



OpenWRT

EmSysLab

Marco Boeris Frusca (marco.boeris@csp.it)

Torino, 21 Aprile 2009

Direzione Ricerca e Sviluppo

- Introduzione a OpenWRT :
 - Cosa è OpenWRT
 - Utilizzo (configurazione, selezione pacchetti, compilazione)
 - Architettura del software
- Mini Demo



Cosa è OpenWRT

- Sistema operativo progettato per sistemi embedded;
 - Minimalistic Busybox/Linux distribution GPL licensed;
 - Open source;
 - Supportato da una comunità attiva (Kamikaze 8.09);
- Utilizza tools e makefile per la costruzione del file system per sistemi embedded;
- Utilizzo di packages per la installazione di software (pacchetti ipkg)
- Genera un file immagine da copiare sulla compact flash del proprio sistema embedded



- OpenWRT crea un filesystem e un immagine da caricare sul sistema embedded consentendo di configurare:
 - tipo di sistema e architettura (X86, Atheros, Broadcom, MagicBox, ...)
 - Il tipo di immagine che si desidera creare (Ramdisk, tgz, ext2, cpilogz, ...)
 - I moduli da installare nel file immagine: Network, Kernel modules, librerie ...)



Requisiti OpenWRT

- Software e pacchetti necessari per l'installazione:
 - subversion
 - g++
 - libncurses5-dev
 - zlib1g-dev
 - bison
 - flex
 - unzip
 - autoconf
 - gawk
 - make
- Codice sorgente di OpenWRT



Download OpenWRT

- Dal sito di OpenWRT (<http://downloads.openwrt.org/kamikaze/>)

- Ultima revisione (development branch) da svn

```
svn co svn://svn.openwrt.org/openwrt/branches/8.09  
trunk
```

- Il contenuto verrà interamente salvato nella directory trunk;
- Nel caso si vogliano scaricare versione meno recenti o con una differente versione del kernel:

```
svn -q checkout -r #revision  
svn://svn.openwrt.org/openwrt/branches/8.09 trunk
```

- Ultima release è la Kamikaze 8.09 (Maggio 2009)



- Scaricato il codice sorgente occorre:
 - Aggiornare i feeds per il menu di configurazione:
cd trunk
./scripts/feeds update
 - Lanciare il menu di configurazione:
make menuconfig



- Il menu di configurazione di OpenWRT consente la selezione di :
 - Target
 - Packages
 - Kernel Module
 - Altre opzioni
- Assomiglia al configurazione del kernel Linux, con significato diverso per il concetto di modulo;
- Ogni opzione ha al massimo 3 scelte:
 - `<*>` : include nell'immagine del firmware
 - `<M>` : compila ma non include
 - `< >` : non compila e non include



- Per creare un immagine EXT2 per una macchina X86 occorre:
- Selezionare nel menu “target system” l’architettura:
 - X86;
 - Altre opzioni disponibili sono: AMD, Atheros, Atmel AVR32, Broadcom, Infineon, Intel, Magicbox, Mikrotik Routerboard, Sony Playstation 3;
- Selezionare il “subtarget”:
 - generic;
- Selezionare il target profile:
 - PCEngines Alix;



- Selezionare nel menu “Target Images” il tipo di immagine e filesystem da generare:
 - EXT2;
 - Altre possibili opzioni: Jffs2, squashfs, tgz, cpio.gz;
- E’ possibile configurare alcune opzioni dell’immagine:
 - Kernel partition size;
 - Filesystem part size;
 - Grub image;



Selezione packages

- OpenWRT dispone di numerosi moduli, librerie e applicazioni.
- Di default non è selezionato nessun modulo;
- E' necessario avere un'idea chiara di ciò che è necessario al proprio lavoro (evitando così di perdersi nel menu);



- Cosa succede quando si seleziona un package?
 - OpenWRT procede automaticamente al download tramite svn del package selezionato
 - Crea una cartella sotto /trunk/feeds/packages e in /trunk/packages
 - La cartella non contiene i file relativi al package ma semplicemente un Makefile e alcuni file di configurazione;
 - In fase di compilazione OpenWRT seguirà le istruzioni presenti nel Makefile



Compilazione packages

- Solo quando il package viene compilato, il pacchetto descritto nel Makefile viene effettivamente scaricato;
- L'archivio scaricato dal browser viene copiato in:
 /trunk/dl
- OpenWRT procede alla compilazione del package creando una cartella in:
 /trunk/build_dir/i386;
- Viene creato il relativo file.ipk (i file installabili di OpenWRT) nella cartella:
 /trunk/bin/packages.



- La compilazione di OpenWRT esegue tutti i Makefile dei package selezionati (Kernel e moduli)
- Quando avrà terminato il firmware risultante sarà nella cartella
 /trunk/bin;
- Vengono creati i file:
 - Openwrt-x86-ext2.fs (filesystem)
 - Openwrt-x86-ext2.image (file immagine)
 - Openwrt-x86-ext2.image.kernel
 - Openwrt-x86-ext2.vmlinux
- Il file da copiare sulla memoria (CF) del sistema embedded è Openwrt-x86-ext2.image



- Tutti i moduli selezionati saranno compilati;
- Quelli selezionati con <*> nel menu di configurazione saranno installati nel file immagine;
- I relativi packages compilati (file.ipk) si troveranno nella cartella /trunk/bin/packages;
- Comandi per la compilazione:
 - make
 - Make V=99 (verbose)
 - make world
- In fase di compilazione è consigliato utilizzare il comando verbose



- Architettura di OpenWRT:
 - I file immagine e relativi moduli compilati:
bin/
 - Moduli scaricati:
dl/
 - Makefile e patch dei package:
feeds/packages/
Package/



- Architettura di OpenWRT:
 - File sorgente dei vari package:
build_dir/i386/
 - Tutte le cartelle relative a compilatori, kernel Linux e librerie:

toolchain/
tools/
target/



Due possibilità

- Creazione di un "pacchetto" da inserire in packages
- Utilizzo diretto della toolchain e creazione di un pacchetto

Per la prima soluzione si rimanda a

<http://oldwiki.openwrt.org/BuildingPackagesHowTo.html>



I passi principali:

- Creare le directory
 - package/helloworld/ipkg
 - package/helloworld/patches
- Creare il file **Config.in** in helloworld
- Creare il file **Makefile** organizzato in più sezioni
 - Download dei sorgenti
 - Predisposizione per la compilazione
 - Compilazione
 - Creazione del pacchetto



Toolchain e pacchetti "a mano"

- I compilatori sono nella directory:
 staging_dir/toolchain-i386_gcc4.1.2/bin/
È sufficiente aggiungerla al path
Il nome dei compilatori è quello tradizione, preceduto da
i386-linux o da i386-linux-uclibc
Con il comando **file** è possibile verificare il tipo di
processore per cui è stato compilato l'eseguibile
- Per usarli direttamente è necessario
Intervenire sulle variabili di ambiente (ad es. GCC)
Modificare direttamente i Makefile
Eeguire **configure** con le indicazioni dei compilatori da
utilizzare



Creazione dei pacchetti

- Cosa contiene un file ipkg?
 - È un archivio creato con tar e gzip
 - Due ulteriori archivi: control.tar.gz e data.tar.gz
- Data.tar.gz
 - File da installare sul sistema, già organizzati nel file system

```
./  
./usr/  
./usr/sbin/  
./usr/sbin/openvpn  
./etc/  
./etc/default/  
./etc/default/openvpn  
./etc/init.d/  
./etc/init.d/openvpn
```



- `control.tar.gz`
Contiene le informazioni di controllo

File control:

Package: `openvpn`

Version: `2.0.9-2`

Depends: `kmod-tun, libopenssl, liblzo`

Provides:

Source: `feeds/packages/net/openvpn`

Section: `net`

Priority: `optional`

Maintainer: `OpenWrt Developers Team <openwrt-devel@openwrt.org>`

Architecture: `i386`

Description: `Open source VPN solution using SSL`



- Copiare tutti i file da includere nel pacchetto
- Organizzati in una struttura di directory che sarà quella definitiva
 - Ad es:
 - `/home/marco/pkt/etc`
 - `/home/marco/pkt/usr`
 - `/home/marco/pkt/usr/bin`
- In pkt creare la directory CONTROL
- Inserire al suo interno il file control
 - È possibile inserire altri file di controllo come conffiles
- Creare il pacchetto con
 - `ipkg-build -c -o root -g root`
- Per utilizzare tale script è necessario aggiungere al path la directory `staging_dir/host/bin`



Alcuni link utili

- OpenWRT:

<http://openwrt.org/>



1. Sistema di acquisizione dati da sensori
1. Sistema di gestione distribuita dei dati
1. Sistema di comunicazione tra processi attraverso l'utilizzo di una pipe e file dati.



1 - Sistema di acquisizione dati da sensori:

- Architettura multithread:
 - thread lettura dati da sensori (phidgets)
 - thread elaborazione dati e attuazione operazioni (es accendere un led su un'uscita digitale)
 - Scambio dati attraverso memoria per limitare gli accessi a disco e salvataggio periodico per creare uno storico



2 - Sistema di gestione distribuita dei dati:

- Architettura Client/Server:
 - Server su alix
 - Client su sistema esterno
 - Server:
 - Alla richiesta del client invia il contenuto di una cartella (file di log)
 - Se dimensione cartella supera una soglia cancella file vecchi
 - Client:
 - Richiede file e scarica dati



3 - Sistema di comunicazione tra processi:

- inviare segnalazioni da un processo produttore a un processo consumatore:
 - utilizzo di una pipe per comandi (es. leggi file)
 - utilizzo di file per scambio dati
- possibilità di avere 2 produttori e un consumatore



CSP innovazione nelle ICT

Sede

via Livorno 60 - 10144 Torino
Edificio Laboratori A1

Tel +39 011 4815111

Fax +39 011 4815001

E-mail: info@csp.it

Seconda sede operativa

Villa Gualino - Viale Settimio Severo 65
10133 Torino

www.csp.it

