

Buildroot

Making Embedded Linux Easy

Luca Ceresoli

luca@lucaceresoli.net

<http://lucaceresoli.net>

Linux Day 2018

BgLUG

Agenda

- Introduzione
- Buildroot
- Packages
- Conclusioni

Introduzione

Che cosa è un sistema embedded

Che cosa è un sistema embedded

- È un computer

Che cosa è un sistema embedded

- È un computer
 - **incorporato** in un sistema

Che cosa è un sistema embedded

- È un computer
 - **incorporato** in un sistema
 - programmato per **una specifica applicazione**

Che cosa è un sistema embedded

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

(https://it.wikipedia.org/wiki/Sistema_embedded)

Che cosa è un sistema embedded

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

(https://it.wikipedia.org/wiki/Sistema_embedded)

- Esempi

Che cosa è un sistema embedded

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

(https://it.wikipedia.org/wiki/Sistema_embedded)

- Esempi
 - Router ADSL

Che cosa è un sistema embedded

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

(https://it.wikipedia.org/wiki/Sistema_embedded)

- Esempi
 - Router ADSL
 - TV

Che cosa è un sistema embedded

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

(https://it.wikipedia.org/wiki/Sistema_embedded)

- Esempi
 - Router ADSL
 - TV
 - 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**

(https://it.wikipedia.org/wiki/Sistema_embedded)

- Esempi
 - Router ADSL
 - TV
 - Stampante 3D
 - Macchina tessile

Che cosa è un sistema embedded

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

(https://it.wikipedia.org/wiki/Sistema_embedded)

- Esempi
 - Router ADSL
 - TV
 - Stampante 3D
 - Macchina tessile
 - Videocitofono

Che cosa è un sistema embedded

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

(https://it.wikipedia.org/wiki/Sistema_embedded)

- Esempi
 - Router ADSL
 - TV
 - Stampante 3D
 - Macchina tessile
 - Videocitofono
 - Autoradio

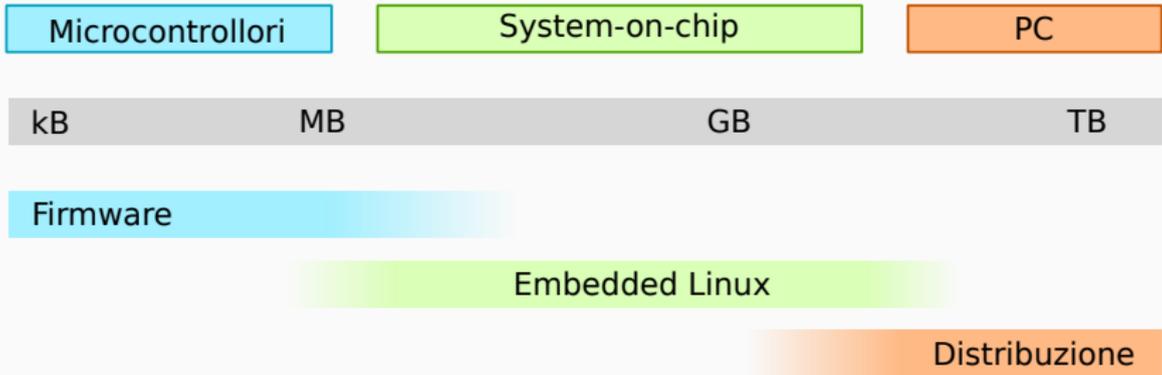
Che cosa è un sistema embedded

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

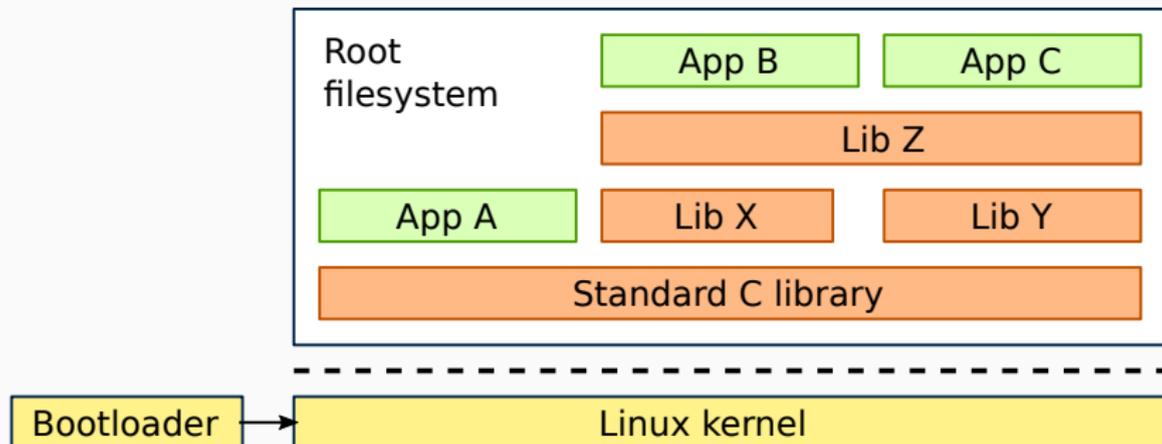
(https://it.wikipedia.org/wiki/Sistema_embedded)

- Esempi
 - Router ADSL
 - TV
 - Stampante 3D
 - Macchina tessile
 - Videocitofono
 - Autoradio
 - ...

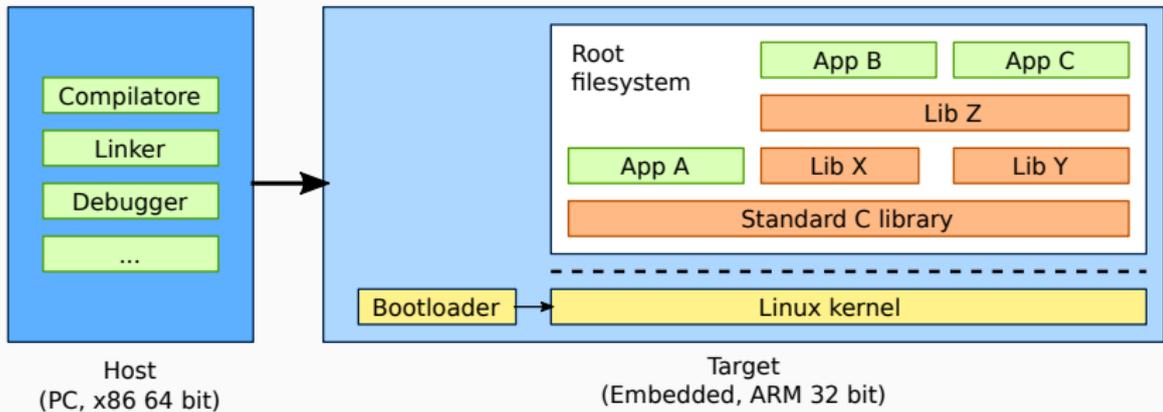
Sistemi embedded

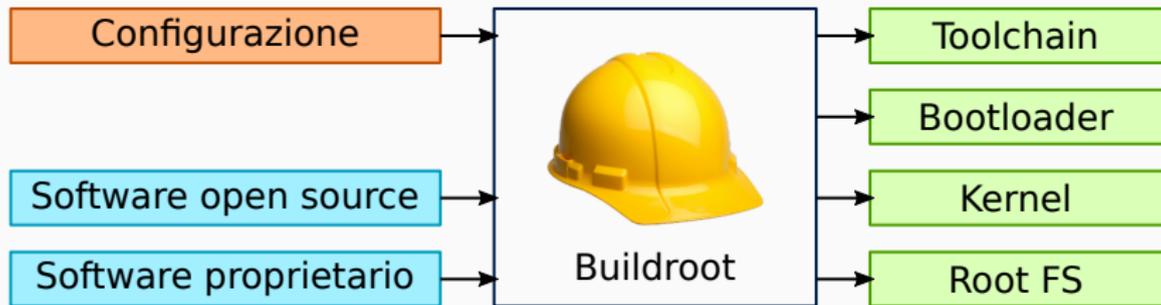


Anatomia di un Sistema Operativo



Host VS Target





Demo: la prima build

BeagleBone Black



- SoC: AM3358/9
- Core: ARM Cortex-A8 32-bit 1 GHz
- RAM: to 512 MB
- Storage: 4GB eMMC
- <https://beagleboard.org/black>
- https://en.wikipedia.org/wiki/BeagleBoard#BeagleBone_Black

<https://beagleboard.org/black>

CC BY-SA 3.0

Buildroot

- Buildroot — Making Embedded Linux Easy

- Buildroot — Making Embedded Linux Easy
- Strumento per generare tutti i componenti
 - Cross-toolchain
 - Bootloader
 - Kernel
 - Root filesystem: librerie, applicativi...
 - Contiene le “ricette” per compilare 2300+ pacchetti
 - File “immagine” da scrivere in flash, SD...

- Buildroot — Making Embedded Linux Easy
- Strumento per generare tutti i componenti
 - Cross-toolchain
 - Bootloader
 - Kernel
 - Root filesystem: librerie, applicativi...
 - Contiene le “ricette” per compilare 2300+ pacchetti
 - File “immagine” da scrivere in flash, SD...
- Obiettivi
 - Semplice da usare, capire, adattare
 - Snello, veloce, efficiente

- Buildroot — Making Embedded Linux Easy
- Strumento per generare tutti i componenti
 - Cross-toolchain
 - Bootloader
 - Kernel
 - Root filesystem: librerie, applicativi...
 - Contiene le “ricette” per compilare 2300+ pacchetti
 - File “immagine” da scrivere in flash, SD...
- Obiettivi
 - Semplice da usare, capire, adattare
 - Snello, veloce, efficiente
- Community project

- Buildroot — Making Embedded Linux Easy
- Strumento per generare tutti i componenti
 - Cross-toolchain
 - Bootloader
 - Kernel
 - Root filesystem: librerie, applicativi...
 - Contiene le “ricette” per compilare 2300+ pacchetti
 - File “immagine” da scrivere in flash, SD...
- Obiettivi
 - Semplice da usare, capire, adattare
 - Snello, veloce, efficiente
- Community project
- <http://buildroot.org/>



Configurazione

- Configurazione = *cosa* vogliamo che Buildroot produca



Configurazione

- Configurazione = *cosa* vogliamo che Buildroot produca
- Buildroot si configura con `Kconfig`
 - Il sistema di configurazione del kernel



- Configurazione = *cosa* vogliamo che Buildroot produca
- Buildroot si configura con Kconfig
 - Il sistema di configurazione del kernel
- Parametri configurabili
 - Tipo di processore
 - Toolchain
 - Bootloader
 - Kernel
 - Pacchetti da installare
 - Tipi di root filesystem
 - Opzioni di compilazione
 - Configurazione del sistema



- Utilizzo
 - `make help`
 - `make list-defconfigs`
 - `make <similar_board>_defconfig`
 - `make menuconfig`
- La configurazione si trova in `.config`
 - `make savedefconfig` — salva in forma compatta



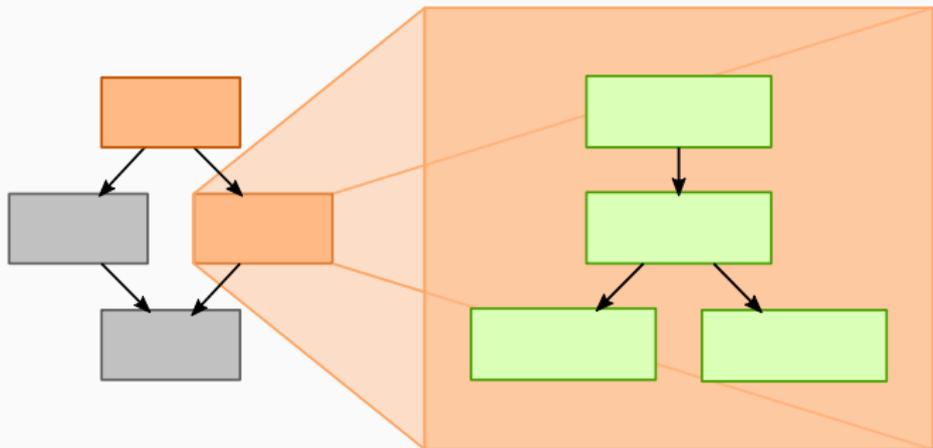
- Build = produzione di kernel, rootfs etc a partire dai sorgenti



- Build = produzione di kernel, rootfs etc a partire dai sorgenti
- Basato su Make
 - Lo strumento più usato per la compilazione di software
 - Gestisce le dipendenze tra pacchetti



Elementi della build



- Per ciascun pacchetto vengono eseguiti diversi passi:
 - Source (download dei sorgenti)
 - Extract
 - Patch
 - Configure
 - Build
 - Install

Demo: boot!

Demo: analisi dell'output

Packages

Un pacchetto semplice

```
buildroot
`-- package
    |-- netcat
        |-- Config.in
        |-- netcat.mk
        |-- netcat.hash
        |-- 0001-signed-bit-counting.patch
```

package/netcat/Config.in

```
config BR2_PACKAGE_NETCAT
```

```
bool "netcat"
```

```
depends on BR2_PACKAGE_BUSYBOX_SHOW_OTHERS
```

```
help
```

Netcat is a featured networking utility which reads and writes data across network connections, using the TCP/IP protocol.

[...]

<http://netcat.sourceforge.net/download.php>

package/netcat/netcat.mk

```
NETCAT_VERSION = 0.7.1
NETCAT_SITE = http://downloads.sourceforge.net/project \
              /netcat/netcat/$(NETCAT_VERSION)
NETCAT_LICENSE = GPL-2.0+
NETCAT_LICENSE_FILES = COPYING

$(eval $(autotools-package))
```

package/netcat/netcat.hash

```
# Locally computed:
```

```
sha256 30719c9a4ffbcf1...0c68a066f netcat-0.7.1.tar.gz
```

```
buildroot
`-- package
   |-- netcat
   |   |-- Config.in
   |   |-- netcat.mk
   |   |-- netcat.hash
   |   |-- 0001-signed-bit-counting.patch
```

- Patch = insieme di modifiche ai sorgenti del pacchetto
- Vengono applicate automaticamente
 - nello step patch (dopo extract)

Un pacchetto un po' più complesso

package/sl/sl.mk

```
SL_VERSION = 5.02
SL_SITE = $(call github,mtoyoda,sl,$(SL_VERSION))
SL_LICENSE = Custom
SL_LICENSE_FILES = LICENSE
SL_DEPENDENCIES = ncurses

define SL_BUILD_CMDS
    $(MAKE) -C $(@D) $(TARGET_CONFIGURE_OPTS)
endef

define SL_INSTALL_TARGET_CMDS
    $(INSTALL) -m 0755 -D $(@D)/sl $(TARGET_DIR)/usr/bin/sl
endef

$(eval $(generic-package))
```

Conclusioni

Grazie per l'attenzione!

Domande?

luca@lucaceresoli.net

<http://lucaceresoli.net>

© Copyright 2018, Luca Ceresoli

Materiale rilasciato sotto licenza

Creative Commons Attribution - Share Alike 3.0

<https://creativecommons.org/licenses/by-sa/3.0/>