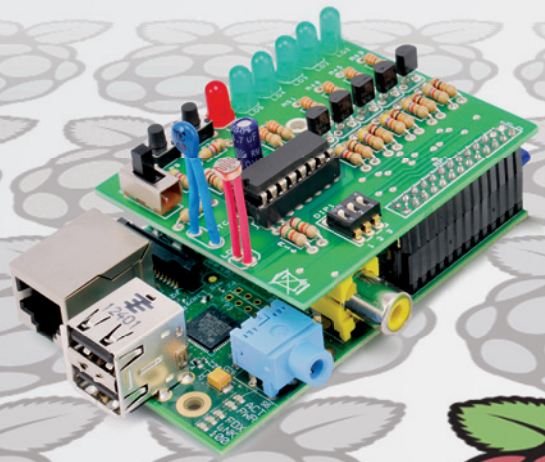
MARCO MAGAGNIN

RASPBERRY PI
il mio primo
Linux Embedded

Scopri le potenzialità di Linux Embedded su Raspberry Pi con esperimenti pratici e di programmazione con i linguaggi più diffusi.



Tanti esperimenti pratici con la scheda dedicata

9 771124 854015 30176
Supplemento a Elettronica In n. 176

Acquisisci le conoscenze necessarie ad avventurarti nel mondo dello sviluppo e della programmazione dei sistemi Linux Embedded, con particolare attenzione all'utilizzo di Raspberry Pi interfacciato a dispositivi, sensori e circuiti elettronici capaci di trasformare la scheda in un ponte fra il web e il mondo reale. Questo libro non richiede conoscenze pregresse e ti avvia all'uso delle moderne piattaforme Linux Embedded. Passo dopo passo potrai non solo capire come il sistema operativo Linux ti permette di operare, ma anche come integrare il mondo dell'hardware attraverso la scheda di sperimentazione analogico/digitale, appositamente progettata per Raspberry Pi e che ne potenzia le possibilità di collegamento, misurazione e controllo. Entrare oggi nel mondo di Linux Embedded con questo approccio è una scelta importante e che ti proietta in un settore della tecnologia molto attivo e in forte espansione: l'Internet delle cose. Scopri oggi se sei portato per le tecnologie di domani e assicurati così una migliore comprensione dei fenomeni tecnologici che condizioneranno la vita di questo decennio.

Raspberry Pi, il mio primo Linux Embedded

di Marco Magagnin

Supplemento al numero 176 di Elettronica In, Maggio 2013.
Rivista mensile registrata presso il Tribunale di Milano con il n. 245 il 3-05-1995.
Direttore Responsabile: Arsenio Spadoni. Prezzo di copertina Euro 13,90.
Redazione e Amministrazione: Vispa Edizioni, Via Adige 11, 21013 Gallarate (VA).
Telefono 0331-752668 Fax 0331-778112
Iscrizione al Registro Operatori della Comunicazione n. 3754 del 27/11/2001.
Distribuzione per l'Italia: SO.DI.P. Angelo Patuzzi S.p.A. Via Bettola 18 - 20092 Cinisello Balsamo (MI).
Stampa: ROTO3 Spa - Via Turbigo, 11/b - 20022 Castano Primo (MI)

Sommario

1

Introduzione

A volte ci si trova di fronte a qualcosa che ci attira e ci affascina ma non sappiamo da dove partire o temiamo per la troppa complessità da affrontare. Una "guida" passo passo può far superare il "blocco" e aprire nuovi orizzonti. Anche gli astronauti hanno cominciato a muoversi imparando a gattonare.

Per chi è questo libro

Questo libro è rivolto a cultori dell'elettronica, analogica, digitale e analogica/digitale, a sviluppatori di sistemi su microcontrollori dotati di I/O digitali ed analogici, ai progettisti di sistemi distribuiti basati su bus cablati, reti di comunicazione con e senza fili, che vogliano comprendere come integrare le loro applicazioni in sistemi multi processo e multi thread, e dotarle di connettività Internet e visibilità WEB.
Sistemisti, amministratori e sviluppatori GNU/Linux, che conoscono gli anfratti più reconditi del sistema operativo più libero del mondo, non hanno certo bisogno di queste note.
Sono comunque i benvenuti se vogliono soddisfare la curiosità di interfacciare il loro beniamino a "oggetti" fisici come periferiche di misura e controllo o altri microcontrollori, in modo diretto, via rete o via bus.

Cos'è questo libro

Un lungo viaggio inizia sempre con un primo passo. A volte si parte per una meta precisa, a volte si segue piuttosto un'idea, una convinzione, un'ispirazione o una visione. A volte si arriva in un posto dopo essere partiti per una destinazione completamente diversa. Stati, città e popoli nel Nord America portano ancora nomi che rispecchiano la convinzione di un coraggioso "innovatore" che voleva raggiungere le Indie via mare.
Molto spesso i migliori ricordi e le esperienze più significative sono quelle che si accumulano durante il cammino, come confermato anche dalla letteratura con i molti resoconti, appunti e diari di viaggio.
L'intento di questo libro, è di fornire un supporto con spiegazioni, progetti pratici, mappe e schede di riferimento per chi vuole avventurarsi per la prima volta in quel mondo trasversale, terra di frontiera e dai confini indefiniti che integra elettronica, sistemi a microcontrollori, periferiche hardware, e sistemi embedded, con il loro corredo di sistemi operativi e software d'ambiente e di sviluppo.

Cosa non è questo libro

Così come i diari di viaggio non sono manuali per imparare le lingue e non contengono mappe accurate di territori, stradari delle città attraversate, o elenchi telefonici degli abitanti, questo

libro non vuole essere un manuale per imparare a programmare con questo o quel linguaggio od un tutorial per imparare a configurare reti, amministrare sistemi GNU/Linux o sviluppare applicazioni "certificate". Si cercherà invece di dare il massimo delle indicazioni per orientarsi e districarsi nella maggior parte delle situazioni in cui ci si può trovare avventurandosi al di fuori dei confini "familiari". Una volta superate le difficoltà iniziali, dopo aver costruito con le proprie mani un'applicazione "embedded" completa e non proprio banale, ciascuno potrà proseguire lungo un percorso proprio e con propri obiettivi, seguendo proprie esigenze di crescita professionale o anche solo di semplice curiosità.

GNU/Linux e Raspberry Pi

Difficile dare un titolo ad un diario di viaggio. Nel nostro caso abbiamo voluto fissare la prima impressione dopo la prima prova con la board Raspberry Pi.
Il sistema operativo più complesso e diffuso del pianeta, motore unico e insostituibile dei supercomputer più inarrivabili è stato portato a funzionare in modo sostanzialmente identico su un computer *bonsai* delle dimensioni di una tessera da supermercato. E non è finita, anzi, con molta probabilità è solo l'inizio. A breve, a parità di dimensioni e costi, le potenze di elaborazione sono destinate a crescere in progressione geometrica. Merito dei processori ARM, in architettura RISC. Nati per gestire autoradio e lavatrici, hanno in seguito monopolizzato il mondo degli smartphone e dei tablet. Ora si affacciano sul mondo delle schede "tutto in uno", con prodotti dalle prestazioni di tutto rispetto a costi e consumi irrisori. Già si profila la prima "schedina" con una CPU a 64 processori da 45 GHz complessivi e 90 GigaFLOP di potenza di calcolo, delle dimensioni di 9 x 5 centimetri, sistema operativo GNU/Linux distribuzione Ubuntu. Vale proprio la pena di capirci di più perché nel nostro futuro, dove passeremo il resto della nostra vita, ci troveremo sicuramente a fare i conti con queste tecnologie.
Meglio, quindi, prepararsi per tempo con un viaggio di esplorazione che ci permetta di visitare i luoghi più significativi del "mondo" GNU/Linux facendo la conoscenza e toccando con mano gli "oggetti" principali, man mano che li incontriamo. Lasceremo anche il nostro "segno" lungo il percorso, un po' come gli ometti di sassi nei trekking, realizzando una nostra applicazione che ci permetterà di approfondire e "fissare" con maggior forza i riferimenti principali del viaggio consentendoci di affrontare con maggior consapevolezza e sicurezza viaggi simili, per hobby o per professione, anche in ambienti diversi basati su GNU/Linux, come le nuove schede sempre più potenti che stanno arrivando sul mercato ad un ritmo sempre più incalzante, o anche per muoversi meglio nella amministrazione e nella gestione di sistemi più grandi, non solo embedded ma anche gestionali o basati su web, sia ospitati in casa che in service.

La quota comprende

In questo libro vengono toccati, al livello di profondità necessario per "andare avanti", una serie di ambienti, applicazioni e linguaggi di programmazione, tutti rigorosamente utilizzabili con licenze open. L'hardware invece, bisogna comprarlo, che si tratti della scheda base o dei prodotti e materiali aggiuntivi per realizzare i progetti presentati.
GNU/Linux distribuzione Raspbian (una distribuzione Debian Wheezy compilata per processori ARM e Raspberry Pi in particolare, totalmente compatibile con Debian standard e distribuzioni derivate).
Funzioni varie di gestione pacchetti e manutenzione ambiente GNU/Linux:

- Desktop tradizionale
- Server SSH
- Server Xming
- Server Web Apache2
- Server database MySql
- Terminale a linea di comando e shell Bash
- Linguaggio Python

- Linguaggio PHP
- HTML5
- Javascript
- Ajax
- Framework emoncms
- Schedulatore Cron

Nella “borsa da viaggio” sono compresi i listati dei programmi e delle configurazioni necessari a realizzare il progetto di riferimento di questo libro.
Sono scaricabili liberamente dal sito di Elettronica In all’indirizzo www.elettronicain.it

La quota non comprende

In questo libro non è inclusa la scheda Raspberry Pi reperibile presso i rivenditori ufficiali RS Components (it.rs-online.com) e Farnell (it.farnell.com). Non sono inclusi il circuito stampato, ed i componenti hardware per realizzare la scheda di supporto al progetto da realizzare nel libro. I componenti sono facilmente reperibili un po’ ovunque mentre questa board è disponibile anche montata e collaudata presso Futura Elettronica (www.futurashop.it) col codice FT1060M, Euro 15,00 IVA compresa.
Sono esclusi caffè, bibite, tempo da dedicare, notti insonni, corti circuiti, e tutta la fatica e la determinazione necessaria a leggere, capire, provare ed acquisire concetti e argomenti di discipline diverse e sviluppare nuove conoscenze ed abilità.

Struttura e convenzioni adottate nel libro

Nella stesura di questo libro si è privilegiato il rispetto del percorso passo passo, corredato da immagini che, speriamo, aiutino a dissolvere dubbi e incertezze. Abbiamo approfondito alcuni argomenti, fornendo tabelle di comandi o schemi di funzionamento per quegli argomenti che riteniamo siano di utilità generale a chi si vuole interessare a GNU/Linux. Abbiamo inserito anche due “mappe di riferimento” dell’applicazione che andiamo a realizzare; una con lo schema generale dell’applicazione arricchita di riquadri con comandi utili, riferimenti all’hardware e alle configurazioni software, oltre agli spazi per ricordare utenti e password dei vari ambienti. La seconda mappa è dedicata alla struttura dati di supporto all’applicazione. Il consiglio è di fotocopiarle e tenerle sempre a “portata di mano”; un po’ come le cartine geografiche nei viaggi veri. Per quanto riguarda le convenzioni abbiamo voluto evidenziare i comandi da digitare a terminale con la grafica:

questo è un comando

I listati o parte di essi con la grafica:

```
questo è un listato
```

Ultime avvertenze

Le tecnologie ed i prodotti presentati in questo libro sono in continua evoluzione. Fino alla data di pubblicazione abbiamo mantenuto aggiornate le versioni di pacchetti e librerie utilizzate ma non possiamo prevedere qualche cambiamento di comportamento nei rilasci successivi. Nonostante la nostra attenzione può essere scappato anche qualche errore, della qual cosa ci scusiamo in anticipo.
Non ci prendiamo responsabilità per questo tipo di errori od omissioni.
Nei limiti del possibile, nel sito dove potete scaricare i listati degli esempi, inseriremo note di aiuto ed eventuali correzioni.
Bene, scarpe e abbigliamento comodi e ... buon viaggio!

2

Un po' di storia

Il bello della storia è che ci consente di analizzare i grandi cambiamenti ed i periodi di evoluzione come in un film, permettendoci di vedere come è andata a finire. L'ambientazione e gli attori possono cambiare ma la trama, in fondo, si ripete. E l'insegnamento che se ne può trarre è ben chiaro: "Nei momenti difficili, l'unione fa la forza".

Come ogni guida di viaggio che si rispetti non può mancare un'introduzione storica su ciò che si va a visitare, e anche noi ci adeguiamo volentieri.
Computer e sistemi operativi, macchine da calcolo all'inizio meccaniche, poi elettroniche a logica "cablata" all'interno, prima fisicamente, e infine "virtualizzata" in strati di software sempre più complessi. Le loro origini vengono da molto lontano. La vita nelle caverne non richiedeva grandi attitudini contabili, ma già le prime forme di organizzazione sociale hanno reso indispensabile sviluppare questa abilità. Nell'antichissima città di UR in Mesopotamia (**Fig. 1**), la silicon valley di 6.000 anni fa, citata nella Bibbia, era necessario contare pecore, mattoni, misure di grano, aree di terreni ed anche il trascorrere del tempo e delle stagioni. Le uniche macchine da calcolo disponibili erano le mani. Col pollice si contava toccando le diverse falangi delle altre dita. Tre falangi per quattro dita fa dodici (base di calcolo). Buffo no? Un metodo a dir poco protostorico ma, per quei tempi, decisamente pratico. Permetteva di calcolare facilmente le frazioni, infatti 12 è divisibile pe 2, 3 e 4 dando sempre risultati interi. Roba da museo? Un momento, che ore sono? Il giorno è diviso in 24 ore, 12 di luce e 12 di buio (la Mesopotamia sta abbastanza vicina all'equatore per permettere al metodo di funzionare), 60 minuti fanno un'ora (5 volte 12) e un minuto è composto da 60 secondi (a quei tempi non serviva, ma una volta che si è presa un'abitudine …) E provate un po' a contare le uova nelle confezioni del supermercato. Nel tempo altri popoli hanno sviluppato sistemi di conteggio e calcolo diversi, basati su diverse "intuizioni" e successivamente realizzato "macchine" basate sull'intuizione iniziale.
Alcuni hanno sviluppato metodi di calcolo in base 5 (dita di una mano) come Cinesi e Maya, dai primi è ancora in uso l'abaco (**Fig. 2**). Altri in base 10 (dita di due mani) o in base 20 (dita di mani e piedi). Alcuni si sono persi miseramente nelle solite complicazioni burocratiche come gli antichi Egizi ed i Romani; provate un po' a fare MDCCXXIV diviso CDXVII; fa un po' più di IV. Niente virgola e niente zero; nella Roma antica, chi sapeva fare qualche calcolo aveva assicurato un posto fisso nei Fori Imperiali (**Fig. 3**).


Fig. 1


Fig. 2

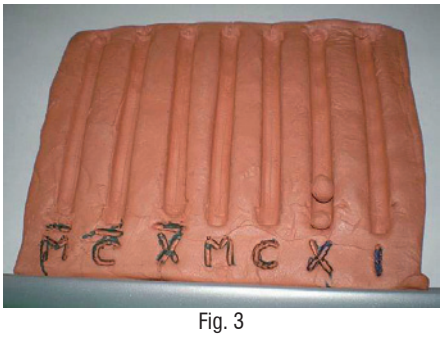


Fig. 3

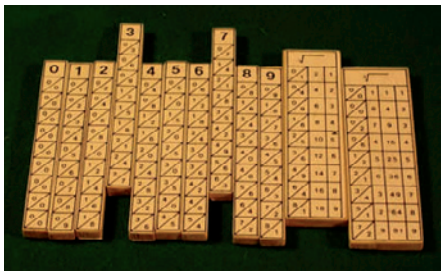


Fig. 4

Fig. 5

Fig. 6

Fig. 7

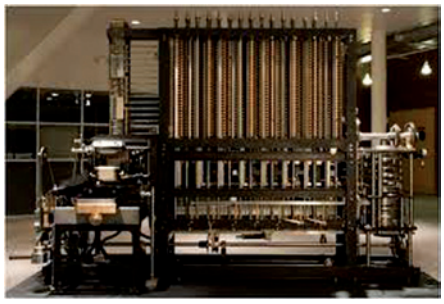


Fig. 8

Egizi e Romani a parte, ci si è sempre ingegnati di trovare ausili meccanici che facilitassero la fatica, le lungaggini e la possibilità di errori del calcolo manuale. Dopo l'abaco, che risale circa al 2000 AC si è fatta poca strada e bisogna aspettare il 1600 (DC) per incontrare i bastoncini da calcolo di Nepero, ideati per la costruzione delle tavole dei logaritmi (**Fig. 4**). La prima macchina calcolatrice fu ideata e costruita dal tedesco *Wilhelm Schickart* nel 1623: permetteva di effettuare le quattro operazioni aritmetiche ed era in grado di estrarre la radice quadrata di un numero. Successivamente tale macchina fu migliorata dal francese *Blaise Pascal* ed è passata alla storia come **Pascalina** (**Fig. 5**). Sulle sue orme il Signor *Gottfried Wilhelm von Leibniz* realizzò una macchina dotata del "*traspositore*", un meccanismo in grado di memorizzare un numero e di utilizzarlo per moltiplicarlo con il risultato di una successiva addizione. L'impresa era lodevole, ma come sempre, quando si vuole strafare, la scarsa precisione con cui vennero costruiti gli ingranaggi non consentì di ottenere risultati soddisfacenti. Durante le prove effettuate il risultato dei calcoli era praticamente sempre errato. Ciononostante il lavoro del Signor Leibniz rappresenta una pietra miliare nello sviluppo delle macchine da calcolo anche perché Leibniz è considerato "l'inventore" del codice binario, il metodo di calcolo in base 2 utilizzato - per ora - da tutti i computer del mondo. La prima macchina da calcolo prodotta in serie fu, nel 1820, l'**Aritmometro** di *Charles Thomas*, di cui furono venduti circa 1500 esemplari (**Fig. 6**), e che sfruttava gli studi e le tecniche sviluppate da Leibniz. Nel 1804 il francese *Joseph Jacquard* inventa il **Telaio Jacquard**, un tipo di telaio per tessitura che ha la possibilità di eseguire disegni complessi con i movimenti del telaio che sono "guidati" automaticamente da una serie di fori praticati su delle schede di cartone (**Fig. 7**).

Nasce così l'idea di **scheda perforata** per programmare le funzioni di una macchina, tecnologia che vide il suo massimo splendore negli anni '60 e '70. L'idea di quello che oggi chiameremmo il primo vero computer venne nel 1834 al matematico inglese *Charles Babbage*. La sua macchina richiederebbe un libro intero per descrivere il progetto ed il metodo di funzionamento, totalmente meccanico, composto da migliaia di parti in ottone lavorate a mano. Il merito principale di *Babbage* consiste nell'aver pensato ed iniziato a costruire una macchina strutturata molto simile agli odierni calcolatori. Se fosse stato costruito, sarebbe stato il primo computer del mondo, ma rimase incompiuto poiché il governo britannico, dopo aver inizialmente finanziato le ricerche, si rifiutò di concedere altri finanziamenti, ritenendo assurda l'idea che si potesse costruire una macchina in grado di eseguire un lavoro simile a quello della mente umana. Quella visibile e funzionante presso il Museo della Scienza di Londra (**Fig. 8**) è stata costruita negli anni '90 rispettando scrupolosamente il progetto e la tecnologia previste da *Babbage*. Da questo momento la crescita si fa tumultuosa, tra macchine completamente meccaniche come lo **Z3** dell'ingegner *Konrad Zuse*, visibile al museo della tecnica di Berlino (**Fig. 9**) e macchine con tantissimi relè come il **Mark-1 Colossus**. Fino ad allora i computer erano costruiti per svolgere un unico "programma" la cui logica era realizzata meccanicamente o cablata con migliaia di relè e valvole termoioniche. Capite bene cosa significava fare una modifica per cambiare una funzionalità o rimediare ad un errore. Nel 1941 avviene la svolta con il calcolatore **ENIAC 1** (**Fig. 10**) un mostro composto da 17.468 valvole termoioniche, 7.200 diodi, 1.500 relè, 70.000 resistenze, 10.000 condensatori e circa cinque milioni di connessioni saldate a mano. In questo caso, però si trattava di una macchina

"programmabile" dall'esterno, anche se non in maniera semplice. Uno dei "programmatori" iniziali fu il *Signor John von Neumann* (teorizzatore del modello di computer astratto noto come macchina di von Neumann). Anche questi erano comunque computer mono programma. Degni di nota sono l'**IBM SSEC** del 1948 (utilizzato poi per tabellare le mappe lunari e per assistere la missione lunare dell'Apollo 11 del 1969) e l'**ERA 1101** del 1950 che è stato il primo computer costruito per scopi commerciali e realizzato interamente da diodi.

La svolta all'impasse della monoprogrammazione l'ha data l'invenzione e il successivo impiego del transistor, che ha permesso di condensare in spazi molto più contenuti una quantità di logica impensabile in precedenza, dando la possibilità di affrontare il tema dell'elaborazione di più programmi sullo stesso computer nello stesso intervallo di tempo. Questo approccio richiedeva la progettazione e lo sviluppo di uno *strato software* di livello superiore, che potesse controllare e coordinare il funzionamento degli altri programmi, distribuendo a ciascuno le risorse necessarie. Questi programmi assunsero il nome di **Master Control Program** (MCP, ricordate il primo film TRON?), l'antenato dei moderni sistemi operativi.

Tra i primi sistemi operativi sviluppati negli anni '50, si può citare il **GMOS**, il sistema operativo della General Motors sviluppato per l'IBM 701, oppure il **FORTRAN Monitor System** (FMS) sviluppato dall'aviazione americana per l'IBM 709.

Ciascun produttore investiva in quegli anni ingenti somme per sviluppare sistemi operativi adatti alle architetture delle proprie macchine. Tra gli altri, negli anni '60, il MIT, i laboratori della AT&T Bell e General Electric iniziarono a sviluppare in consorzio il sistema operativo sperimentale *Multics* per il calcolatore **GE-645** (**Fig. 11**), ai tempi molto innovativo per quanto riguardava la flessibilità di gestione delle risorse e la scalabilità della configurazione. Purtroppo l'iniziativa non ebbe molto successo per via dei numerosi problemi incontrati nella progettazione e realizzazione. Ma l'idea era buona ed i laboratori AT&T Bell, lentamente, si allontanarono dal consorzio e iniziarono una riprogettazione del sistema su una scala più ridotta, che denominarono *Unics*. I ricercatori *Ken Thompson*, *Dennis Ritchie*, *M. D. McIlroy*, e *J. F. Ossanna* legarono il loro nome a questo progetto, sviluppando, insieme ad altri, su un PDP-7 usato (**Fig. 12**), il primo file system organizzato gerarchicamente, il concetto di processo, i file di mappatura dei dispositivi, oggi noti come **driver**, ed il primo interprete a linea di comando.

Il passo successivo fu lo sviluppo e l'integrazione del linguaggio C in *Unics*, che nel frattempo cambiava nome in **UNIX**. *Space Travel* è il video game che segna il raggiungimento degli obiettivi di UNIX come sistema operativo multiutente e multitasking, in grado di gestire la memoria virtuale, portabile, aperto e basato sul linguaggio C. Proprio apertura e portabilità sono le caratteristiche che ne garantiranno il successo; apertura soprattutto verso la comunicazione con gli altri computer, in quanto implementava nativamente il TCP/IP, il protocollo di **Internet** che vedeva la luce nel 1969; portabilità perché la sua architettura "a cipolla" prevedeva che tutta la gestione dell'hardware che lo ospitava fosse demandata ad un kernel centrale molto compatto e ben circoscritto nelle sue funzionalità (oggi si direbbe ben "incapsulato"). Con questa architettura, per portare UNIX su un diverso tipo di hardware ed adattarlo alla nuova architettura bastava riscrivere (portare) il solo kernel (o parte di esso) che allora era di circa 10.000 righe di codice in linguaggio C.

Vent'anni dopo, negli anni '90, sfruttando proprio le caratteristiche di porta-

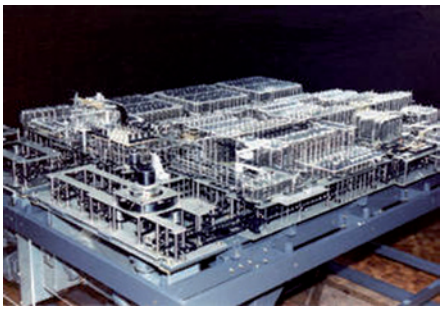


Fig. 9

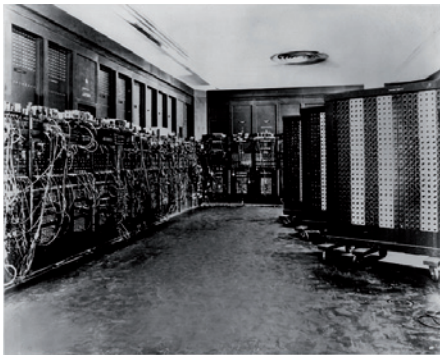
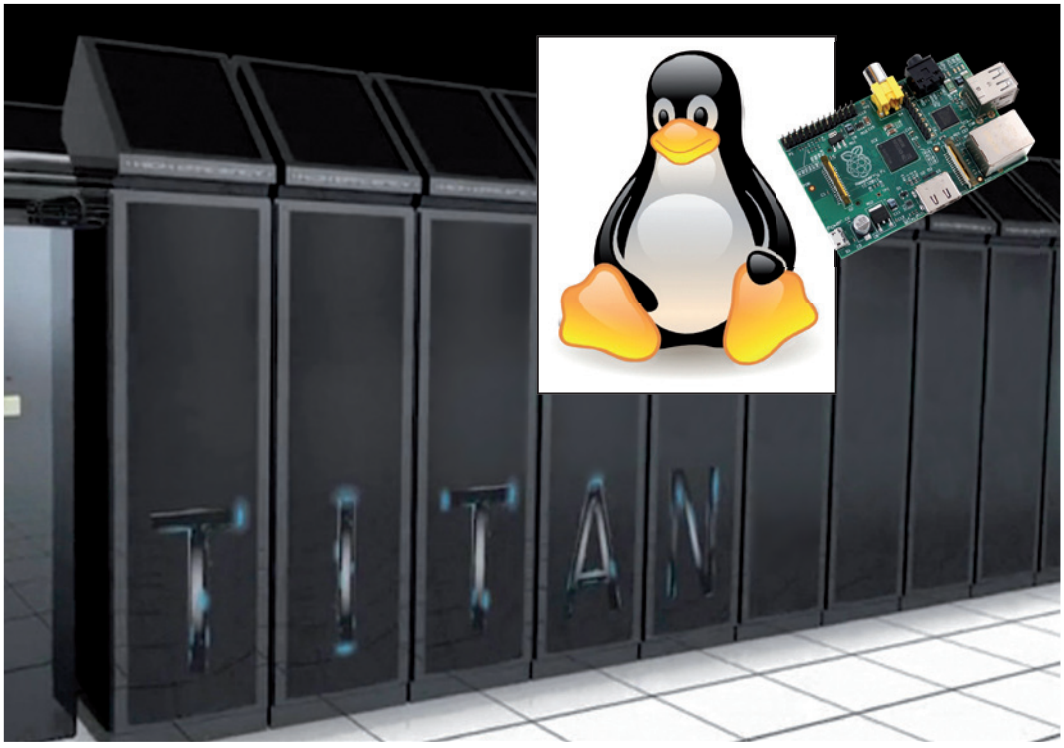


Fig. 10

Fig. 11

Fig. 12

bilità, *Andrew Tanenbaum* scrive una versione di UNIX, chiamata **MINIX** (MInimal uNIX) basata su un microkernel, che aveva la capacità di girare sui PC di allora.
Ed è proprio a questo progetto che, nei primi anni '90, si ispirò *Linus Torvalds*, il creatore di **Linux**.
Linux si è rapidamente diffuso da progetto di una singola persona a progetto di sviluppo mondiale grazie al coinvolgimento di una comunità di migliaia di programmatori e dalla adozione della **licenza GPL** (GNU General Public License). Sotto la GPL, il kernel GNU/Linux è stato protetto dallo sfruttamento commerciale ed ha anche tratto benefici dai programmi disponibili in user-space che facevano parte del progetto GNU (di *Richard Stallman*, che ha scritto tanto di quel codice che in confronto il sorgente del kernel GNU/Linux appare minuscolo).
Tutto ciò ha permesso una crescita esplosiva, la diramazione in diverse "distribuzioni" note come **Debian**, **Fedora**, **Ubuntu**, **Knoppix** e infinite altre, con l'inglobamento di altri progetti, package e applicativi, shell a linea di comando e ambienti grafici. Alcune in grado di partire direttamente da CD, da chiavetta USB o da "SD Card".
Alcune distribuzioni sono destinate a specifici campi di utilizzo, come la sicurezza informatica, firewall, gateway, router, grossi provider di servizi, supercomputer, fino alla gestione di macchine CNC o la gestione video e grafica. Tutto questo ha portato allo sviluppo di nuove professionalità, molto ricercate nel mondo del lavoro come specialisti di networking, di service providing e di sicurezza (questi ultimi spesso provenienti dalle file degli hacker).
Nel frattempo il "piccolo" kernel ha raggiunto la dimensione di circa 6 milioni di righe di codice e, grazie alla sua portabilità, ha potuto facilmente "migrare" anche sui processori ARM RISC, come dimostra il fenomeno **Raspberry Pi**. Nella nostra breve storia non abbiamo trattato la nascita dei PC storici, dei fenomeni Microsoft, Apple, Digital, Nixdorf, Honeywell, Olivetti, Univac, Sperry, Remington, Bull, Spectrum, Acorn, Atari, Commodore, HP, NOVEL e così via. Il motivo è che tutte queste realtà meritano di essere ricordate come tante storie individuali, magari grandi e di successo, ma private, proprietarie. La sintesi è che l'unico sistema operativo che si trova a proprio agio sia nel supercomputer TITAN che nel minuscolo Raspberry Pi è il sistema operativo del pinguino TUX.



3

Il mondo open source

Utilizzare software open source permette di aumentare le proprie conoscenze senza impegnare capitali in strumenti di sviluppo. Qualcosa che assomiglia molto a democrazia e libertà nel mondo della conoscenza.

Dopo una lunga maturazione nell'ambiente esclusivo di sistemisti, hacker e professionisti informatici, il sistema operativo GNU/Linux sta vivendo un momento di crescente popolarità tra gli utilizzatori comuni, fino ad ora pressoché totalmente orientati all'utilizzo di sistemi operativi di Microsoft ed Apple, almeno per quel che riguarda i personal computer. I primi segnali di questo fenomeno sono venuti alla luce quando alcuni utilizzatori istituzionali come amministrazioni statali e locali, soprattutto all'estero, ma anche aziende grandi e meno grandi, hanno deciso di eleggere il sistema operativo GNU/Linux, e più in generale il software open source, come l'architettura strategica di riferimento per il supporto alle proprie esigenze gestionali, informative ed informatiche. La prima motivazione di questa scelta è dettata dalla concreta esigenza delle istituzioni di proteggere i propri investimenti in tecnologie informatiche. E questa è maggiormente garantita se tutto il codice che costituisce il patrimonio informatico è disponibile in formato sorgente e quindi ispezionabile, riutilizzabile, modificabile e personalizzabile. Caratteristiche che sicuramente non appartengono ai sistemi "proprietari", che di conseguenza sono legati all'esistenza dell'azienda produttrice e contemporaneamente legano l'utilizzatore verso specifici fornitori. Per semplificare il concetto di protezione dell'investimento prendiamo il caso di un impianto industriale, come una raffineria, costruita con centinaia di componenti meccanici, idraulici, termici, di lattoneria, ecc. Tutti i disegni e la documentazione sono in possesso della azienda proprietaria dell'impianto, così che, quando si manifesta un guasto oppure in occasione della manutenzione periodica, operatori qualificati, non necessariamente appartenenti all'impresa che ha costruito l'impianto, possono intervenire riparando, sostituendo e, se necessario, addirittura ricostruendo parti dell'impianto. Ipotizziamo ora che questo impianto sia gestito da un sistema automatizzato di controllo di processo e quindi basato su una architettura hardware e software.

Se questo software è proprietario e l'azienda, a fronte di una ristrutturazione all'impianto, ha necessità di modificare anche parti del software proprietario, l'unica possibilità a sua disposizione è quella di rivolgersi al produttore originario, sperando che sia ancora in attività e possegga ancora le competenze e la conoscenza del sistema da modificare. Se queste condizioni non si dovessero verificare, la nostra azienda si troverebbe ad affrontare un problema che potrebbe comprometterne l'efficienza, la sicurezza, la competitività sul mercato e, nelle conseguenze più estreme, la sua stessa sopravvivenza.

Nel caso di utilizzo di sistemi open source, invece, come nel resto dell'impianto, si troverebbe in possesso di tutti gli schemi, della documentazione e del codice sorgente, ovvero di tutto il proprio patrimonio software e per di più nella condizione di poter contare su una vasta comunità di professionisti indipendenti per le manutenzioni e le modifiche.

La filosofia del software open source si fa risalire agli anni ottanta quando lo sviluppo software aveva già raggiunto livelli elevati, con tantissime aziende operanti nel campo dei personal computer e dei sistemi operativi e che si stavano diffondendo a macchia d'olio nei mercati.

In quegli anni erano molto diffusi i software proprietari, ovvero software forniti agli utilizzatori finali solo in formato eseguibile, senza i codici sorgenti, che restavano gelosamente custoditi all'interno delle aziende produttrici.

Si narra che in occasione di un cambio generazionale di computer al MIT, gli utenti non furono più in grado di personalizzare alcune funzioni di gestione remota delle stampanti, che consentiva, ad esempio di accorgersi dell'inceppamento della carta senza doversi muovere dalla scrivania.

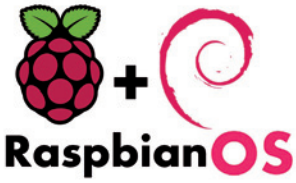
Questo e, molto più probabilmente il fatto che, grazie al mercato in forte crescita aumentava in modo esponenziale anche la mobilità degli sviluppatori, soprattutto in uscita dal MIT, portò all'idea della condivisione del software, per evitare di dover ricreare tutto daccapo ad ogni rimescolamento dei gruppi di lavoro. In questo contesto molti programmatori - fra i quali Richard Stallman che sarebbe diventato il portabandiera del software libero - si rifiutarono di lavorare per società private.

Stallman fondò nel 1985 la Free Software Foundation (FSF), un'organizzazione senza fini di lucro per lo sviluppo e la distribuzione di software libero, in particolare per lo sviluppo di un sistema operativo completo, compatibile con UNIX, ma distribuito con una licenza permissiva, con tutti gli strumenti necessari altrettanto liberi. Il riferimento era quello al progetto nato l'anno precedente, ovvero GNU, acronimo ricorsivo per ricollegarsi e contemporaneamente distinguersi da UNIX, ovvero "GNU's Not UNIX". «L'obiettivo principale di GNU era essere software libero. Anche se GNU non avesse avuto alcun vantaggio tecnico su UNIX, avrebbe avuto sia un vantaggio sociale, permettendo agli utenti di cooperare, sia un vantaggio etico, rispettando la loro libertà.»

Tale progetto, finanziato dalla FSF, venne pertanto prodotto da programmatori appositamente stipendiati. I principali contributi vennero da Stallman stesso, con il compilatore gcc e l'editor di testo Emacs, ma questo l'abbiamo già raccontato.

Abbiamo raccontato questa storia per invitarvi a sperimentare ed approfondire l'ambiente operativo GNU/Linux e le applicazioni disponibili in questo ambito, oltre che nella configurazione embedded che caratterizza Raspberry Pi, anche come sistema operativo per il vostro PC personale. Se inizialmente non volete dedicare un PC allo scopo, potete provare utilizzando una distribuzione live, ad esempio Debian, pressoché la stessa di Raspberry Pi, su chiavetta USB, dalla quale è possibile fare il boot direttamente e utilizzare GNU/Linux senza toccare un solo bit della vostra installazione originale. In questo modo potete utilizzare e configurare GNU/Linux, i pacchetti applicativi già disponibili ed installarne di nuovi. Se invece volete dedicare da subito una parte del vostro PC per sperimentare GNU/Linux potete orientarvi verso la distribuzione Ubuntu, una distribuzione destinata a far conoscere ed apprezzare il mondo "Linux" agli utenti comuni, che si sta ponendo come possibile alternativa in occasione dell'uscita delle nuove versioni dei sistemi operativi di Microsoft e Apple. Ricordiamo infine, per chiudere que-

sto capitolo, che anche nel nostro paese esiste una norma, precisamente la legge n. 134 del 7 agosto 2012, che impone alla Pubblica Amministrazione di privilegiare le soluzioni open source nelle proprie scelte di architetture, sistemi e applicazioni software,



Anche sul fronte Raspberry Pi si susseguono le novità, rappresentate dalla disponibilità di nuove distribuzioni del sistema operativo. Citiamo gli importanti upgrade a Raspbian, che permettono di utilizzare fino in fondo le potenzialità hardware di Raspberry Pi, sfruttando anche in modo automatico l'overclocking del processore, quando necessario. Tra le altre distribuzioni che stanno giungendo a maturazione citiamo Raspbmc, una personalizzazione di Raspbian dedicata a chi vuole impiegare la Raspberry Pi come gestore del proprio media center domestico, e la distribuzione Occidentalis, messa a punto da Adafruit, molto specialistica e destinata agli sviluppatori di sistemi embedded e che include tutti i driver ed i moduli necessari a gestire il GPIO ed il colloquio con dispositivi esterni che supportano i bus SPI, I2C, one wire e seriale.
Entrambe le distribuzioni sono basate su Raspbian ed in particolare la seconda è totalmente compatibile con le realizzazioni che vi stiamo proponendo in questo libro. Tutti possono contribuire ad arricchire il patrimonio di software libero in circolazione in molteplici forme: partecipando ad una comunità di sviluppo ufficiale o anche con contributi individuali. Non è detto che contribuire allo sviluppo di software open source significhi lavorare gratuitamente, anche se a volte, soprattutto agli inizi, può accadere. Molte comunità sono sponsorizzate da aziende che, per altri versi, sviluppano software proprietario, e chi vi lavora è regolarmente remunerato. In altri casi le ricadute economiche provengono da consulenze professionali e da attività di formazione, o ancora dalla fornitura di supporti hardware che realizzano le funzionalità del software libero. Un esempio di successo? Raspberry Pi, o Arduino, che molti di voi già conoscono. □

Finalmente... Raspberry Pi

4

Nato come computer di bassissimo costo per essere utilizzato quale supporto didattico nei paesi in via di sviluppo, ha conosciuto un successo che ha superato ogni previsione. Rispetto ai 10.000 esemplari previsti inizialmente ne sono stati prodotti più di un milione in un solo anno.

Preceduto da un carico di storia e di sviluppi scientifici e tecnici come quelli descritti nei capitoli precedenti, è ora di fare la conoscenza del nostro Raspberry Pi, sicuri che non saremo indotti a sottovalutarlo a causa delle sue dimensioni lillipuziane e del suo costo ridotto. Infatti, Raspberry Pi è un **vero e proprio computer** su scheda singola, delle dimensioni di una carta di credito. Per inciso *raspberry* significa "lampone" il frutto che è anche il logo della fondazione.

È stato sviluppato nel Regno Unito dalla Raspberry Pi Foundation *(http://www.raspberrypi.org/)* con l'intento di realizzare un computer a basso costo per stimolare l'insegnamento dell'informatica nelle scuole, in modo particolare nei paesi in via di sviluppo. La Fondazione aveva previsto un volume di produzione e vendita di circa 10.000 esemplari. Da subito, invece, è stato un successo planetario, tale da mettere in crisi i produttori per mesi e mesi. Al momento in cui scriviamo ne sono state prodotte più di un milione. Google Giving ne ha donate 15.000 alle scuole inglesi e la richiesta non accenna a diminuire. La board viene prodotta in due modelli, la versione B (**Fig. 1**) è quella più diffusa e recentemente ne è stata resa disponibile la revisione 2. La revisione 2 differisce dalla revisione 1 (**Fig. 2**) per la maggior dimensione della memoria RAM (512 MB nella rev. 2 rispetto ai 256 MB della rev. 1), per un paio di fori di fissaggio nella nuova revisione e, purtroppo, anche per alcune differenze nei collegamenti al connettore GPIO, che descriviamo in seguito. La versione A (**Fig. 3**) differisce dalla B per la mancanza del connettore Ethernet, la presenza di un solo connettore USB e la memoria RAM di 256 MB. Il cuore del computer è il SoC (System On a Chip) Broadcom BCM2835 che include un processore ARM1176JZF-S da 700 MHz, un processore grafico (GPU) VideoCore IV, un processore di segnali digitali (DSP) e 512 MB di RAM nella release 2, attualmente in commercio (**Fig. 4**). Non sono necessari dischi fissi o allo stato solido in quanto il sistema operativo e la memoria di massa sono allocati su SDCard che funge anche da unico device di boot, ma questo non toglie che possano essere comunque utilizzate anche unità di memoria di massa esterne.

Il testo di questo estratto gratuito finisce qui.

Il libro completo è acquistabile in edicola oppure on-line su:
www.futurashop.it

Cod. RASPBOOK1
Prezzo € 13,90