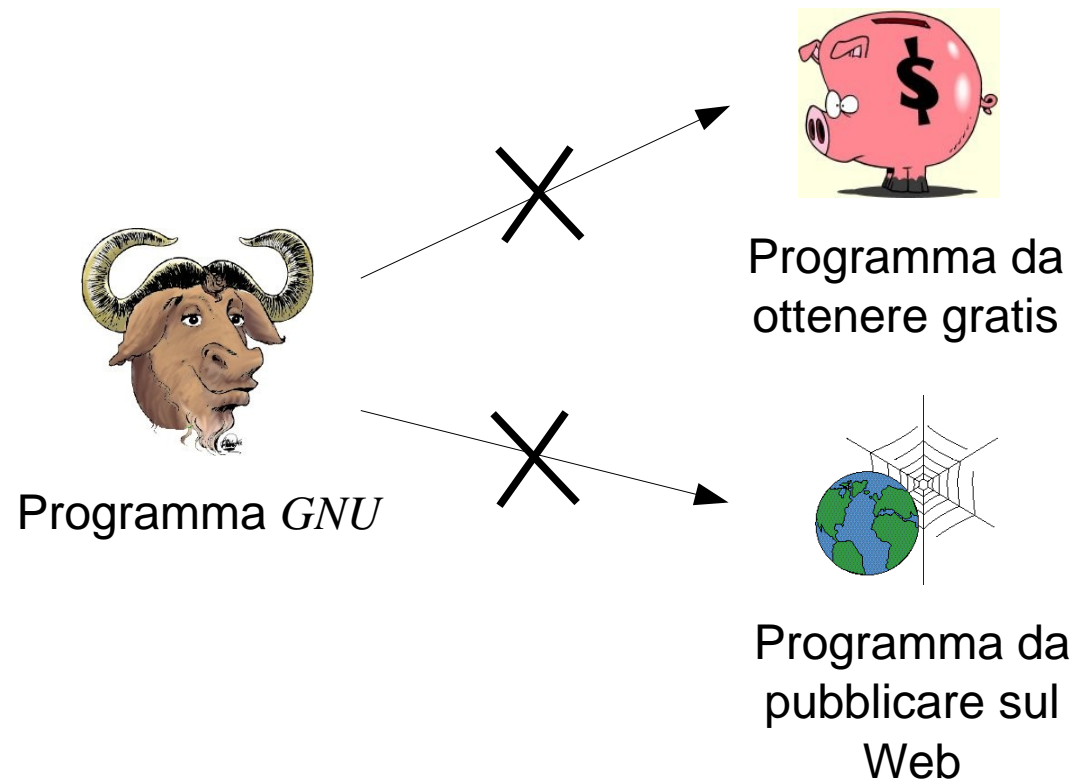
FUD is the fear, uncertainty, and doubt that sales people instill in the minds of potential customers who might be considering products.

[Jargon File]



Miti da sfatare

Molti erroneamente pensano che un programma coperto dalla licenza *GNU/GPL* sia automaticamente gratis e/o da pubblicare sul *Web* coercitivamente.



Una regola da seguire

L'unico che ha dei diritti verso di voi (produttori del software) è solo il cliente a cui viene fornito il prodotto software (anche solo in forma compilata), la licenza *GNU/GPL* dice infatti che solo in questo caso siete obbligati a fornire gratuitamente (o al costo di copia) i sorgenti coperti da tale licenza.



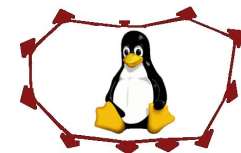
Programma *GNU*
compilato

← Legame indissolubile →

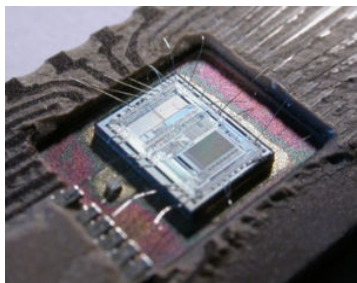


Programma *GNU*
sorgente

La *GNU/GPL* richiede questo per garantire che anche i vostri clienti godano gli stessi diritti di cui voi avete goduto.



Cos'è un sistema embedded?



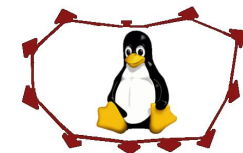
Piccolo → Risorse ridotte

Piccolo → Risorse non «troppo» ridotte

Piccolo → Risorse non ridotte



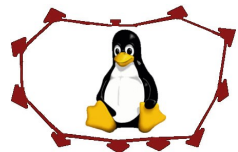
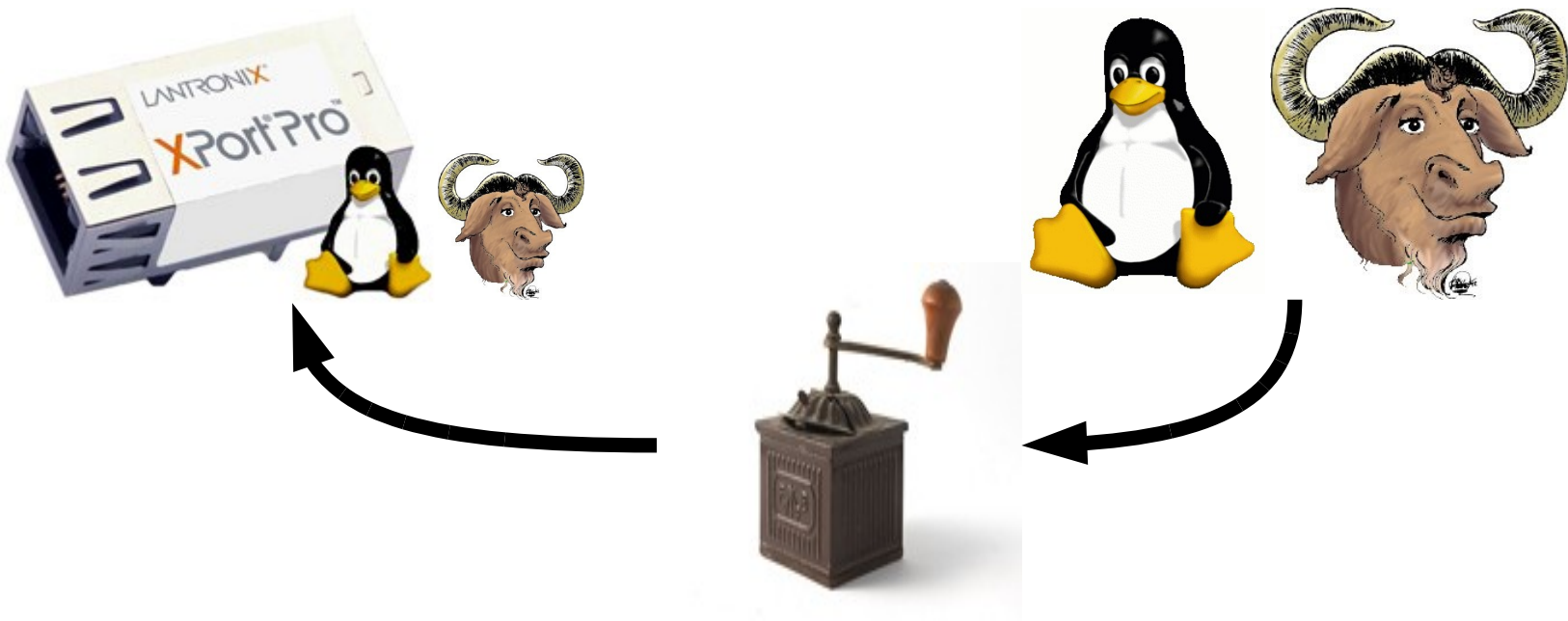
Embedded vuol dire «incassato», quindi non necessariamente piccolo!



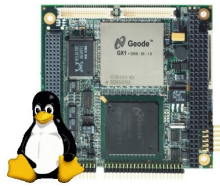
GNU/Linux in un embedded

GNU/Linux è un sistema UNIX, quindi ha bisogno di un minimo di risorse per funzionare, ma ciò non toglie che non possa stare in sistemi molto piccoli!

Quello che si fa di solito, quando le risorse sono scarse, è ritagliare/comprimere il sistema in modo che funzioni ugualmente.



Un po' di nomi...



Con il termine *target* si intende il sistema *embedded* sul quale gira la distribuzione *embedded*, ed è quindi il sistema finale da sviluppare.



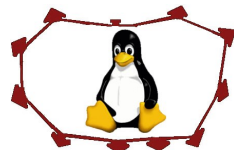
Con il termine *host* si intende il sistema di sviluppo, cioè è la macchina che utilizziamo per *cross-compilare* e che ci aiuta nello sviluppo.



La *toolchain* è invece l'insieme delle *binutils*, delle librerie (libc, ecc.) e del *cross-compilatore*.



Il *JTAG* è un dispositivo con il quale è possibile realizzare sessioni di debug molto spinte (si può mandare la CPU passo passo!)



Sistema di sviluppo embedded

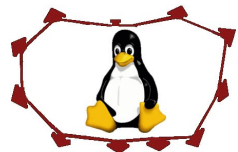
Quando scarichiamo o acquistiamo una distribuzione *embedded GNU/Linux*, solitamente ci vengono fornite diverse componenti.



Se questo non avviene allora occorre *arrangiarsi*:

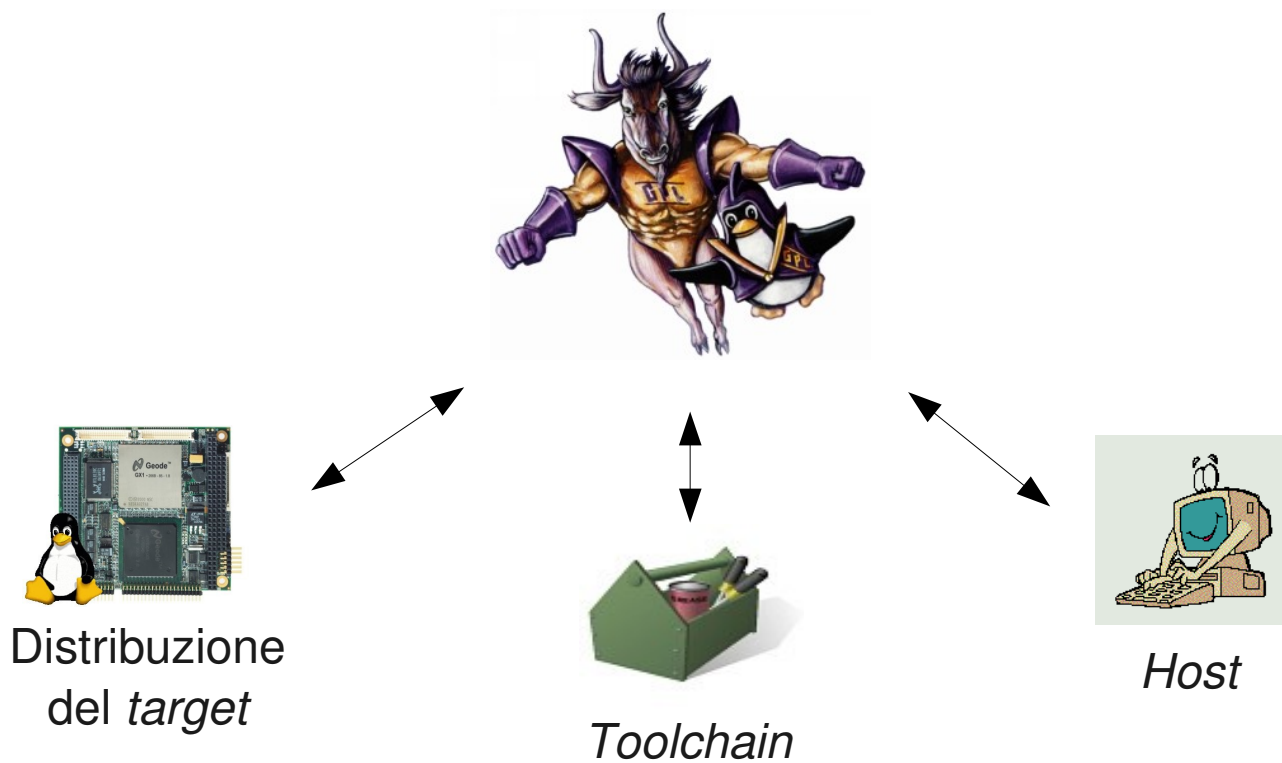


- Prendere i sorgenti
- Applicare le *patch*
- Trovare una *toolchain* & distribuzione
- Compilare, compilare, compilare...

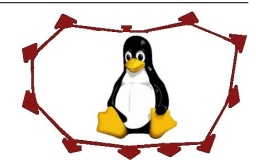


Distribuzioni GNU/Linux

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*).

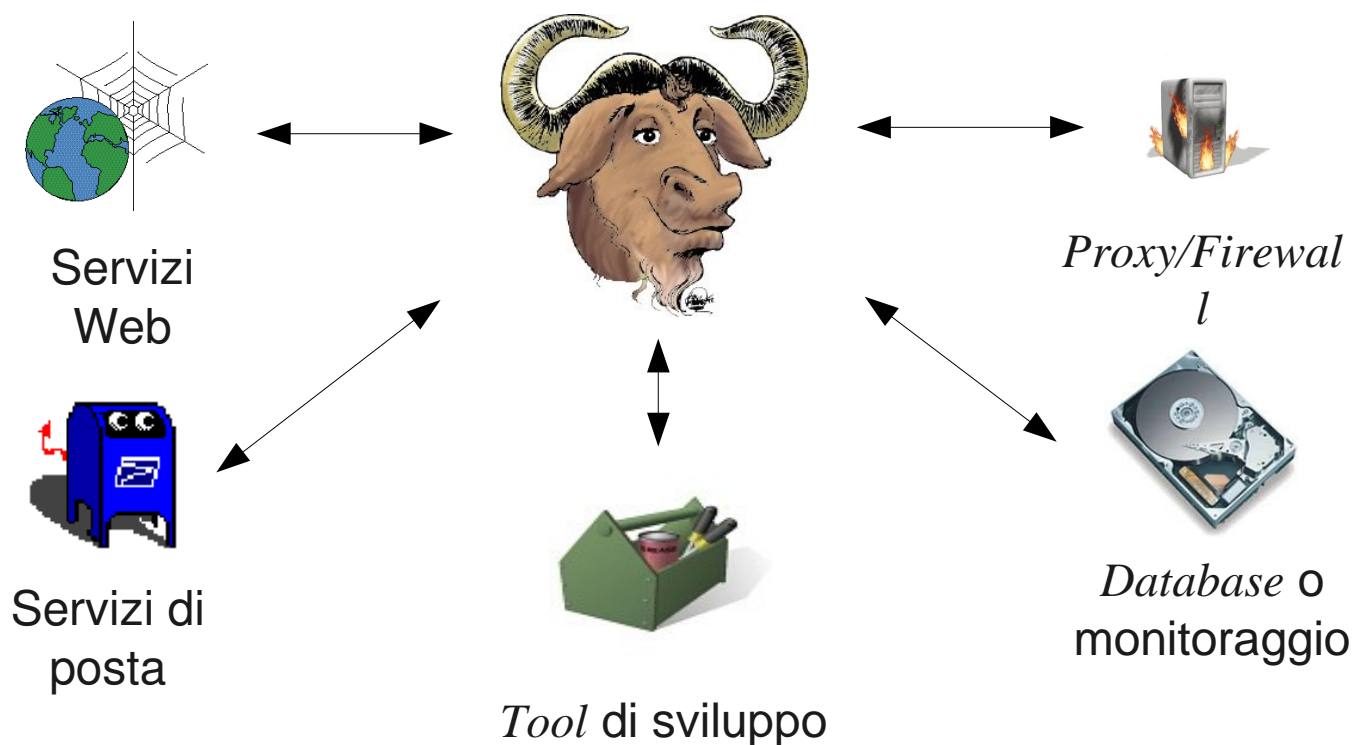


Creative Commons - Attribuzione-Non commerciale-Condividi allo stesso modo 2.5 Italia - <http://creativecommons.org/licenses/by-nc-sa/2.5/it/>



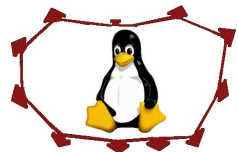
Distro & applicativi

Quando parliamo di sistemi *GNU/Linux* parliamo sostanzialmente di sistemi che posseggono già di per se una grande quantità di programmi applicativi (molte volte però non vengono forniti!!!).



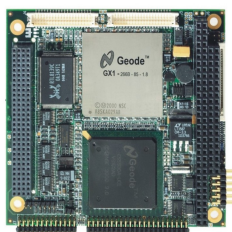
Cosa ci viene dato

- La versione di *Linux* che andiamo ad utilizzare deve (o dovrebbe) supportare i *driver* per tutte le periferiche del sistema. Nel caso che questo non avvenga occorre farsela fare (dal venditore o altri) o farsela in modo che si possa controllare completamente la macchina.
- Una buona *toolchain* dovrebbe includere anche altre librerie oltre alla `libc`, come le librerie grafiche (`libX`, `libgtk`, ecc.).
- Anche il bootloader è un componente importante che dovrebbe essere libero.
- Un buon supporto per lo sviluppo è dato anche dalla disponibilità o meno di tool come: `gdb`, `strace`, `netcat`, `tcpdump`, ecc..
- In fine, per la programmazione a basso livello è importante avere un JTAG.



Debian everywhere!!!

Se la nostra macchina *embedded* ha poi risorse sufficienti (una scheda di rete o porta USB-device e almeno 32MB di RAM (ma potrebbe andare anche con meno..)) potremmo anche pensare di installare una distribuzione completa magari attraverso *NFS*!
La distribuzione Debian, per esempio, essendo disponibile per molte architetture è un ottimo candidato.



Via NFS



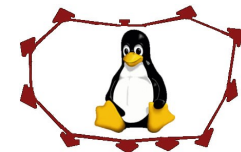
Via /etc/exports



Ethernet



USBnet

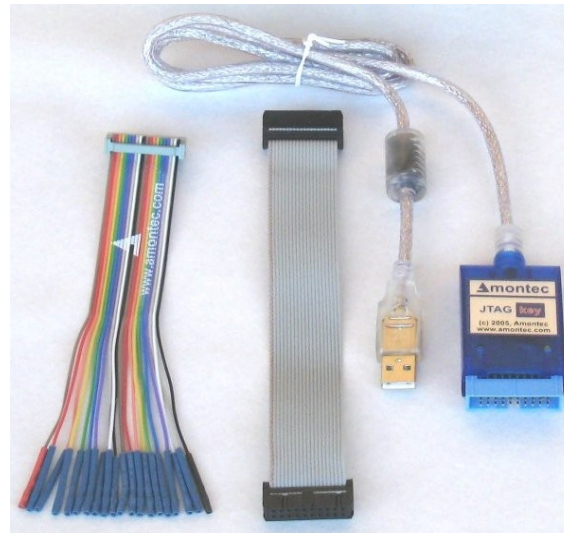


II JTAG

Molte volte non ci si pensa, ma è fondamentale avere a disposizione un JTAG con il relativo software di gestione (funzionante) se dobbiamo realizzare applicazioni di basso livello come:

- suspend/resume;
- boot non da NOR flash;
- test delle fasi iniziali di start up del kernel;
- ecc.

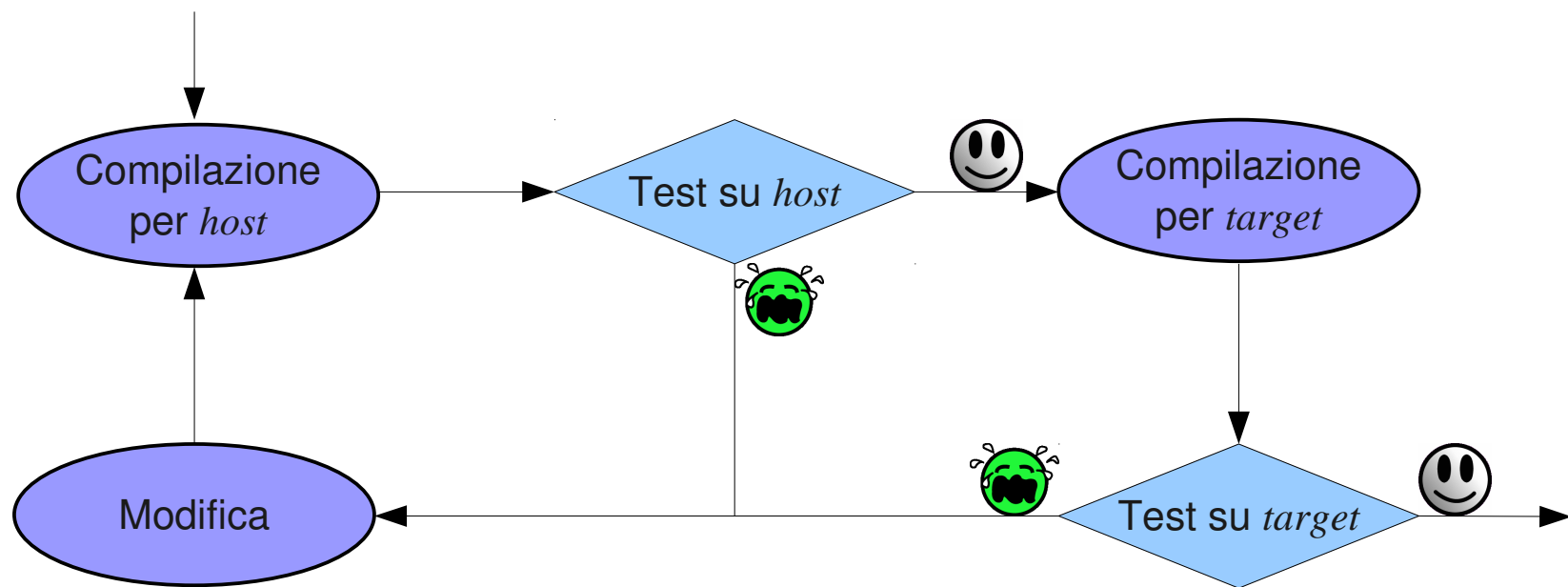
OpenOCD +



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 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!

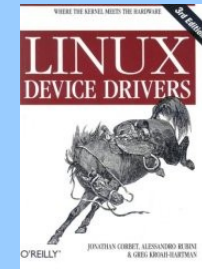
Thanks to LucasArts



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/kernel space.



E in fine ci sono anche i consulenti...

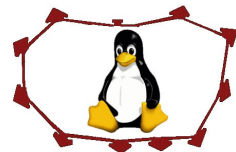


Creative Commons - Attribuzione-Non commerciale-Condividi allo stesso modo 2.5 Italia - <http://creativecommons.org/licenses/by-nc-sa/2.5/it/>



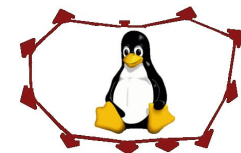
Better Software 2010

Professionisti
Consulenti
ICT - Italia



I vantaggi di usare GNU/Linux

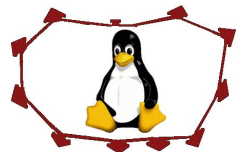
- Ogni *driver* di sistema può essere definito come un modulo di nucleo.
- Nei sistemi con *MMU* è possibile utilizzare tutta una serie di meccanismi di protezione ed astrazione della memoria (memoria virtuale) nonché tecniche di accesso diretto ai dispositivi (`mmap()`).
- I meccanismi di *IPC* sono molteplici (*pipe*, *fifo*, *socket*, ecc.) e ben standardizzati.
- Essendo un sistema *UNIX-like* possiede l'astrazione file per cui ogni cosa nel sistema è visto come un file il che permette di risolvere molti problemi in maniera veloce ed elegante.
- Sono disponibili «tonnellate» di programmi applicativi e molti driver dei più comuni dispositivi.



I vantaggi di usare Software Libero

Utilizzare software libero in generale, oppure in un sistema *embedded* in particolare, ha enormi vantaggi:

- Tutto il codice sorgente è disponibile e può essere analizzato per sapere come il sistema reagirà ad una data sollecitazione.
- Oltre ad essere disponibile, il codice è anche **modificabile** e **ridistribuibile**! Questo permette quindi di adattarlo alle proprie esigenze e di correggerne gli eventuali malfunzionamenti. Completa indipendenza dal fornitore del servizio.
- È possibile allora **avere più consulenti per diverse parti del sistema**. Inoltre i **costi** di queste consulenze **possono essere ripartiti sul numero di prodotti** realizzati poiché non siamo in presenza di royalties.
- Abbiamo molte **risorse di aiuto direttamente in Internet** (maillist, siti dedicati, autore del software, ecc.) che ci possono fornire spunti per risolvere i problemi implementativi.



(alcuni) Riferimenti

<http://www.enneenne.com>
<http://www.gnu.org>
<http://www.kernel.org>
<http://www.denx.de/wiki/U-Boot>
<http://wiki.openembedded.net>
<http://www.debian.org/>
<http://openocd.berlios.de/web/>



Creative Commons - Attribuzione-Non commerciale-Condividi allo stesso modo 2.5 Italia - <http://creativecommons.org/licenses/by-nc-sa/2.5/it/>



Better Software 2010

