

UMLNETSYM

Un sistema di simulazione di reti basato su User-mode Linux

Introduzione

User-mode Linux (<http://user-mode-linux.sourceforge.net/>) è il kernel di Linux compilato come un programma in user space, che non richiede privilegi particolari per essere eseguito, quindi può essere lanciato da qualunque utente.

Questo kernel ogni volta che deve accedere all'hardware per comunicare con la console, il disco o un'interfaccia di rete, richiama il kernel della macchina.

Ogni kernel UML quindi realizza una macchina virtuale con il minimo possibile di sovraccarico per il sistema.

Per funzionare richiede un file che, come un'immagine ISO, contenga l'immagine di un file system, nel quale sia installata una versione di Linux.

ULM può utilizzare una finestra per visualizzare la console con cui interagiranno con il sistema.

Per quanto riguarda la rete, potremo utilizzare un demone esterno (anch'esso un programma che potremo lanciare senza alcun privilegio particolare) che funzionerà come uno switch di rete per fare comunicare tra loro più macchine virtuali.

Questo strumento mi è sembrato l'ideale per creare uno strumento per simulazioni di reti.

Caratteristiche

Il sistema è costituito da:

- un'immagine del file system basata su Debian 6.0 Squeeze (sia in versione a 32 che a 64 bit) caricata con i software fondamentali per svolgere i vari compiti necessari in una

rete. Questa immagine verrà montata in readonly e quindi condivisa da tutte le macchine virtuali.

- un'immagine di un file system contenente le directory etc e var, copiate dall'immagine di cui sopra. Questo file, di dimensione di 6 MB, quindi molto modesta, verrà copiata per ogni macchina virtuale avviata. Verrà montata in read/write e potrà venire modificata durante la configurazione delle macchine virtuali
- uno script che interpreta la descrizione dell'esercizio e produce gli switch, le macchine virtuali, il grafo della rete con la finestra per visualizzarlo e la finestra per visualizzare la documentazione in PDF.

Per funzionare ovviamente richiede che sul sistema siano installati alcuni programmi:

- **user-mode-linux** per le macchine virtuali
- **graphviz** per generare il grafo della rete
- **unoconv** ed **OpenOffice** o **LibreOffice** per convertire la documentazione in PDF
- **zenity** per gestire l'interfaccia grafica dello script
- **imageMagick**, **gpicview** o **eog** di Gnome per visualizzare il file PDF
- **evince** di Gnome o **xpdf** per visualizzare il file PDF della documentazione

Per quanto riguarda l'immagine dei sistemi avviati, essa contiene tutti i programmi necessari per creare situazioni di rete anche estremamente complesse.

In particolare:

- strumenti di base **ifconfig**, **route**, **ping** e **traceroute**
- il comando **ip** per il rule based routing
- **iptables** per creare firewall
- i comandi per la configurazione di **IPv6**
- i comandi per gestire **vlan 802.1q**
- **ssh** server e client, con **sftp**
- un piccolo **server web** e **wget** per recuperare pagine
- il server **dns** **bind9**
- il server ed il client **dhcp**
- il server **ftp** **proftpd**
- il server **vpn** **openvpn**
- **tcpdump** per analizzare il traffico di rete
- alcuni **editor di testo**, oltre a **vi** (**mc** o **nano**)
- il server per i più comuni **protocolli di routing** (forniti da quagga)

Come si vede ce ne è d'avanzo non solo per una scuola superiore ma anche per corsi di networking specialistici.

Esecuzione di un esercizio

Un esercizio, come vedremo dopo, può essere distribuito come una serie di file in una directory oppure come un file compresso con estensione `.umlnz` contenente tutti i file necesari.

Per avviare una simulazione occorrerà invocare lo script `umlnet` da riga do comando o da interfaccia grafica, indicando l'esercizio da eseguire. Nel caso si utilizzi l'interfaccia grafica non è possibile invocare direttamente lo script `umlnet` da menù, in quando non c'è la possibilità di scegliere l'esercizio dall'interno dello script. Sarà piuttosto necessario selezionare l'esercizio e dal menù contestuale selezionare "apri con", quindi da li (magari scegliendo "altri programmi") selezionare lo script.

Se l'esercizio è in formato compresso, lo selezioneremo direttamente. Se invece è in una directory, dovremo cercare il file con estensione `.umlnet` e selezionare quello.

Una volta lanciato, lo script verifica la presenza dei programmi necessari per la sua esecuzione, quindi analizza l'esercizio e, se tutto va bene, inizia a predisporre l'esecuzione.

Per ogni macchina che fa parte dell'esercizio verranno create due finestre:

- un terminale con la console, che ci consentirà di lanciare comandi e fare verifiche
- una finestrella di controllo che ci consentirà, se necessario, di terminare l'esecuzione della macchina.

Inoltre verranno aperte due ulteriori finestre con:

- Il grafo della rete dedotto dalla descrizione fornita nel file `.umlnet`
- Il testo dell'esercizio allegato alla descrizione o incorporato in essa, tradotto in PDF

Verrà anche creata, nella home directory dell'utente, una directory `~/umlnet/` nella quale verrà creata una directory con il nome dell'esercizio (il nome del file `.umlnet` o `.umlnz` senza l'estensione). Questa directory verrà in seguito chiamata *directory dell'esercizio*.

Dentro quest'ultima directory verranno create le immagini dei dischi etc di tutte le macchine dell'esercizio, che rsteranno memorizzate in modo da poter riprendere un esercizio e ritrovare le configurazioni salvate.

Verrà poi creata una directory il cui nome sarà composto dalla data ed ora di lancio dell'esercizio. In questa directory compariranno una serie di sottodirectory, con il nome delle macchine create, trasformato in minuscolo.

Queste directory daranno montate nelle rispettive machine virtuali nella directory `/root` che è la home directory dell'utente `root` e consentiranno di salvare dei risultati di test o altre informazioni, da rivedere successivamente o da consegnare.

L'unico utente che può fare login nella macchina è `root`, l'amministratore, che non ha password

Al termine dell'esercizio la directory con la data ed ora dell'esecuzione corrente e le immagini dei dischi etc verranno salvati in un file `.tgz` il cui nome sarà composto dal nome dell'esercizio, dal nome del computer e dell'utente e dalla data ed ora di esecuzione.

Questo file potrà essere consegnato nel caso si debba correggere un esercizio assegnato.

Linguaggio di definizione della rete

Per produrre un esercizio si deve predisporre un file di descrizione della rete.

Questo file conterrà la definizione delle singole reti, rappresentate da switch virtuali e delle singole macchine connesse a questi switch tramite le relative schede di rete.

Alla fine il file potrà contenere un testo che descrive l'esercizio e/o il riferimento ad un documento esterno che lo descriva.

Le definizioni di switch e di macchine possono essere mischiate, ma se una macchina è collegata ad uno switch, esso deve essere già stato definito.

Le righe che non cominciano con una parola chiave vengono ignorate.

Le parole chiave devono iniziare all'inizio della riga e possono essere scritte sia in maiuscolo che in minuscolo.

Esistono quattro parole chiave:

- **switch** che definisce uno switch di rete, quindi una rete
- **macchina** che definisce un apparato di rete (router, server o terminale non fa differenza)
- **docfile** che indica il nome del documento con il testo dell'esercizio, se non si utilizza il testo incorporato
- **==testo==** che indica l'inizio del testo incorporato. Tutte le righe successive faranno parte del testo dell'esercizio e non verranno analizzate.

Dopo ogni parola chiave tranne `docfile` e `==testo==` troveremo una lista di parametri, ognuno con il suo nome, separati dal carattere `:`

Passiamo ora ad analizzare una per una le quattro parole chiave, con la loro sintassi.

switch

Definisce una rete virtuale, quindi uno switch virtuale al quale saranno collegate più macchine tramite le rispettive schede di rete. I possibili parametri sono:

- **nome** *obbligatorio*, deve essere una sola parola, indica il nome con cui ci si riferirà allo switch nelle macchine
- **descr** *opzionale*, se manca viene preso il nome, serve per un nome descrittivo per lo switch nello schema della rete
- **indirizzo** *opzionale*, ha solo scopo di documentazione e compare nello schema della rete. Serve per indicare indirizzo e maschera della rete.

Esempi:

- switch nome=SW1:descr=Switch 1
- Switch nome=SW2:descr=Switch 2:indirizzo=192.168.33.0/24

macchina

Definisce una macchina virtuale, che possiederà le proprie interfacce di rete connesse agli switch, la propria directory di configurazione indipendente ed il proprio sistema operativo. I possibili parametri sono:

- **nome** *obbligatorio*, deve essere una sola parola, indica il nome con cui verranno nominati i dischi "etc" con le configurazioni e le directory contenenti i dati esportati dalle macchine
- **descr** *opzionale*, se manca viene preso il nome, serve per un nome descrittivo per la macchina nello schema della rete
- **rete** *obbligatoria*, contiene un elenco, separato da , (virgola) di schede di rete con i loro attributi.

La definizione di una rete ha la seguente forma:

```
eth<numero>-><switch> [;<indirizzo>/<bit_maschera>|dhcp]
```

dove:

- **eth<numero>**, *obbligatorio*, con <numero> che partono da 0 è il nome dell'interfaccia, come la troveremo dentro alla macchina virtuale
- **<switch>**, *obbligatorio*, è il nome dello switch cui la scheda è connessa. Deve già essere stato definito
- **<indirizzo>/<bit_maschera>**, *opzionale*, è l'indirizzo con cui verrà configurata la scheda di rete. Indirizzo deve essere un indirizzo **IPv4** valido, mentre <bit_maschera> indica il numero di bit della netmask (ad esempio 192.168.9.3/24), oppure la parola chiave **dhcp**. Se non viene indicato un indirizzo, la scheda viene lasciata da configurare a mano o tramite il file `/etc/network/interfaces`, come parte dell'esercizio. Per ora non sono gestite le configurazioni delle sottointerface o delle *vlan* da riga di comando. Non è ancora gestita la configurazione di **IPv6**

Esempi:

- **rete=eth0->SW1,eth1->SW2**

- `rete=eth0->SW1;80.90.112.1/8`
 - `rete=eth0->SW1;dhcp`
 - `mem=<quantità_di_memoria>`, *opzionale*, dove `<quantità_di_memoria>` è un numero seguito da una M, per indicare una quantità di memoria in MegaByte. Consente di allocare alla macchina virtuale una quantità di memoria diversa dal default di 96M
 - `gateway=<indirizzo>`, *opzionale*, indica il default gateway della macchina; `<indirizzo>` deve essere l'indirizzo **IPv4** del default gateway
 - `route`, *opzionale*, consente di definire una serie di rotte statiche, separate da , (virgola). Ogni rotta ha la seguente forma:
`<ip_rete>/<bit_maschera>;<ip_gateway>`
- Esempi:
- `route=192.168.33.0/24;80.90.112.96`
 - `forward`, *opzionale*, abilita il forwarding
 - `nameserver=<ip_dns1>[,<ip_dns2>[,<ip_dns3>]]`, *opzionale*, consente di definire fino a tre indirizzi di DNS server

docfile

Indica il nome del file con il testo dell'esercizio. Il testo potrà essere scritto in uno qualunque dei formati interpretati da OpenOffice, oppure in PDF.

Il file indicato verrà cercato nella directory dove risiede la definizione della rete.

Se il file è in formato PDF, verrà visualizzato direttamente, altrimenti verrà convertito in PDF e verrà visualizzato il file PDF risultante.

E' quindi sempre consigliabile inserire direttamente un file PDF nell'esercizio.

Esempio:

- `docfile=network.pdf`

==testo==

Questo, più che un comando è un separatore. Se manca o è illeggibile il file di documentazione che contiene il testo dell'esercizio indicato con `docfile`, viene trasformato in PDF il testo che segue il separatore `==testo==` e viene visualizzato quello.

Questo separatore va pensato come una descrizione interna del file e come una misura di sicurezza nel caso in cui il file di documentazione non sia raggiungibile o sia rovinato.

Preparazione di un esercizio

Un esercizio è composto da tre tipi di file

- la descrizione della rete, scritta nel linguaggio appena descritto, registrata in un file con estensione `.umlnet`
- il testo dell'esercizio, inserito in un documento di qualunque tipo che possa essere aperto da OpenOffice, oppure direttamente in PDF. Questo testo dovrà essere indicato nella descrizione della rete tramite il comando `docfile=<nome_doc_file>`
- un'eventuale serie di immagini etc per le macchine.

Tutti questi file devono essere nella stessa directory.

Se per una macchina virtuale non viene fornita una immagine etc configurata, verrà copiata quella di default di `umlnet`.

Le immagini etc potranno venir fornite per avere delle macchine con le schede di rete o dei servizi preconfigurati, per fare esercizi di test o di configurazione del client.

L'immagine etc per una macchina virtuale viene riconosciuta perché ha il nome `etc.<nome_macchina>` dove `<nome_macchina>` è il nome della macchina come definito nel file di descrizione della rete, trascritto in lettere minuscole.

Il modo più semplice per ottenere un'immagine per la nostra simulazione è lanciare la simulazione stessa priva dell'immagine etc personalizzata, configurare le macchine che ci interessano e poi copiare nel testo dell'esercizio il file etc delle macchine configurate dalla *directory dell'esercizio*.

Una volta preparato l'esercizio nella sua directory, potremo trasformarlo in un esercizio compresso, più facilmente distribuibile.

Questo file dovrà avere l'estensione `.umlnz` e non dovrà contenere sottodirectory.

Il file di definizione della rete dovrà obbligatoriamente, in questo caso, avere lo stesso nome dell'archivio, ma con l'estensione `.umlnet`

Se ad esempio il nostro esercizio è definito con il file `network.umlnet` ed il file di documentazione si chiama `network.pdf`, per produrre l'archivio dell'esercizio dovremo, in un terminale, spostarci nella directory dove sono entrambi i nostri file e digitare il comando:

```
tar -cvzf network.umlnz network.umlnet network.pdf
```

È possibile creare un esercizio compresso con lo script `umlnetsym-mkexercise` sia invocandolo da riga di comando che da interfaccia grafica, indicando il file `.umlnet` dell'esercizio sulla riga di comando oppure cliccando con il tasto destro sul file dell'esercizio e scegliendo "Apri con".

Personalizzazione dell'immagine di boot

Crazione di una nuova immagine

Esempi

Questo esercizio crea una rete con quattro macchine:

- una che rappresenta Internet
- una che funge da router
- due che fungono da client

e due switch

- uno che connette Internet alla scheda esterna del router
- uno che connette la scheda interna del router e i due client

La macchina Internet è configurata da riga di comando ma per nessuna è previsto un file etc con la configurazione

```
# Script di descrizione della rete
# Descrivo gli switch - il nome deve iniziare per SW
switch nome=SW1:descr=Switch 1
Switch nome=SW2:descr=Switch 2:indirizzo=192.168.33.0/24
; switch nome=SW1:descr=Switch 1
# Descrivo le macchine - il nome deve iniziare per M
macchina
nome=M1:descr=Internet:rete=eth0->SW1;80.90.112.1/8:route=19
2.168.33.0/24;80.90.112.96
; macchina nome=M2:descr=Router:rete=eth0->SW1,eth0->SW2
; macchina nome=M2:descr=Router:rete=eth0->SW1,eth1->SW3
```



```
; macchina nome=M2:descr=Router:rete=eth0->SW1,eth1->SW2
macchina
nome=M2:descr=Router:rete=eth0->SW1;80.90.112.96/8,eth1->SW2
:gateway=80.90.112.1

macchina nome=M3:descr=Client 1:rete=eth0->SW2:mem=128M

macchina nome=M4:descr=Client 2:rete=eth0->SW2

; macchina nome=M4:descr=Client 2:rete=eth0->SW2

# Indico il testo, o lo incorporo

docfile=network.pdf

==testo==

RETE CON ROUTER

La macchina "Internet" è già configurata con l'indirizzo
80.90.112.1/8

Configurare l'interfaccia eth0 della macchina "Router" con
l'indirizzo 80.90.112.96/8

Configurare l'interfaccia eth1 della macchina "Router" con
l'indirizzo 192.168.33.1/24

Configurare la macchina "Client1" con l'indirizzo
192.168.33.10/24

Configurare la macchina "Client2" con l'indirizzo
192.168.33.20/24

Configurare tutte le rotte statiche necessarie perché ogni
macchina possa comunicare con qualunque altra
```

Verificare la connettività con i comandi ping e traceroute

Salvare i risultati dei comandi (tramite la redirectione)
nella directory /host di ogni macchina



Quest'opera è distribuita con Licenza [Creative Commons Attribuzione - Non commerciale 3.0 Unported](https://creativecommons.org/licenses/by-nc/3.0/).