

PIETRO LOREFICE

Sviluppatore Embedded

Introduzione a **SWUpdate** per l'aggiornamento di schede Linux

Techlab

03/06/2020

develer

I INTRODUZIONE

SWUpdate è uno strumento *gratuito e open-source* che fornisce una soluzione off-the-shelf per il problema dell'aggiornamento affidabile di un sistema embedded basato su Linux.

<https://github.com/sbabic/swupdate>

IL PROBLEMA

- | Criticità di un sistema Linux
- | Criticità di un sistema di aggiornamento
- | Esempio di sistema di aggiornamento

IL PROBLEMA

Linux è un sistema complesso

- | **Bootloader**
- | **Kernel**
- | **Root filesystem**
- | **Dati applicativi**

Aggiornare Linux in sicurezza è ancora più complesso!

- **Aggiornamenti corrotti**
- **Spegnimenti improvvisi**
- **Bug nel software di update**
- **Cancellazione dati utente**

Criticità di un sistema di aggiornamento

- **Preservare il bootloader**
- **Gestire eventi imprevedibili**
- **Garantire sempre l'uptime**
- **Preservare i dati utente**

I Un possibile approccio

- Doppio partizionamento (attiva + backup)
- Gestione partizioni tramite bootloader
- Download pacchetto di aggiornamento tramite rete (eg. HTTP)
- Client di aggiornamento *ad hoc* in userspace

| Un possibile approccio

SWUpdate fa già questo e molto altro!

SWUPDATE

- | Panoramica su SWUpdate
- | Funzionalità
- | Implementazione

Panoramica su SWUpdate

- | **Software updater per sistemi embedded**
- | **Flessibile ed estensibile**
- | **Gratuito e open-source**

Funzionalità di base

- | **Supporto per varie tecnologie di storage**
 - | **eMMC**
 - | **SD**
 - | **Flash NAND / NOR**
 - | **Flash SPI-NOR**

Funzionalità di base

- | **Installazione da vari tipi di sorgente**
 - | **Chiavina USB**
 - | **SD / microSD**
 - | **UART**
 - | **Server remoto**

Funzionalità di base

- | **Integrazione con vari bootloader**
 - | **U-Boot**
 - | **GRUB**
 - | **EFI**

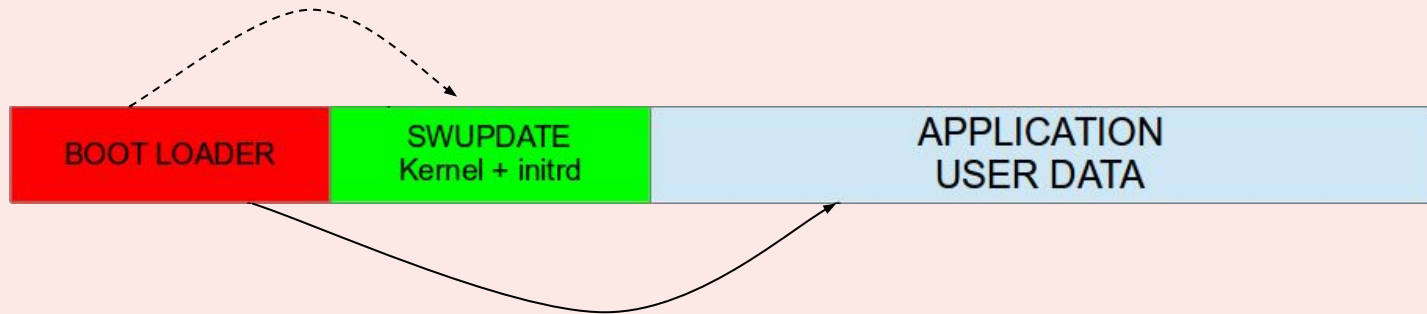
Funzionalità di base

- **Resiliente a spegnimenti**
- **Installazione da remoto**
- **Script di pre- e post-install**
- **Supporto per sistemi *single-copy* e *double-copy***

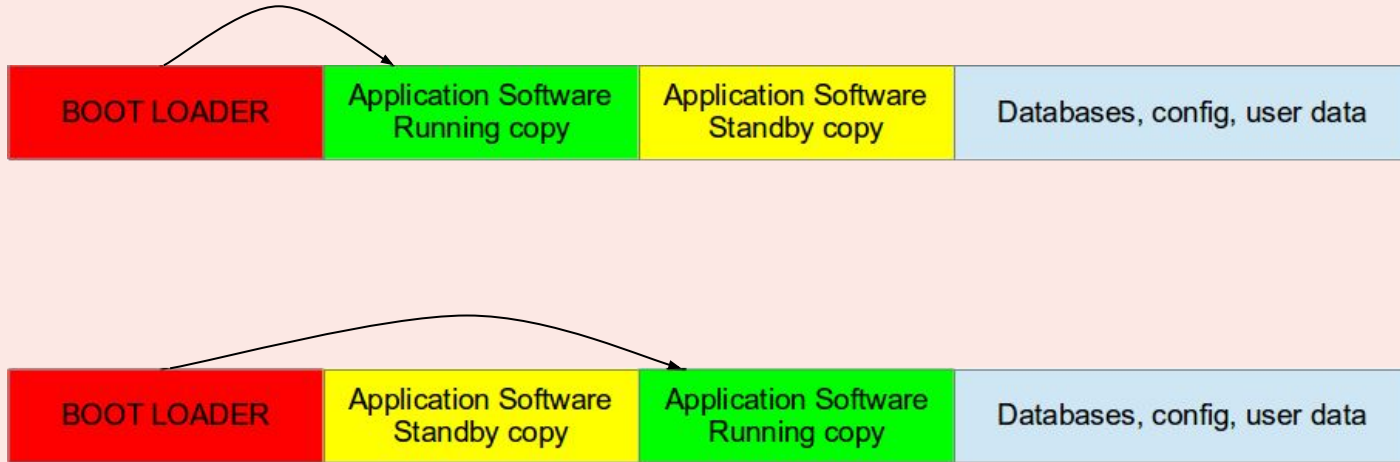
Funzionalità avanzate

- **Firma e verifica dell'aggiornamento**
- **Verifica di compatibilità tra SW e HW**
- **Estensibile tramite *handlers***
- **Integrazione con Yocto**

| **Single-copy vs. double-copy**



I **Single-copy vs. double-copy**



Single-copy vs. double-copy

- | **Affidabilità**
- | **Dimensione immagine**
- | **Necessità di riavvii**
- | **Compatibilità dati utente**

UTILIZZARE SWUPDATE

- Compilazione
- Creazione immagine
- Installazione sul target

Processo di build

- | **Basata su *Makefile***
- | **Segue l'approccio "kernel"**
 - | `make menuconfig`
 - | `make savedefconfig`
 - | `make`
- | **Supporto nativo per cross-compilazione**

I Configurazione

```
swconfig - swupdate configuration
Swupdate Configuration
Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus
----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M>
modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search.
Legend: [*] built-in [ ] excluded <M> module < > module capable
-----
| Swupdate Settings --->
|   Bootloader (U-Boot, ..) (U-Boot) --->
|   (/etc/fw_env.config) U-Boot Environment Configuration file (NEW)
|   (/etc/u-boot-initial-env) U-Boot Initial Environment file (NEW)
|   SSL implementation to use (OpenSSL) --->
|   [ ] Enable image downloading (NEW)
|   [ ] Allow to add sha256 hash to each image (NEW)
|   [ ] Enable verification of signed images (NEW)
|   [ ] Images can be encrypted with a symmetric key (NEW)
|   [ ] Suricatta (NEW) ----
|   [ ] Web Server (NEW) ----
|   [ ] Zstd compression support (NEW)
+   v(+)
|
|   <select>   < Exit >   < Help >   < Save >   < Load >
```

I Cross-compilazione

```
root@19 ~ # ./configure --help
> Resources Settings > Build Options

Build Options
Arrow keys navigate the menu. <Enter> selects submenu ---> (or empty submenu
----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M>
modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search.
Legend: [*] built-in [ ] excluded <M> module < > module capable
-----
() Cross Compiler prefix
() Path to sysroot
() Additional CFLAGS
() Additional LDFLAGS
() Additional LDLIBS
-----
<select> < Exit > < Help > < Save > < Load >
```

Dipendenze

■ **Necessarie**

- | libz
- | libcrypto
- | libconfig

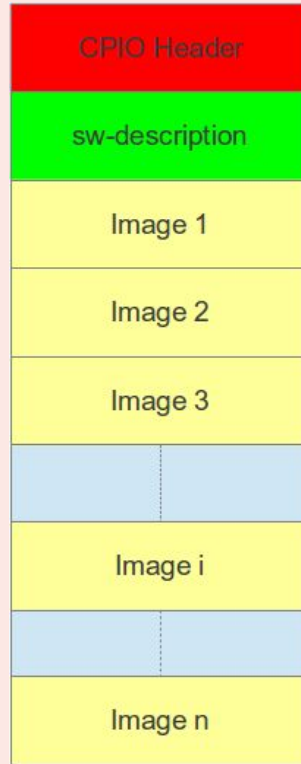
■ **Opzionali**

- | mtd-utils
- | openssl
- | libubootenv
- | libcurl

Creazione di un aggiornamento

- | **Archivio in formato CPIO**
- | **Estensione *.swu***
- | **Contiene metadati e file**

I Formato immagine



I Creazione immagine

```
PRODUCT_NAME="my-software"  
CONTAINER_VER="1.0"  
FILES="sw-description image1.ubifs image2.gz.u-boot uImage.binsdcard.img"  
for i in $FILES; do echo $i; done | cpio -ov -H crc > ${PRODUCT_NAME}_${CONTAINER_VER}.swu
```

I sw-description - single-copy

```
software = {  
    version = "0.1";  
    hardware-compatibility = [ "revA" ];  
  
    images: (  
        {  
            filename = "uImage";  
            volume = "kernel";  
        },  
        {  
            filename = "rootfs.ubifs";  
            volume = "rootfs";  
        }  
    );  
}
```

sw-description - double-copy

```
/* ... */
images: {
  main: {
    rootfs: ({
      filename = "rootfs.ext3";
      device = "/dev/mmcblk0p2";
    });
    uboot: ({ name = "bootpart"; value = "0:2"; });
  };
  alt: {
    rootfs: ({
      filename = "rootfs.ext3";
      device = "/dev/mmcblk0p1";
    });
    uboot: ({ name = "bootpart"; value = "0:1"; });
  };
};
```

UTILIZZARE SWUPDATE

Installazione aggiornamento

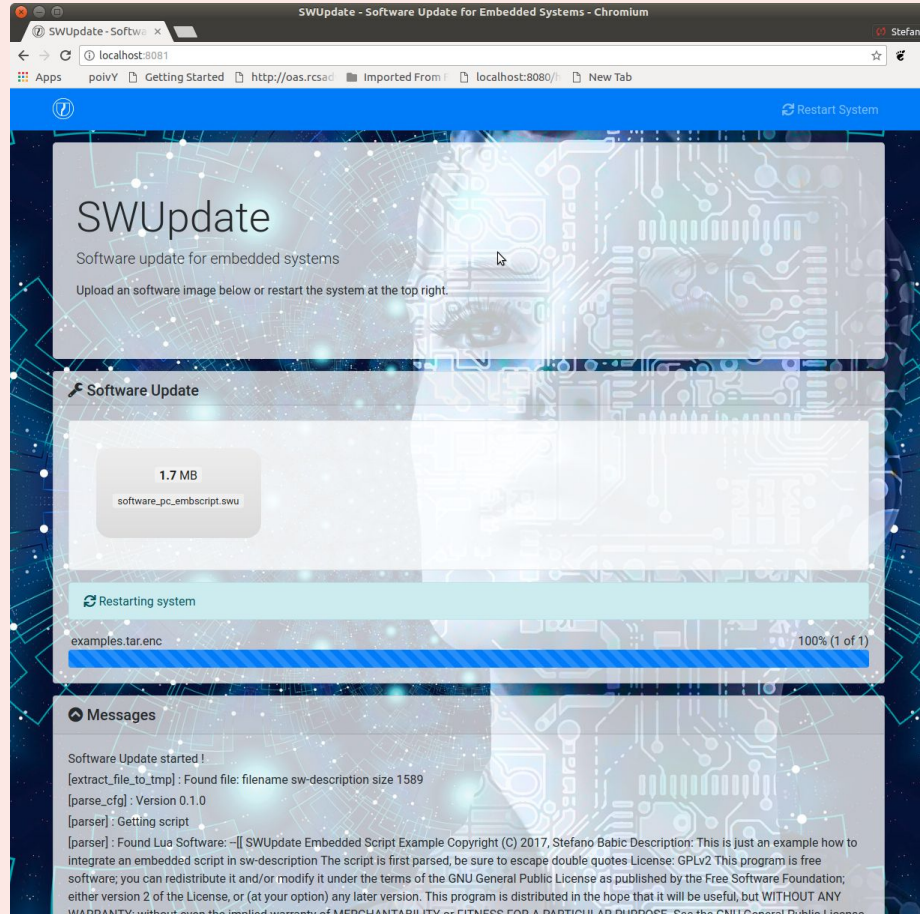
- **Mediante CLI**
- **Tramite interfaccia web**

I Installazione aggiornamento

```
$ swupdate -i <filename> # install image from a file
```

```
$ swupdate -w "--document-root ./www --port 8080" # start embedded web server
```

Web server



INTEGRAZIONE CON YOCTO

- Yocto?
- meta-swupdate
- Configurazione
- Generazione immagine

Yocto?

- **Ecosistema di tool per la creazione di distro custom**
- **Build offline e riproducibili**
- **Creazione di toolchain ad-hoc**
- **Estensibile tramite *layers***

meta-swupdate

- | **Integrazione con il processo di build**
- | **Deploy del binario sul target**
- | **Creazione del file *.swu***

I Configurazione - swupdate_%.bbappend

```
FILESEXTRAPATHS_append := "${THISDIR}/${PN}:"

PACKAGECONFIG_CONFARGS = ""

SRC_URI += " \
    file://defconfig \
    file://swupdate.cfg \
    "

do_install_append() {
    install -d ${D}${sysconfdir}
    install -m 644 ${WORKDIR}/swupdate.cfg ${D}${sysconfdir}
}
```

I Configurazione - swupdate.cfg

```
globals : {  
    verbose = true;  
    loglevel = 5;  
    syslog = true;  
    /* public-key-file = "test.pem"; */  
};  
  
download : {  
    retries = 3;  
    timeout = 1800;  
};  
  
identify : ({ name = "raspberrypi3"; value = "1.0"; });
```

I Configurazione - my-yocto-image-recipe.bb

```
# ...  
  
IMAGE_INSTALL += "\  
    swupdate \  
    swupdate-www \  
    "
```

Generazione immagine

- **Creare una nuova ricetta**
- **Farla derivare da *swupdate***
- **Includere *sw-description***
- **Indicare i file che faranno parte dell'immagine**
- **`bitbake update-image`**

update-image.bb

```
DESCRIPTION = "Example image demonstrating how to build SWUpdate compound image"
LIC_FILES_CHKSUM = "file://${COMMON_LICENSE_DIR}/MIT;md5=0835ade698e0bcf8506ecda2f7b4f302"
LICENSE = "MIT"

inherit swupdate

SRC_URI = "file://sw-description"

# images to build before building swupdate image
IMAGE_DEPENDS = "core-image-full-cmdline"

# images and files that will be included in the .swu image
SWUPDATE_IMAGES = "core-image-full-cmdline"

SWUPDATE_IMAGES_FSTYPES[core-image-full-cmdline] = ".ext4.gz"
```

I Risorse utili

<https://sbabic.github.io/swupdate>

<https://github.com/sbabic/swupdate>

<https://github.com/sbabic/meta-swupdate>

<https://github.com/sbabic/meta-swupdate-boards>

Pietro Lorefice

pietro@develer.com

Vuoi rimanere aggiornato sugli eventi Develer?
Seguici nei nostri canali social:



develer

www.develer.com