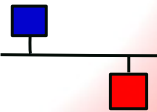


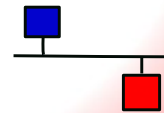
Emulatore di reti - 2

Primi Esercizi



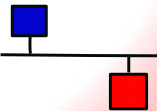
UML

- User Mode Linux (<http://user-mode-linux.sourceforge.net/>) è il porting del kernel di Linux per utilizzare i dispositivi di Linux come driver.
- In questo modo è possibile **lanciarlo come programma utente**, e quindi avere un **sistema Linux che gira come un processo** dentro ad un altro sistema Linux
- E' una forma molto economica per realizzare una **macchina virtuale Linux**
- Può montare più **file che contengano un filesystem** e partire da uno di questi
- E' anche in grado di montare una **directory della macchina ospite** come fosse un disco.
- Per finire, è possibile creare **schede di rete**.
- Queste schede possono:
 - essere **connesse alla rete reale** tramite l'interfaccia **TUN/TAP**
 - essere **connesse ad un processo che funge da switch** virtuale



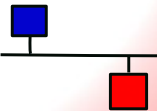
UmlNetSym

- A partire da UML viene creato **UmlNetSym** (<http://sourceforge.net/projects/umlnetsym/>) che è un sistema per creare esercitazioni di reti tramite macchine virtuali UML
- Un esercizio prevede la creazione di **un certo numero** di macchine virtuali
- Ogni macchina virtuale avrà a disposizione **96 MiB di RAM**
- Queste macchine virtuali saranno connesse da una **serie di reti (switch) virtuali**
- Ogni esercizio conterrà anche lo **schema della rete** ed il **testo dell'esercizio**



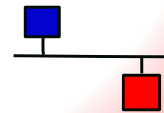
UmlNetSym - Dischi

- Ogni macchina virtuale possiede tre dischi virutali:
 - 1)/ = Sistema operativo Devuan 2 condiviso in sola lettura
 - 2)/etc/ = un disco di configurazione per ogni macchina in lettura/scrittura
 - 3)/root/ = una directory del sistema, diversa per ogni machina in lettura/scrittura
- Il sistema viene **avviato dal disco 1**, montato in **sola lettura**.
- Per le **prime fasi** dell'avvio viene utilizzata la directory **/etc/** di **questo disco**, poi viene **montato il disco /etc/ (2)** della singola macchina



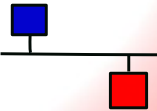
UmlNetSym - Disco di avvio

- Il **disco di avvio** è montato in sola lettura ed è **condiviso** da tutte le macchine virtuali
- In questo disco sono installati tutti i pacchetti per:
 - Configurazione della rete in IPv4
 - Configurazione della rete in IPv6
 - Diagnostica di rete
 - Server e client DHCP
 - Server DNS
 - Server e client SSH
 - Server e client FTP
 - Server e client HTTP
 - Firewall IPTables
 - Vpn con OpenVPN
- E molto altro...



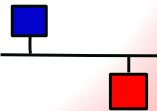
UmlNetSym - Disco /etc/

- Tutta la **configurazione** di un sistema Linux è contenuta nella directory **/etc** e nelle sue sottodirectory
- **All'avvio** i file di configurazione vengono presi dalla directory **/etc** contenuta nel disco di avvio
- **Appena vengono montati i dischi supplementari** la directory **/etc** della macchina virtuale prende il controllo.
- **Se non esiste** già un **disco /etc** creato da una precedente esecuzione, ne viene **copiato** uno di **default**.
- Un **esercizio** può contenere anche dei **dischi /etc preconfigurati** per alcune sue macchine
- I **dischi /etc** vengono **conservati** da un'esecuzione all'altra (a meno che non chiediamo di cancellarli e ricostruirli) e quindi **manterranno le nostre configurazioni**.
- Ogni **disco /etc** occupa meno **8 MiB**.



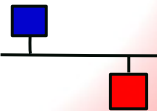
UmlNetSym - Disco /root

- La **directory /root** è la **home directory** dell'utente root, **amministratore** del sistema
- La **directory /root** di ogni macchina virtuale è **mappata su di una directory del sistema ospite**
- In questa directory potranno essere **salvati dei file nella macchina virtuale**, che potranno essere poi recuperati una volta spente le macchine virtuali
- È anche possibile **passare file alla macchina virtuale**, copiandoli in questa directory.
- Questa directory è utile per **salvare risultati di test** sulla rete e sui servizi, da esaminare poi, magari per **correggere un esercizio**.



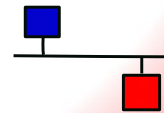
UmlNetSym - Rete

- Ogni macchina può avere **più schede di rete**
- Possono essere creati **tanti switch virtuali** separati quante sono le **reti** del nostro esercizio
- Ogni esercizio, oltre alle **macchine virtuali** necessarie, definirà anche gli **switch necessari**
- Nell'**esercizio** è definito **quale scheda di rete** di ogni macchina **si connette a quale switch**
- Nell'esercizio, anche in assenza di un disco /etc preconfigurato, è possibile **preconfigurare** alcune o tutte le **schede di rete**.



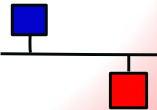
UmlNetSym - La directory dell'esercizio

- **Durante** l'esecuzione dell'esercizio viene creata la directory `/tmp/network.<codice_univoco>` nella quale viene **scompattato l'esercizio** e vengono creati i **file e le directory di lavoro**
- In particolare, nella **sottodirectory machines** viene creata una sottodirectory contenente una sottodirectory per ogni macchina, che verrà montata come **'file system utente'** della macchina virtuale
- Questo **'file system utente'** conterrà quattro sottodirectory:
 - **etc** – che conterrà la **copia di lavoro**, eventualmente personalizzata, usata dalla macchina e modificabile, della directory `/etc`
 - **modules** – sarà collegata alla **directory modules del kernel di usermodelinux** installato e sarà visibile in `/lib/modules/`
 - **root** – conterrà la directory `/root` – la home directory dell'utente root
 - **user** – conterrà la directory `/home/user` – la home directory dell'utente user



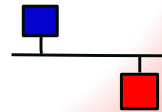
UmlNetSym - La directory umlnetsym

- Nella home dell'utente viene poi creata la directory `~/umlnetsym`
- Questa directory contiene:
 - Il file **umlnetsym.err** che contiene gli **errori** (stderr) dell'eseguibile **umlnetsym**, resettato ad ogni lancio
 - Il file **umlnetsym.log** che contiene i **messaggi** (stdout) dell'eseguibile **umlnetsym**, resettato ad ogni lancio
 - Una directory per ogni esercizio svolto con all'interno
 - Un **archivio tgz** contenente la versione compressa della **directory machines** della directory temporanea, con il nome **<esercizio>-<utente>-<data>-<ora>.tgz**, da poter consegnare, per ciascuna volta che l'esercizio è stato eseguito.
- I **file di salvataggio degli esercizi**, oltre ad essere utili per la **correzione** degli esercizi, saranno anche necessari per creare le **directory etc personalizzate** per gli esercizi



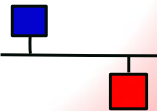
UmlNetSym – Esempio di File di Salvataggio

- Ad esempio, l'**esercizio 1** completo, eseguito alle 9:19 del 20/1/2014 produrrà il file **esercizio1-salvi-20140120-0919.tgz**
- Il contenuto di questo archivio sarà:
 - **./** (dir) il contenitore
 - **m1** (dir) directory della macchina **m1**
 - **etc** (dir - 1,9MiB) – la **etc** di m1, con le modifiche fatte dall'utente
 - **modules** (dir – vuota) – il punto di mount della directory dei moduli del kernel
 - **root** (dir - vuota) – la **home** di **root** di **m1**
 - **user** (dir – vuota) – la **home** dell'utente **user**
 - **m2** (dir) directory /root della macchina m2
 - **etc** (dir - 1,9MiB) – la **etc** di m1, con le modifiche fatte dall'utente
 - **modules** (dir – vuota) – il punto di mount della directory dei moduli del kernel
 - **root** (dir – 2,9 KiB) – la **home** di **root** di **m2**
 - **.bash_history** (file) l'elenco dei comandi digitati, salvato da bash
 - **host.txt** (file) risultato del comando **host server.testnet.org > host.txt**
 - **hostr.txt** (file) risultato del comando **host ip_della_nostra_macchina > hostr.txt**
 - **ifconfig.txt** (file) risultato del comando **ifconfig > ifconfig.txt**
 - **ping.txt** (file) risultato del comando **ping -c 2 192.168.4.1 > ping.txt**
 - **route.txt** (file) risultato del comando **route -n > route.txt**
 - **tcpdump.txt** (file) risultato del comando **tcpdump -c 10 > tcpdump.txt & ping -c 10 192.168.4.1**
 - **user** (dir – vuota) – la **home** dell'utente **user**



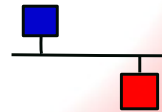
Esercizio 1 - salvataggio risultati

- L'esercizio si può scaricare da http://www.salvi.mn.it/stefano/corsi/umlnetsym/umlnetsym_1/esercizio1.umlhz
- Scopo dell'esercizio è prendere confidenza con i metodi per **salvare i risultati di un esercizio**
- Vengono create due macchine virtuali, **Server** e **Client**, connesse da una rete
- Il **Server** contiene già un **server DHCP** ed un **DNS configurati**
- Il **Client** è impostato come client **DHCP**, quindi è automaticamente configurato
- Dovremo loggarci nel **Client** con utente **root**, senza password e dare i comandi di diagnostica di rete indicati nel **testo dell'esercizio**.
- Potremo poi verificare il risultato nei file lasciati nella directory dell'esercizio
`~/umlnetsym/esercizio1/esercizio1-<utente>-<data>-<ora>.tar.bz2`



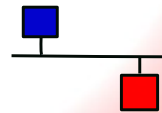
Esercizio 2 - Configurazione statica manuale

- L'esercizio può essere scaricato da http://www.salvi.mn.it/stefano/corsi/umlnetsym/umlnetsym_1/esercizio2.umlنز
- Scopo dell'esercizio è quello di **configurare manualmente** una macchina Linux.
- Dovremo impostare l'**IP e la maschera** di rete per l'interfaccia **eth0**
- Dovremo impostare il **default gateway**
- Dovremo configurare la **risoluzione dei nomi tramite DNS**
- Dovremo impostare l' **hostname**
- Utilizzeremo i comandi **ip** ed **hostname**
- Modificheremo il file **/etc/resolv.conf**
- Utilizzeremo il comando **man** per conoscere i parametri da utilizzare e un editor tra **vi**, **ne**, **nano** o **mc**



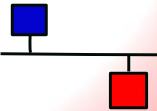
Esercizio 3 - Salvataggio configurazione

- L'esercizio si può scaricare da http://www.salvi.mn.it/stefano/corsi/umlnetsym/umlnetsym_1/esercizio3.umlنز
- Scopo dell'esercizio è di creare una **configurazione permanente** della macchina, che venga ricaricata al riavvio.
- In questo caso configureremo **IP, maschera, default gateway e DNS**.
- **Non modificheremo il nome host**, in quanto questo viene **sovrascritto** all'avvio della macchina con quello configurato nell'esercizio.
- Dovremo questa volta modificare **/etc/resolv.conf**, come nell'esercizio precedente, ed il file **/etc/network/interfaces**



Esercizio 4 - tabella di routing nel client

- L'esercizio si può scaricare da http://www.salvi.mn.it/stefano/corsi/umlInetsym/umlInetsym_1/esercizio4.umlInz
- In questo esercizio abbiamo una rete più complessa, con **due router**.
- Dovremo aggiungere una **rotta statica** per la seconda rete nella nostra tabella di routing.
- Per prima cosa ripeteremo la **configurazione della rete permanente**, come nell'esercizio precedente
- Poi potremo aggiungere la rotta **a mano** utilizzando il comando **ip route**
- Per finire **aggiungeremo** la rotta statica nel file **/etc/network/interfaces** aggiungendo, tra i parametri dell'**interfaccia eth0**, una riga contenente **up** seguito dal comando **ip route** (utilizzando il percorso completo **/sbin/ip**) precedentemente dato a mano.



Esercizio 5 - Configurazione del router

- L'esercizio si può scaricare da http://www.salvi.mn.it/stefano/corsi/umlnetsym/umlnetsym_1/esercizio5.umlنز
- In questo caso configureremo il **routing** (o meglio il **server** per al sua funzione di router)
- Come prima faremo da **riga di comando** la configurazione delle **interfacce** (questa volta due e non una) e della **rotta statica** per il **client-interno**
- Dal **client-esterno** controlliamo la **connettività** verso le interfacce del server e verso le macchine interne.
- Scopriamo che tutte le **interfacce** del server sono **raggiungibili**, ma le **reti interne no**. Questo deriva dal fatto che di default **il server non instrada pacchetti**.
- Per **abilitare l'instradamento (forwarding)** dei pacchetti dovremo modificare un parametro del kernel. Possiamo farlo **scrivendo 1** nel file virtuale **/proc/sys/net/ipv4/ip_forward** ad esempio con il comando **echo**
- Per finire renderemo questa configurazione **permanente** modificando i **file** come nell'esercizio precedente

