

Un Pinguino piccolo piccolo

Linux nei sistemi embedded

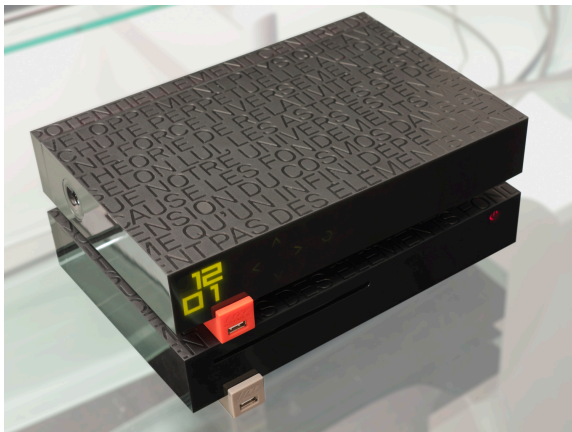
Luca Ceresoli

Linux Day — BgLUG — Sabato 24 ottobre 2015

Introduzione

Che cosa è un sistema embedded?

Router ADSL





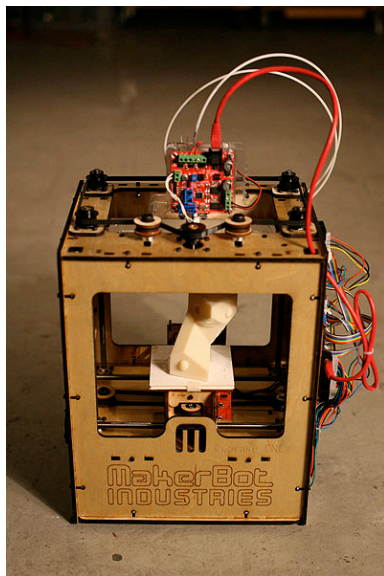
Terminale POS



Tagliatrice laser



Stampante 3D



Che cosa è un sistema embedded

- È un computer
- **incorporato** in un sistema
- programmato per **una specifica applicazione**
- con una piattaforma **hardware ad hoc**

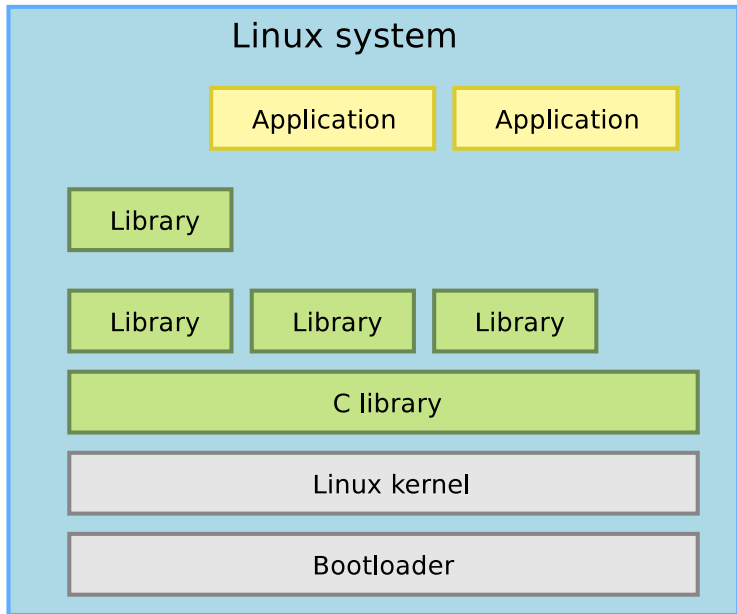
Embedded = piccolo?



- 65% of smart mobile devices
- 95% of high performance computing market
- **55% of embedded systems market**
- 40% of enterprise server market
- 90% of world's stock exchange

Source: Jim Zemlin, State of Linux, LinuxCon 2014, <https://www.youtube.com/watch?v=YdmT2ar0ZBw>

Anatomia di un Sistema Operativo



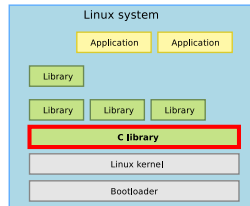
- Open Source
 - Nessun costo di acquisto (ma vanno rispettate le licenze!)
 - Esiste una montagna di software pronto da usare
 - Personalizzabile e adattabile in ogni sua parte
- Affidabile
- Efficiente
- Formato da molti piccoli pezzi
 - Quasi nessuno è obbligatorio
 - Molto sono sostituibili

Alcuni sistemi embedded Linux in funzione

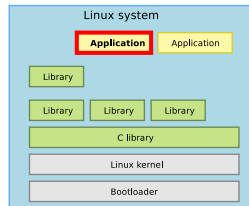
Sfida #1: Risorse disponibili

	PC	Android	Embedded
CPU cores	2..8	1..4	1..8
CPU clock	2..3 GHz	1..2 GHz	100 MHz .. 3 GHz
RAM	4..16 GB	1..4 GB	8 MB .. 16 GB
Storage	120 GB .. 6 TB	4..64 GB	8 MB .. 1 TB
Networking	1 GB/s, WiFi	WiFi	???
USB	3.0	2.0 .. 3.0	nulla .. 3.0

- GNU libc: ~2 MB
 - La più completa, e conforme agli standard
 - Usata nelle distribuzioni per PC e server
- uClibc-ng: < 1 MB
 - Configurabile: si possono disattivare alcuni componenti che non servono
- musl
 - Alternativa recente
 - Più compatta di uClibc-ng
 - Configurabile

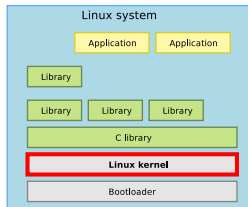


[, [, addgroup, adduser, adjtimex, ar, arp, arping, ash, awk, basename, bbconfig, bbsh, brctl, bunzip2, busybox, bzcat, bzip2, cal, cat, catv, chat, chattr, chcon, chgrp, chmod, chown, chpasswd, chpst, chroot, chrt, chvt, cksum, clear, cmp, comm, cp, cpio, crond, crontab, cryptpw, cttyhack, cut, date, dc, dd, dealloctv, delgroup, deluser, depmod, dfvfd, df, dhcrelay, diff, dirname, dmesg, dnsd, dos2unix, dpkg, dpkg_deb, du, dumpkmap, dumpleases, e2fsck, echo, ed, egrep, eject, env, envdir, envuidgid, ether_wake, expand, expr, fakeidentd, false, fbset, fbsplash, fdflush, fdformat, fdisk, fetchmail, fgrep, find, findfs, fold, free, freeramdisk, fsck, fsck_minix, ftpget, ftpput, fuser, getenforce, getopt, getsebool, getty, grep, gunzip, gzip, halt, hd, hdparm, head, hexdump, hostid, hostname, httpd, hush, hwclock, id, ifconfig, ifdown, ifenslave, ifup, inetd, init, inotifyd, insmod, install, ip, ipaddr, ipcalc, ipcrm, ipcs, iplink, iproute, iprule, iptunnel, kbd_mode, kill, killall, killall5, klogd, lash, last, length, less, linux32, linux64, linuxrc, ln, load_policy, loadfont, loadkmap, logger, login, logname, logread, losetup, lpd, lpq, lpr, ls, lsattr, lsmod, lzmacat, makedevs, man, matchpathcon, md5sum, mdev, mesg, microcom, mkdir, mke2fs, mkfifo, mkfs_minix, mknod, mkswap, mktmp, modprobe, more, mount, mountpoint, msh, mt, mv, nameif, nc, netstat, nice, nmeter, nohup, nsllookup, od, openvt, parse, passwd, patch, pgrep, pidof, ping, ping6, pipe_progress, pivot_root, pkill, poweroff, printenv, printf, ps, pscan, pwd, raidautorun, rdate, rdev, readahead, readlink, readprofile, realpath, reboot, renice, reset, resize, restorecon, rm, rmdir, rmmod, route, rpm, rpm2cpio, rtcwake, run_parts, runcon, runlevel, runsv, runsvdir, rx, script, sed, selinuxenabled, sendmail, seq, sestatus, setarch, setconsole, setenforce, setfiles, setfont, setkeycodes, setlogcons, setsebool, setsid, setuidgid, sh, shasum, showkey, slattach, sleep, softlimit, sort, split, start_stop_daemon, stat, strings, stty, su, sulogin, sum, sv, svlogd, swapoff, swapon, switch_root, sync, systemctl, syslogd, tac, tail, tar, taskset, tcpsvd, tee, telnet, telnetd, test, tftp, tftpd, time, top, touch, tr, traceroute, true, tty, ttysize, tune2fs, udhcpc, udhcpd, udpsvd, umount, uname, uncompress, unexpand, uniq, unix2dos, unlzma, unzip, uptime, usleep, uudecode, uencode, vconfig, vi, vlock, watch, watchdog, wc, wget, which, who, whoami, xargs, yes, zcat, zcip



- Un sistema Linux ha bisogno di molti comandi di base per funzionare
 - un sistema di init, una shell, un semplice editor di testo, programmi per manipolare files, esaminare lo stato del sistema, configurare la rete...
- In una distribuzione sono tanti eseguibili
 - Non pensati per un sistema embedded
 - Occupano spazio
 - Diverse fonti, vanno gestiti in modi diversi
- Soluzione: Busybox!
 - Unico eseguibile, molti comandi
 - Ottimizzato per un sistema embedded
 - Molto configurabile

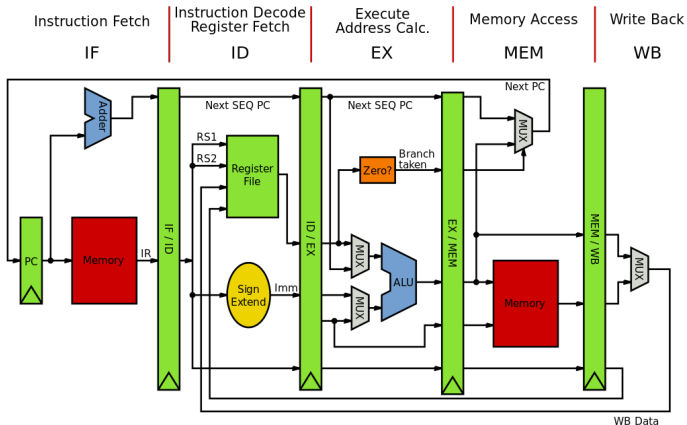
- È il nocciolo dell'intero sistema!
 - Relativamente piccolo ma estremamente importante
 - Grande potere (e grande responsabilità!)
- **Gestisce tutte le risorse hardware:**
CPU, memoria, I/O...
- Fornisce un'interfaccia **portabile**
all'hardware per le librerie ed applicazioni
- È mostruosamente configurabile



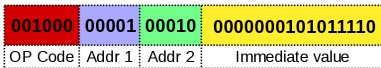
Configurazione del kernel

Sfida #2: Cross-compilazione

Architettura delle CPU



MIPS32 Add Immediate Instruction



Equivalent mnemonic: `addi $r1, $r2, 350`

Development PC

Tools
compiler
debugger
...

Embedded system

Application

Application

Library

Library

Library

Library

C library

Linux kernel

Bootloader

- Toolchain = strumenti per tradurre i sorgenti in eseguibili e librerie = compilatore + linker + altri strumenti
- Il compilatore standard del PC è una **toolchain nativa**
 - è un programma x86 che gira sul PC
 - produce un programma x86 che gira sul PC
- Per un sistema embedded (non x86) serve una **cross toolchain**
 - è un programma x86 che gira sul PC (“host”)
 - produce un programma ARM, MIPS... che gira sul sistema embedded (“target”)

Come mi procuro una cross-toolchain?

- Fai da te (e buona fortuna!)
- crosstool-NG
- Openembedded
- Buildroot
- Toolchain già pronta

Cross-compilazione

Sfida #3: Comporre un puzzle

Sistemi operativi standardizzati

- Distribuzioni Linux, Windows, Android...
- Molte librerie e servizi di base sempre presenti
- Le applicazioni possono contare su un ambiente sempre uguale

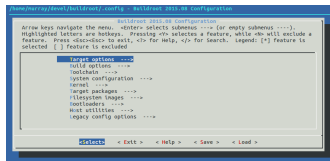
- Probabilmente c'è dell'hardware in meno
 - Non è detto che ci siano schermo, tastiera, mouse, touchscreen
- Probabilmente c'è dell'hardware in più
 - Controllo motori, lettore di carte magnetiche, telecomando, pannelli LED a scritta variabile...
- Non si può dar per scontato di avere librerie e servizi di base come su PC
- Spesso vanno realizzati in modo sartoriale
- Bisogna trovare i pezzi, sceglierli, assemblarli...
 - Possono essere molti
 - Può essere difficile farli cooperare correttamente

- Fai da te (e buona fortuna!)
- Build system
 - Buildroot
 - Openembedded
 - OpenWRT
 - ...
- Distribuzioni embedded pronte
 - Utilizzabile solo con hardware relativamente potente

- Buildroot — Making Embedded Linux Easy
- Strumento per generare tutti i componenti
 - Cross-toolchain
 - Bootloader
 - Kernel
 - Root filesystem: librerie, applicativi...
 - Contiene le “ricette” per compilare oltre 1700 pacchetti
 - File “immagine” scrivere in flash, SD...
- <http://buildroot.net/>

Buildroot: configurazione

- Si configura con kconfig come il kernel:
make menuconfig
- Permette di definire
 - L'architettura della CPU
 - Le caratteristiche della toolchain
 - Le applicazioni e librerie includere
 - I tipi di file immagine da generare
 - La configurazione del kernel
 - La configurazione del bootloader



Da zero a login con Buildroot

Domande?

© Copyright 2015, Luca Ceresoli
luca@lucaceresoli.net

<http://www.lucaceresoli.net>

Materiale rilasciato sotto licenza Creative Commons Attribution - Share Alike 3.0
<https://creativecommons.org/licenses/by-sa/3.0/legalcode>

Materiale da:

<http://free-electrons.com/doc/training/embedded-linux/>

https://en.wikipedia.org/wiki/File:Mips32_add1.svg

[https://en.wikipedia.org/wiki/File:MIPS_Architecture_\(Pipelined\).svg](https://en.wikipedia.org/wiki/File:MIPS_Architecture_(Pipelined).svg)

https://commons.wikimedia.org/wiki/File:NN-K125MBGPG_Grill-Mikrowelle_silber_Panasonic.png

https://commons.wikimedia.org/wiki/File:Bancomat_ATM_italy.jpg

https://commons.wikimedia.org/wiki/File:MakerBot_ThingOMatic_Bre_Pettis.jpg