

HAX: Hackers are experienced



La seconda parte degli articoli e gli approfondimenti li saranno pubblicati su www.haxonline.org

Nel mondo della carta stampata e negli sfondi blu assicuranti dei telegiornali gli Hacker sono davvero brutta gente

di Enzo Jetpac Aggazio

L'atto di modificare le funzionalità di un oggetto a proprio piacimento (dall'inglese to hack), è diventato nel corso del tempo un comportamento fuorilegge, da deprecare, anche quando l'oggetto è stato regolarmente acquistato.

Installare gli homebrew sul proprio Nokia N95, Apple iPhone, console Nintendo Wii, per fare alcuni esempi, consente di saltare le protezioni della macchina volte non a garantire il funzionamento e la sicurezza, ma a che le applicazioni che girano sulla macchina siano certificate dal potente produttore hardware, dietro corresponsione di un esoso contributo (fee) di ingresso. Una proposta, che i produttori di software non potranno rifiutare, secondo i nuovi Padri dell'industria dell'intrattenimento.

Brutta gente che si fregia del titolo di Hacker ha anche, in questi giorni (scrivo in Estate), violato le piattaforme del cloud computing, complesse applicazioni web che consentono da una parte di gestire i nostri dati, dall'altro li ritengono e manipolano e compiono statistiche e profilazione a nostra insaputa.

Eppure non li condivido, non sono d'accordo con questa visione dell'Hacker, del moderno pirata informatico che solca i mari di un oceano ormai vigilato dai pesanti cannoni della nuova Compagnia delle Indie.

Mi piace pensare che il mondo dell'Hacking, la figura dell'Hacker sia quella che presentiamo in queste pagine di Hax, approfittando della vetrina del Linux-Day 2009, tradizionale appuntamento annuale che

attrae curiosi e va ben al di là della presentazione di un singolo sistema operativo libero.

Gli Hacker che conoscerete in queste pagine di HAX non sono fuorilegge, non crackano sistemi e non si introducono nei dorati alvei di banche di denaro e informazioni che valgono oro.

Qui conoscerete il lato costruttivo e sarete coinvolti dall'energia positiva che si sprigiona dall'underground hacker, che prende vita spesso nei pressi della bistrattata università e dei centri di ricerca italiani.

Volenterosi ragazzi e ragazze (sempre più numerose), che creano prodotti secondo il modello Open, libero e aperto all'utilizzo, alla modifica, alla rielaborazione.

E in queste letture, seppur a volte criptiche e per iniziati, leggerete di Androidi che giocano a calcio, microcontrollori hardware aperti che fanno funzionare piccoli Robot, nonché lavatrici e frullatori, Firewall fatti in casa sotto l'occhio vigile della moglie, il recupero dei vecchi computer dismessi ma ancora funzionanti, la Top Ten dei videogiochi per il sistema operativo Linux.

La speranza è che in questi articoli leggerete pagine di futuro, ancora aperto e dove gli hacker producono idee e le condividono, un futuro costruttivo contrapposto all'idea fuorilegge propagandata dall'industria che intende chiudere circuiti, codici e (nostri) dati in costose scatole a pagamento.

Buona lettura!

ASSOCIAZIONE VERDEBINARIO

Non buttarmi, il mio chip batte ancora!

Verde Binario è un'associazione culturale nata nel 2002 e attiva nel Sud Italia, con sede a Cosenza. Il nome riflette l'interesse dell'associazione, focalizzato sulla relazione tra ambiente e tecnologia. Sin dalla sua nascita Verde Binario lavora a due progetti principali, entrambi consistenti nel recupero di materiale hardware informatico dismesso.

Il primo progetto, Museo Interattivo di Archeologia Informatica, è una mostra permanente di computer storici. Nella nostra esposizione i visitatori possono interagire con le macchine, utilizzando i sistemi operativi originali, i software e le interfacce dell'epoca, tra cui mastodontici mainframe, una timeline di PC, e console giochi. Il museo, attualmente ospitato nei locali della nostra associazione, è visitabile negli orari di apertura della sede. È possibile anche prenotare delle visite guidate scrivendo a museo@verdebinario.org.

Lo scopo di questo articolo è quello di facilitare il lavoro a chiunque voglia riprodurre questo esperimento e intraprendere una attività di trashware. Quella che segue è la nostra personale "ricetta" di recupero. Potete utilizzarla tutta o in parte, modificarla e redistribuirla nello spirito del Free Software.

Negli anni abbiamo imparato a nostre spese che nulla deve essere dato per scontato: anche quando i donatori assicurano il perfetto funzionamento di un PC, bisogna procedere ad alcuni test oggettivi sull'hardware, per evitare successive perdite di tempo.

La prima cosa da fare è dare un'occhiata alla scheda madre in cerca di condensatori esplosi o di altre anomalie. Visto che ci siamo, è il caso anche di eliminare un po' di polvere, aiutandoci con un aspirapolvere (o con un compressore), con uno spazzolino morbido (per pulire gli slot) e con una gomma bianca, con cui puliremo i contatti della RAM.

A questo punto si può passare a controllare l'alimentatore, utilizzando un multimetro o - meglio - un apposito tester ATX (si trova in commercio per circa 20 euro). È anche il momento di misurare il voltaggio della batteria CMOS e, se scarica, sostituirla con una nuova. Anche in questo caso avremo bisogno del multimetro.

Il passo successivo consiste nel verificare la RAM con memtest86+ (www.memtest.org), un software pensato specificamente per questo scopo. Memtest è disponibile sui CD di avvio di numerose distribuzioni GNU/Linux, tra cui la live System rescue CD (www.sysresccd.org), Ubuntu. Se durante il test alcune celle di memoria dovessero risultare difettose, sostituite la barretta senza esitazioni. Un'altra componente da controllare è il disco rigido. Qui ci viene in aiuto badblocks, un programma presente su tutte le distribuzioni. Al prompt dei comandi sarà sufficiente eseguire badblocks, specificando come argomento il disco rigido che vogliamo controllare, per ottenere la lista degli

eventuali blocchi difettosi. Anche in questo caso, se dovessero venir fuori dei difetti, il consiglio è quello di sostituire il pezzo: un vecchio disco che presenta blocchi difettosi non promette nulla di buono. L'ultimo test da effettuare riguarda la CPU. Anche qui ci viene in aiuto un semplice software: epuburn (pages.sbcglobal.net/redelm/), disponibile su diverse distribuzioni live, compresa la System Rescue CD. Il programma non fa altro che portare al massimo l'utilizzo del processore: se durante l'esecuzione il sistema si blocca completamente, è possibile che la CPU sia difettosa. Per avere dei risultati attendibili, vi consigliamo di far girare epuburn per almeno una decina di minuti.

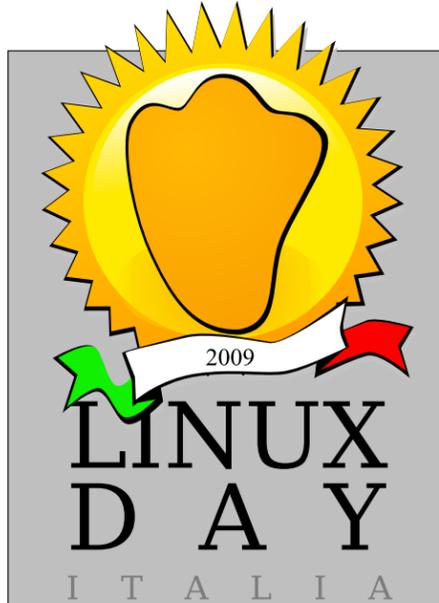
È importante ricordare che tutto il materiale guasto ed irrecuperabile va smaltito nel rispetto dell'ambiente, seguendo le norme sui Rifiuti di Apparecchiature Elettriche ed Elettroniche (RAEE, D.Lgs 151/05).

Il procedimento che abbiamo seguito per "sistemare a puntino" il nostro sistema Debian è lungo e meticoloso. Da subito ci siamo posti il problema di come replicare le installazioni: installare da capo il sistema operativo, i pacchetti, configurare e personalizzare il tutto per ciascun "nuovo" PC sarebbe un lavoro lungo, noioso e soggetto ad errori. A questo punto entra in scena Clonezilla (www.clonezilla.org), una utilissima distribuzione live che permette il backup ed il ripristino di partizioni o di interi dischi rigidi via rete.

In sostanza, quello che facciamo di solito è un backup dell'intero hard disk di un unico PC, il "master", installato e configurato al meglio. L'immagine disco, archiviata su un altro computer della nostra LAN, viene ripristinata sui nuovi pc, che diventano in pochi minuti dei veri e propri cloni dell'originale.

Poiché Linux carica i driver durante ogni avvio, la clonazione può essere tranquillamente effettuata anche su configurazioni hardware diverse (l'unica accortezza consiste nell'azzerare la configurazione di udev prima di effettuare il backup). Inoltre, le nuove versioni di Xorg rilevano la scheda video installata, caricando automaticamente il driver più adatto. Infine, Clonezilla consente, nella fase di ripristino, la clonazione su hard disk diversi dall'originale, ridimensionando opportunamente i filesystem. Una volta terminata la procedura di ripristino, il nuovo PC è pronto per il boot.

Per quanto le ultime evoluzioni delle interfacce utente abbiano permesso al grande pubblico di avvicinarsi al mondo Linux, l'approccio ad esso da parte dell'utente medio è ancora oggi poco immediato, soprattutto per chi è già abituato ad altri sistemi. Abbiamo poi considerato, dati gli scopi divulgativi che Verde Binario si prefigge, che la



Oggi 24 ottobre, in contemporanea in oltre 100 città italiane si svolgerà l'ottava edizione del LinuxDAY.

Una manifestazione promossa da ILS, Italian Linux Society, che coinvolge un gran numero di LUG e FSUG.

L'iniziativa ha lo scopo di presentare il Software Libero alle persone, cercando di mostrare sia la disponibilità di strumenti informatici sia le grandi risorse messe a disposizione dalla comunità per aiutare i principianti e anche chi principiante non è.

www.linuxday.it



Italian Linux Society

Sede legale:
Via Aldo Moro, 223
92026 Favara (AG)
Fax: 02 700 533 967

Codice Fiscale: 92043980090
Partiva IVA: 02438840841

Nao

il robo-calciatore con un cuore da pinguino

di Giuseppe Cotugno

Si chiama Nao, ma non è il cugino malefico di Keanu Reeves, è solo un calciatore, un calciatore robotico che ogni anno si cimenta, insieme ad altri suoi tre colleghi, nella Robocup, ovvero nei campionati di calcio robotico per la categoria Standard Platform League. Ogni squadra è composta da tre Nao più una riserva ed è programmata da un team di studenti, ricercatori e professori, alla competizione partecipano università da ogni parte del mondo. In verità, la Standard Platform non è l'unica categoria disponibile, esistono anche la Humanoid League, con robot di varie dimensioni interamente costruiti dalle università, la Middle-size e la Small-size, ove partecipano robot su ruote a grandezza umana o grandi quanto una noce di cocco. Non c'è solo il calcio-spettacolo, nella robocup c'è anche una competizione per le sonde robot (Rescue Robots), che dovrebbero trovare vittime in ambienti disastrati, ed il robot-maggiordomo (Robot@Home) che dovrebbe servire un umano in casa. Terminano il quadro le competizioni precedenti riprodotte su un simulatore.

Come mai, in tutto questo ben di Dio, ci occuperemo della cosa più semplice? Perché è quella che ha costi più abbordabili per un'università italiana (2000 euro per un Nao contro 100 000 per un robot-maggiordomo), inoltre i Nao saranno a breve commercializzati dalla azienda produttrice, la Aldebaran con sede a Parigi, e soprattutto perché il grosso del software che gira sul Nao è open, per cui è bene essere ben preparati ad accogliere questo nuovo prodotto quando sarà rilasciato sul mercato.

Innanzitutto parliamo un po' dell'hardware, il Nao dispone della bellezza di 25 gradi di libertà, ovvero giunti attuabili indipendentemente con il solito motorino in corrente continua, la struttura è alta circa un metro. L'autonomia del robot è, come al solito, abbastanza scarsa: tra la mezz'ora e un'ora, ma considerato il fatto che una partita dura 20 minuti non è un gran problema. Il robot è ben sensorizzato, dispone di due bumpers (una sorta di pulsanti) sui piedi, sensori di forza sempre sui piedi (utili per capire quando il Nao si è accapponato), ultrasuoni sul petto, microfoni omnidirezionali nelle orecchie, due telecamere 320x240 in formato YUV (che purtroppo si alimentano dallo stesso bus), giroscopi, accelerometri eccetera. Completano la struttura una porta

continua a pag.4

segue a pag.3

Il lato funzionale del Pitone

di Maurizio Lupo e il team di ReLabs

Python è un linguaggio multiparadigma. Possiamo programmare con uno stile imperativo (una istruzione dopo l'altra) oppure ad oggetti (incapsulando funzioni e dati in un'unica entità). Non tutti però sanno che Python ha preso in prestito e implementato molte idee della programmazione funzionale. Dedicando questo non voglio sottointendere che Python sia un linguaggio funzionale puro come Lisp o Haskell. Ad esempio non privilegia l'utilizzo della ricorsione come struttura primaria di controllo ma integra alcuni costrutti funzionali nella propria sintassi in modo elegante.

Ma andiamo per ordine, questo articolo vuole essere una panoramica di come l'aspetto funzionale si integra con il linguaggio. Prima di tutto è il caso di definire cos'è la programmazione funzionale!

Il paradigma funzionale vede le "funzioni" come le protagoniste! Possono essere passate come argomento ad altre funzioni, combinate, utilizzate per elaborare sequenze di dati, costruite a partire da altre funzioni. Le strutture di controllo sono spesso demandate a espressioni booleane (vedi gli operatori corto circuito nella seconda parte dell'articolo), i cicli sono spesso sostituiti dalla ricorsione. Il codice assume l'aspetto di una grossa espressione piuttosto che di una sequenza di comandi.

Ad un occhio non abituato, forse, può risultare un po' più difficile da leggere ma usato senza eccedere può rendere il nostro codice più compatto, chiaro: la ricchezza del linguaggio ci dà gli strumenti per scrivere del codice semplice ed elegante utilizzando sempre il paradigma che più si adatta al risultato che vogliamo ottenere.

Andiamo per ordine ed esploriamo gli aspetti "funzionali" del linguaggio e come si integrano con la sintassi imperativa e ad oggetti

Siamo abituati a pensare che le funzioni siano delle operazioni che restituiscono un valore. In effetti è così. Ma la differenza più evidente tra una funzione ed un oggetto è che un oggetto ha uno stato (il valore dei suoi attributi) mentre una funzione non ce l'ha.

In Python invece le funzioni sono oggetti "di prima classe" e possono tranquillamente avere degli attributi.

```
>>> def somma(a,b):
    return a+b

>>> somma(2,3)
5

>>> somma.nuovoattributo=10

>>> print somma.nuovoattributo
10
```

Normalmente non utilizziamo questi attributi ma possono tornare utili in qualche occasione. Ecco un esempio un po' più utile:

```
>>> def sommatore(a):
    if not hasattr(sommatore,"totale"):
        sommatore.totale=a
    else:
        sommatore.totale+=a
    return sommatore.totale

>>> sommatore(1)
1
>>> sommatore(2)
3
>>> sommatore(5)
8
```

Oggetti richiamati come funzioni

Le classi Python possono avere un metodo speciale `__call__`. Questo viene richiamato quando un oggetto viene utilizzato come una funzione, senza specificare il metodo o l'attributo.

Ad esempio:

```
>>> class ClasseSomma(object):
    def __call__(self,a,b):
        return a+b

>>> sommatore = ClasseSomma()
>>> sommatore(2,3)
5
```

Funzioni passate come argomento

Visto e dimostrato che le funzioni sono oggetti possiamo quindi utilizzarle come tali:

```
>>> def somma(a,b):
    .. return a+b

>>> altra_somma=somma
>>> altra_somma(3,4)
7
```

Possiamo anche passarle come argomento:

```
>>> def quadrato(x):
    .. return x*x

>>> def cubo(x):
    .. return x*x*x

>>> def somma_e_eleva(a,b,potenza):
    .. return potenza(a+b)

>>> somma_e_eleva(3,4,quadrato)
49

>>> somma_e_eleva(3,4,cubo)
343
```

Passando la funzione quadrato o cubo a somma_e_eleva possiamo e aggiungere in seguito altri tipi di operazione (ad esempio elevare alla quarta) senza dover modificare il codice scritto fino ad ora.

Le funzioni Lambda

L'istruzione lambda ci permette di creare delle funzioni anonime che possono essere utilizzate o passate come argomento.

La sintassi è:

lambda parametri : istruzione

La funzione lambda non può contenere più di una istruzione. Il risultato di questa istruzione verrà restituito senza bisogno del comando "return".

Questo costrutto si rivela utilissimo quando dobbiamo passare una semplice funzione come argomento.

Vediamo come poter rifattorizzare il codice precedente utilizzando questa sintassi:

```
>>> somma_e_eleva(3,4,lambda x: x*x)

>>> somma_e_eleva(3,4,lambda x: x*x*x)
```

Questo esempio ci mostra che, nel caso di funzioni molto semplici e richiamate solo in un punto, la funzione lambda ci permette di risparmiare alcune righe di codice e complessivamente risulta più leggibile.

Closures e decoratori

Python ci permette di definire delle funzioni all'interno di altre funzioni. Queste prendono il nome di closures:

```
>>> def elabora(numero,lista_addendi):
    def somma(a,b):
        return a+b
    lista_output=[]
    for addendo in lista_addendi:
        lista_output.append(somma(numero,addendo))
    return lista_output
```

```
>>> print elabora(5,[1,2,3])
[6,7,8]
```

Le closures rendono possibile la creazione dei decoratori: i decoratori costituiscono un classico design pattern della programmazione. Sono delle funzioni che aggiungono o modificano il comportamento di altre funzioni. Vediamo un esempio classico: il decoratore memoize:

```
>>> def memoize(func):
    def _memoize(*args):
        if not hasattr(func,'cache'):
            func.cache={}
        if args not in func.cache:
            func.cache[args]=func(*args)
        return func.cache[args]
    return _memoize
```

Ho definito una funzione memoize che prende come argomento una funzione. Definisco una closure di nome `_memoize`. Quest'ultima cerca in un attributo della funzione in input un dizionario "cache" e lo crea se non esiste. In questo dizionario verranno messi da parte i valori di ritorno della funzione "func" con determinati argomenti in input. Memoize restituisce infine la closure.

Questo decoratore permette di memorizzare i risultati di una funzione richiamata con determinati argomenti e di restituirne i risultati senza dover richiamare nuovamente la funzione.

Il decoratore si può applicare in questo modo:

```
>>> def add(a,b):
    return a+b

>>> add = memoize(add)
```

A questo punto richiamando `add(3,4)`, per due volte, la seconda volta il risultato verrà prelevato dalla cache.

Questa sintassi però ha un problema di leggibilità dovuto al fatto che la decorazione avviene al fondo e, nel caso di funzioni lunghe, non è subito evidente.

Fortunatamente in Python c'è una sintassi su misura per questo caso ricorrente:

```
>>> @memoize
def add(a,b):
    return a+b
```

<http://redomino.com/it/labs/>

Redomino Plone Tour

Giunto a metà del suo percorso, il Redomino Plone Tour può già tirare le somme: più di 200 presenze, 4 tappe svolte con successo, molto entusiasmo per le 3 tappe rimanenti!

Il Tour proseguirà infatti a ottobre, con una tappa a Torino martedì 6 e una a Marghera, in provincia di Venezia, martedì 20, ospitata dal CNA Veneto (Confederazione dell'Artigianato e della Piccola e Media Impresa); terminerà a novembre, con l'ultima tappa a Milano, mercoledì 11.

Per chi non conoscesse ancora questa iniziativa, ricordiamo che il Redomino Plone Tour è un ciclo di incontri gratuiti, che hanno come obiettivo presentare e diffondere conoscenza su una tecnologia open flessibile, potente e user-friendly: il CMS Plone.

Plone

Plone (plone.org) è un sistema per la gestione dei contenuti libero ed a sorgente aperta, distribuito sotto licenza GPL e basato sul server per applicazioni web Zope e sul linguaggio di programmazione Python. E' uno strumento che consente di creare, aggiornare e gestire i contenuti di siti web e intranet in modo semplice ed intuitivo, indipendentemente dalle proprie conoscenze informatiche; inoltre presta particolare attenzione all'usabilità ed all'accessibilità, rispettando la US Section 508, le linee guida del W3C in materia di accessibilità e le disposizioni in materia di Privacy (D.P.S.).

In altre parole, Plone è uno strumento che permette di gestire in modo autonomo un sito web, una intranet, un sito di e-commerce o di e-learning, indipendentemente dalle proprie conoscenze informatiche; ma allo stesso tempo è un ambiente di sviluppo flessibile, un sistema scalabile e integrabile da plasmare secondo le proprie esigenze.

Il Redomino Plone Tour ti aspetta per le prossime date:

- * Torino, martedì 6 ottobre 2009, dalle 10:00 alle 16:00
- * Marghera, martedì 20 ottobre 2009, dalle 10:00 alle 16:00
- * Milano, mercoledì 11 novembre, dalle 10:00 alle 16:00

La partecipazione è completamente gratuita, iscrizioni e maggiori informazioni all'indirizzo: redomino.com/it/promozioni/tour

ASSOCIAZIONE VERDEBINARIO

continua da pag.1

accortezza consiste nell'azzerare la configurazione di udev prima di effettuare il backup). Inoltre, le nuove versioni di Xorg rilevano la scheda video installata, caricando automaticamente il driver più adatto. Infine, Clonezilla consente, nella fase di ripristino, la clonazione su hard disk diversi dall'originale, ridimensionando opportunamente il filesystem. Una volta terminata la procedura di ripristino, il nuovo PC è pronto per il boot.

Per quanto le ultime evoluzioni delle interfacce utente abbiano permesso al grande pubblico di avvicinarsi al mondo Linux, l'approccio ad esso da parte dell'utente medio è ancora oggi poco immediato, soprattutto per chi è già abituato ad altri sistemi. Abbiamo poi considerato, dati gli scopi divulgativi che Verde Binario si prefigge, che la cultura del software libero possa e debba essere diffusa anche attraverso canali "concreti". Da qui l'idea di consegnare un sistema pronto all'uso corredandolo di un manuale cartaceo illustrato e di facile comprensione.

Il manuale che abbiamo approntato in occasione del progetto: "Un computer responsabile: non buttarmi, il mio chip batte ancora" cerca, in que-

st'ottica, di venire incontro alle esigenze degli utenti finali partendo dal presupposto che i nostri interlocutori non abbiano competenze tecniche, e perciò si rivolge ad essi nel modo più semplice possibile e rispecchiando fedelmente le situazioni pratiche che si troveranno ad affrontare quotidianamente.

Buon trashware a tutti!



Locator

un'applicazione Gps per il sistema operativo Android per gli smartphone

Marco Rizzo - HALT Hacklab Altomonte (CS)

Il progetto Locator è un esempio di applicazione realizzata per la piattaforma per smartphone Android. Esso consiste in un sistema di localizzazione costante tramite GPS, che visualizza la propria posizione sulla mappa aggiornandola costantemente e fornendo altre informazioni come Latitudine, Longitudine, Altitudine e Velocità.

Adesso creiamo il nostro progetto: File -> New -> Project selezioniamo l'etichetta "Android" ed infine selezioniamo "Adroid Project" e clicchiamo su Next.

Nella schermata successiva impostiamo il nome del progetto, package, il nome dell'Activity e quello dell'applicazione.

Finita la fase di creazione del progetto, possiamo catapultarci nello sviluppo dell'interfaccia grafica(UI).

In android l'interfaccia grafica di un'Activity viene in genere definita da una o più View con l'ausilio di vari tipi di layout (se ne possono creare anche di personalizzati).

Le varie View possono essere definite da codice oppure tramite un file xml.

In genere il file xml è quello più utilizzato oltre che consigliato in quanto meno complicato e velocemente modificabile. All'interno della directory /res/layout troviamo il seguente file main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/mainlayout"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <TableLayout
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:stretchColumns="1" >
        <TableRow>

            <TextView
                android:id="@+id/lat"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Latitude:"
                android:padding="3dip" />

            <TextView
                android:id="@+id/log"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="right"
                android:text="Longitude:"
                android:padding="3dip" />
        </TableRow>

        <TableRow>

            <TextView
                android:id="@+id/alt"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Altitude:"
                android:padding="3dip" />

            <TextView
                android:id="@+id/speed"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="right"
                android:text="Speed:"
                android:padding="3dip" />
        </TableRow>
    </TableLayout>

    <RelativeLayout
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content" >

        <com.google.android.maps.MapView
            android:id="@+id/mapview"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:clickable="true"
            android:apiKey="Your Maps API Key" />

        <LinearLayout
            android:id="@+id/zoomview"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignBottom="@+id/mapview"
            android:layout_centerHorizontal="true" />
    </RelativeLayout>
</LinearLayout>
```

Abbiamo appena scritto una UI con l'ausilio dei layout standard di android. Dal momento che non è obiettivo di questo articolo spiegare in modo approfondito cosa siano le View, si darà giusto qualche accenno sui layout utilizzati per il progetto: LinearLayout dispone i componenti figli in singole righe o singole colonne; RelativeLayout un layout in cui le posizioni dei componenti figli sono relazionate le une alle altre; TableLayout dispone i componenti figli in righe e colonne.

Si ha la possibilità di definire anche i vari attributi delle View utilizzando sempre la medesima struttura XML:

android:id	Fornisce un nome identificativo per questo tipo di component, per poi recuperare con View.findViewById () o Activity.findViewById (). Rappresenta una risorsa di riferimento; in genere si usa questa sintassi @+ per creare un nuovo ID. Ad esempio: android: id ="@+id/my_id" che ti permette di recuperare il component View in seguito con findViewById (R.id.my_id).
android:layout_width	Questo attributo definisce le dimensioni in larghezza sullo schermo. Se si desidera espandersi sull'intera dimensione dello schermo basta impostare il valore su "fill_parent".
android:layout_height	Questo attributo definisce le dimensioni in altezza sullo schermo. Se si desidera espandersi sull'intera dimensione dello schermo basta impostare il valore su "fill_parent".
android:orientation	Definisce l'orientamento del View che può essere settato in "vertical" o "horizontal"

<http://www.haxonline.org>

Raccogliere informazioni da una pagina HTML con XPath

di Luca Bruno

Molte volte troviamo delle informazioni preziose su dei siti web, in formato HTML o XML, e vorremmo estrapolarle ad esempio per indicizzarle in un nostro database o comunque farne altri usi particolari.

Immaginiamo un negozio online con una lista di categorie o di marche, e le vogliamo listare (anche quando sono più di 10 comincia ad essere noioso e meccanico), oppure una lista di IP e di porte, o delle descrizioni di bugs.

Quest'articolo ha lo scopo di organizzare un semplice, anzi semplicissimo, script generico per affrontare uno dei tanti scenari che si presentano in queste situazioni.

Avrete sicuramente capito che questo script potrebbe essere utilizzato per un web crawler, sia per ottenere i vari links delle pagine da scaricare, sia per processare i dati (e da qui il passo verso XSLT è breve).

XPath

XPath è l'ennesima diavoleria del consorzio W3, e permette di specificare dei path (simili a quelli del filesystem) al fine di ricercare uno o più specifici tags di un documento XML. La cosa che rende potente XPath è che non si tratta di un semplice path ma all'interno è possibile specificare una serie di condizioni sugli stessi tags da ricercare.

Qualche esempio:

```
<catalog>
  <product tested="true">Prodotto 1</product>
  <product tested="false">Prodotto 2</product>
  <product>
    Prodotto 3
    <desc>Descrizione 3</desc>
  </product>
  ...
</catalog>
```

Ecco qualche esempio di xpath:

/catalog	restituisce la lista di tutti i <catalog> che si trovano nella root del documento
/catalog/product	restituisce la lista di tutti i tag <product> che... insomma avete capito
/catalog/product[1]	restituisce il primo <product>
/catalog/product[@tested="true"]	lista di tutti i tag <product tested="true">
/catalog/product/@tested	tutti i valori degli attributi tested degli elementi product
//product	tutti gli elementi <product> del documento, anche innestati ecc.
//product/desc	tutte le <desc> dei <product>
//desc	tutti gli elementi <desc> anche al di fuori di <product>
//desc/text()	il testo contenuto in ciascun elemento <desc>
//product[desc]	tutti i prodotti che hanno un elemento <desc>

Con la @ quindi si indicano gli attributi. Attenzione, notate come con lo slash /, si definisce la "selezione" di ciò che vogliamo, mentre con le quadre [e] si specificano delle condizioni sul nodo contestuale.

Da notare che product[1] è il primo, non product[0], e in questo caso non restituisce una lista ma un singolo elemento perché il path non è ambiguo.

Capite bene che per ottenere la lista di tutti i links all'interno di una pagina XML, l'xpath è immediato:

```
//a/@href
```

Ok ma dove sta l'output? Beh questo è un linguaggio, e come tale per essere usato ha bisogno di un qualche interprete, che nella maggior parte dei casi è una libreria di un qualche linguaggio di programmazione.

Da HTML a XML

Prima di andare avanti dobbiamo capire che XPath lavora su documenti XML, quindi tag aperto con relativo tag chiuso, attributi con apici, ecc., tutte cose che non si trovano nei documenti HTML. Al più sarebbe possibile parsare direttamente un documento XHTML. Nel caso di HTML e broken HTML, ci serve un qualche strumento di conversione in XML. In questo i linguaggi di alto livello ci aiutano, in particolare useremo Python e la libreria LXML (che ha delle parti scritte in C).

Esempio d'utilizzo:

```
import lxml.html
import sys
tree = lxml.html.parse (sys.argv[1])
```

Chiamamo il file parser.py, scarichiamo una pagina.html e ora è possibile invocare genericamente il nostro miniscript nel modo seguente:

```
$ python parser.py pagina.html
```

A questo punto nella variabile "tree" avremo il nostro bel documento pronto per elaborare i nostri xpath.

Esiste una variante che è lxml.html.fromstring, dove si passa una stringa invece che il nome di un file (utile quando si scarica una pagina HTML con le sockets senza salvarla su disco).

Finalizziamo lo script aggiungendo alla fine:

```
print tree.xpath (sys.argv[2])
```

Perfetto, ora come secondo argomento possiamo passare una xpath e come output avremo qualcosa di pythoniano (poi sta a voi trasformarlo in qualcos'altro, oppure usare XSLT).

Esempi reali d'utilizzo

Qui abbiamo una lista di mailing lists <http://lists.debian.org/completeindex.html>, salviamola.

Vogliamo ottenere una lista di tutte le mailing lists. Analizziamo il codice html, e notiamo che si trovano tutte in corrispondenza di un <a>, che sta dentro un , che sta dentro un .

```
Un xpath potrebbe essere:
//ul/li/a/text()
```

Ma questo potrebbe essere ambiguo, poiché potrebbero esserci altri <a> in altre posizioni della pagina. A questo punto l'xpath è molto semplice, tanto vale complicarlo un pò per essere sicuri di prendere solo le mailing lists. Nella pagina vediamo che i sono almeno 10; un buon filtro potrebbe essere:

```
//ul[li[10]]/li/a/text()
```

In questo modo garantiamo che esista il 10° all'interno di , ovvero che ci siano almeno 10 .

Attenzione, se avessimo fatto //ul/li[10]/a/text() avremmo preso solo il 10° , invece di porre la condizione.

Eseguiamo il nostro script:

```
$ python parser.py completeindex.html "//ul[li[10]]/li/a/text()"
['cdwrite', 'debian-68k', 'debian-accessibility', 'debian-admin', 'debian-admintool', 'debian-alpha', 'debian-amd64', 'debian-announce', 'debian-apache', 'debian-arm', 'debian-autobuild', 'debian-beowulf', 'debian-blends', 'debian-books', 'debian-boot', 'debian-bsd', 'debian-bugs-closed', ...]
```

Ora fate una ricerca su Yahoo! (per cambiare un pò), e se non è cambiato il sito questo script di seguito dovrebbe darvi i titoli e gli url dei risultati principali:

continua a pag.4

Xpath da pag.3

```
import sys
import lxml.html
import urllib

baseurl = "http://search.yahoo.com/search?"
xpath = "//li/div/div/h3/a"

query = urllib.urlencode( { "p": sys.argv[1] } )
page = urllib.urlopen(baseurl+query).read()
tree = lxml.html.fromstring(page)
elements = tree.xpath(xpath)
for element in elements:
    print element.text_content(), "- ", element.get('href')
```

E ora vi lascio con qualche link interessante su xpath, xslt e lxml :)
 \$ python search.py "xpath lxml xslt"
 XML Path Language (XPath) - http://www.w3.org/TR/xpath
 XSL Transformations - Wikipedia, the free encyclopedia -
 http://en.wikipedia.org/wiki/XSLT
 XPath and XSLT with lxml - http://codespeak.net/lxml/xpathxslt.html
 Why XSLT 2.0 and XPath 2.0? - http://www.altova.com/XSLT_XPath_2.html
 ...

Conclusione

Nonostante a me personalmente non piace tutto ciò che sta intorno a XML, bisogna ammettere che finalmente è uscito qualcosa di interessante da questo mondo contorto.

In particolare, XSLT è un linguaggio a forma di XML (purtroppo si, avete sentito bene) che con l'ausilio di XPath trasforma un qualsiasi documento XML in un qualsiasi altro documento (molto utile per astrarsi dal linguaggio di programmazione). Nel caso di cui sopra abbiamo usato un "for" e quindi usato il linguaggio ospite; con XSLT questo poteva essere evitato. Tutto il resto ve lo lascio immaginare :)

<http://www.haxonline.org>

Stampato da  Fatto con Scribus

Impaginazione Rosa Fiori
<http://graficalibera.org/rosafiori/>



LinuxTrent - www.linuxtrent.it



<http://www.haxonline.org>



Attribuzione-Non commerciale-Non opere derivate 2.5 Italia

Tu sei libero:

di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire e recitare quest'opera.

Alle seguenti condizioni:

Attribuzione. Devi attribuire la paternità dell'opera nei modi indicati dall'autore o da chi ti ha dato l'opera in licenza e in modo tale da non suggerire che essi avallino te o il modo in cui tu usi l'opera.

Non commerciale. Non puoi usare quest'opera per fini commerciali.

Non opere derivate. Non puoi alterare o trasformare quest'opera, né usarla per crearne un'altra.

- Ogni volta che usi o distribuisi quest'opera, devi farlo secondo i termini di questa licenza, che va comunicata con chiarezza.

- In ogni caso, puoi concordare col titolare dei diritti utilizzi di quest'opera non consentiti da questa licenza.

- Questa licenza lascia impregiudicati i diritti morali.

Nao da pag.1

ethernet, una scheda wi-fi, degli speaker e delle lucette (utilissime per avere un feedback senza collegarsi al robot). Si può capire facilmente che un robot così sensorizzato ha del potenziale.

Il cuore "informatico" del robot è nella sua testa. Essa ospita, infatti, un processore AMD Geode LX 800 su cui gira un kernel Linux 2.6.22-9 avviato dal Grub; a sua volta il sistema operativo avvia un demone, il NaoQi che si occupa della gestione vera e propria di tutte le periferiche del Nao, e che dovrebbe essere l'unico punto di riferimento per un futuro acquirente. Sempre secondo l'idea di Aldebaran, la programmazione dovrebbe essere assistita da un programma, il Coreographe, che permette di elaborare movimenti ed azioni in maniera molto semplice, e ad occhio sembra riuscire, in pratica non ho mai avuto necessità di usarlo.

Difatti, nel laboratorio della Sapienza, a fianco al NaoQi utilizziamo un framework open-source, OpenRDK, sviluppato completamente in C++ all'interno dell'università, ma disponibile per la comunità su Source Forge. Ad esso abbiamo affiancato, talvolta, delle librerie di aiuto, come OpenCV per i problemi di visione.

In verità quanto proposto da Aldebaran è pacifico finché l'utilizzatore è il solito utonto a cui basta poco per apprezzare la piattaforma, considerato che il Nao viene usato per una competizione accademica, ogni università, o quasi, ha sviluppato la propria alternativa più o meno open, addirittura un'università tedesca ha completamente bypassato il NaoQi e questo non è stato molto gradito da Aldebaran, visto che il NaoQi si cura anche di evitare che i giunti vadano in saturazione o si danneggino a causa di un comando troppo "ambizioso".

OpenRDK è un framework altamente modulare, e non esistono versioni pacchettizzate. È necessario compilare il software da sorgente creando un makefile grazie al programma Cmake, tutto ovviamente open. OpenRDK carica dentro sè dei moduli C++ che implementano tre funzioni, una per inizializzare le proprietà condivise tra tutti i moduli, uno per inizializzare le sue strutture interne ed uno per eseguire il compito specifico (ad esempio raggiungere la palla). Tutte le variabili condivise vengono conservate in uno specifico oggetto, da cui si alimentano tutti i moduli, per cui nel framework è anche cablata un minimo di gestione della concorrenza.

Giustamente il lettore si chiederà, ma come è possibile che del codice compilato su una macchina Intel possa girare su un processore Geode? Ed infatti non lo fa! Qualsiasi cosa si voglia far girare sul Nao è necessario prima cross-compilarla sul pro-



prio PC per il processore Geode e poi caricarla sul Nao. In parole povere, prima si testa il funzionamento del codice sulla propria macchina linux, poi si ricompila tutto per il Geode e si copia sulla stick da 1 GB che sta nella capoccia del Nao.

Vediamo in pratica qual è la procedura che è necessario fare per eseguire del codice con un esempio.

Supponiamo che vogliamo riconoscere una palla arancione su un'immagine. Innanzitutto viene implementato un algoritmo di riconoscimento usando i tre metodi messi a disposizione da OpenRDK; dopodiché il codice viene compilato, corretto ed eseguito in locale sul proprio PC. Anche il codice viene eseguito da OpenRDK, in particolare da Regent, un programma che simula un agente, ovvero che esegue tutta l'infrastruttura di base di OpenRDK su cui poi viene caricato il modulo di visione.

Se si è soddisfatti, si ricompila il codice dalla propria macchina per il Geode (Cmake si preoccupa di generare un makefile che istruisca correttamente il compilatore), si copia il codice oggetto sul Nao e, finalmente si avvia Regent sul Nao caricando il modulo di visione.

Ovviamente, il Nao sarà in grado solo di vedere la palla e niente più, difatti nelle partite di calcio della robocup, i task fondamentali da eseguire sono trovare la palla, avvicinarsi, posizionarsi e calciarla, tutto senza accapottarsi. Certamente una partita è più complicata, è possibile far cooperare i robot per evitare che si buttino tutti sulla palla, è possibile fargli distinguere le porte, localizzarsi eccetera, ma per adesso ci accontenteremo.

OpenRDK è stato pensato per un uso generale, in laboratorio viene usato anche per programmare i Rescue Robots in simulazione e per gestire un quadricottero che monta un processore ARM, l'unico della Robocup capace, tra le altre cose, di riconoscere le linee a bordo, e forse un domani le finestre.

L'ultima robocup è stata letteralmente sbancata

Questo giornale in edizione speciale nazionale è stato ideato in collaborazione con ILS (www.linux.it) da LinuxTrent e da HackLab Cosenza in occasione del LinuxDAY 2009. I vari articoli provengono da diverse realtà italiane che utilizzano il Software Libero. Tutti gli articoli che per ragioni di spazio non sono presenti su questa stampa sono disponibili sul sito web del LinuxTrent (www.linuxtrent.it) e sul sito web di HAX (hax.cosenzainrete.it).

Sponsorizzato da

ILS - Italian Linux Society - www.linux.it

Open VPN

ovvero *Virtual Private Network made in Linux!*

di Antonio Gentile - Hacklab Cosenza

Una VPN (Virtual Private Network) si usa quando si ha la necessità di creare un collegamento tra due o più reti private attraverso una rete pubblica (come internet). Una volta stabilita la connessione tra le 2 reti private, gli utenti vedranno la rete opposta in modo del tutto trasparente come se fossero collegate fisicamente tra di loro. Bisogna tenere però conto che la velocità di connessione massima tra le due reti è definita dalla rete pubblica: se usassimo una connessione Adsl normale con banda in upload di 256kbs ad esempio, la velocità massima di trasferimento dei file sarà circa 32kbyte al secondo, e non i classici 10/100mbit della LAN. Altra caratteristica molto importante tipica di OpenVPN è quella di creare un sistema di comunicazione protetto: potete stare quindi tranquilli anche nel caso dobbiate trasferire dati confidenziali e riservati.

Privata perché offre un servizio praticamente identico ad una LAN privata.

Virtuale perché in realtà il flusso dati passa dalla rete pubblica (Internet).

Serve: alle aziende per collegare sedi staccate senza aver bisogno di connessioni dedicate; a due comuni utenti per condividere delle risorse attraverso internet; ad un admin per amministrare una macchina remota; a rendere confidenziale una rete condivisa (WLAN o LAN su hub).

VPN: come funziona?

Tutto il traffico fra i due endpoint della VPN viene incapsulato in tunnel predefiniti.

I tunnel possono essere su diversi livelli della pila ISO/OSI: IPSEC (a layer 3), PPTP (a layer 3), OpenVPN, l2tp, vtund, cipe, etc. (a layer 4).

OpenVPN usa i device tun/ tap, interfacce virtuali che permettono ai programmi in user-space di spedire/ ricevere pacchetti. Ogni pacchetto mandato sul tunnel viene criptato (confidenzialità)

Tun: device virtuale IP punto-a-punto, trasporta pacchetti IP, simile all'interfaccia ppp.

Tap: interfaccia ethernet virtuale, trasporta frame ethernet, simile all'interfaccia eth.

Se si ha bisogno di broadcast ethernet, protocolli non-IP (Netbios), o creare LAN virtuali si deve forzatamente ricorrere a tap.

Ci sono tre modalità di funzionamento: Tunnel non criptato, Crittografia a chiave condivisa (shared-key), TLS: autenticazione e scambio di chiavi.

TLS

1. Stabilisce un canale dati e un canale di controllo sopra una singola porta tcp/udp

2. Canale di controllo: mutua autenticazione con certificati e Diffie-Hellman per scambio di chiavi (sicuro!)

3. Canale dati: crittografia e autenticazione simmetrica (veloce!) Durante l'handshake un host avrà il ruolo di server l'altro (gli altri) di client.

Certificati X.509

In un sistema basato sui certificati ogni host deve disporre di una coppia chiave privata/ certificato.

CA: entità di cui tutti gli host si devono fidare. Garantisce l'identità degli host coinvolti.

Firma le richieste di certificato. Pubblica il proprio certificato che deve essere disponibile a tutti i partecipanti alla VPN.

Riassumendo

OpenVPN aiuta a creare VPN in modo semplice, e non soffre dei problemi di IPsec (nat)

L'infrastruttura TLS offre un ottimo grado di sicurezza e scalabilità

È utile in diversi contesti

È in costante e rapida crescita

Funziona su linux, *BSD, Win2k/WinXP, MacOSX, Solaris...

Quali sono le differenze tra modalità bridging e routing?

Bridging e routing sono due sistemi per collegare sistemi con VPN.

1. Vantaggi Bridging

1. I Broadcasts viaggiano sulla VPN -- Questo permette ai software che fanno uso di broadcast di funzionare, come ad esempio la condivisione dei file di windows o i giochi in lan, di funzionare.

2. Non sono necessarie configurazioni di routing.

3. Funziona con tutti i protocolli che viaggiano su ethernet incluso IPv4, IPv6, Netware IPX, AppleTalk, etc.

4. Relativamente facile da configurare per soluzioni roadwarriors (molteplici sedi distaccate, utenti che si collegano da posti diversi)

2. Vantaggi Bridging

1. Meno efficiente rispetto la modalità routing, e poca scalabilità.

3. Vantaggi Routing

1. Efficienza e scalabilità

2. Permette un miglior runing dell' MTU per una maggiore efficienza.

4. Vantaggi Routing

1. I client devono usare un WINS server (come samba) per poter usare la condivisione di file su VPN.

2. Ogni instradamento deve essere configurato in modo da collegarsi alla subnet.

3. I software che dipendono dai broadcast non vedono le altre macchine in lan sulla VPN.

4. Funziona solo con IPv4, e IPv6 nel caso che i dispositivi che entrambi le parti lo supportino.

Installazione di Openvpn.

Per installare OpenVPN occorrono: libreria LZ0, pacchetto iproute2 moduli tap/tun compilati nel kernel, il vostro gestore di pacchetti attivo e funzionante e... se proprio siete dei geeks, il vostro gcc sempre al vostro fianco per la compilazione da sorgenti del tutto :D

Per quanto riguarda LZ0 ed iproute2 affidatevi al vostro gestore di pacchetti, che renderà le operazioni molto veloci. Come al solito riporto un breve elenco di come installare i pacchetti per la vostra distro preferita (sempre che sia nell'elenco :D)

Debian/ Ubuntu:

```
apt-get install lzo iproute2 / aptitude install lzo iproute2
```

Slackware:

```
slapt-get -install lzo iproute2 / swaret install lzo iproute2
```

Fedora:

```
yum -y install lzo iproute2
```

Mandriva:

```
urpml lzo iproute2
```

Freebsd:

```
pkg_add lzo iproute2 (sincronizzatevi col vostro mirror favorito prima)
```

Archlinux/Frugalware:

```
pacman -S lzo iproute2
```

Sabayon:

```
equo install lzo iproute2
```