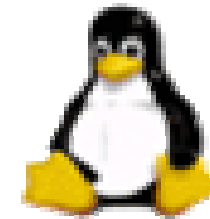




Linux & Netkit

Andrea Fava



Dipartimento di informatica

Tecnologie di Convergenza su IP

A.A 2006-2007

Introduzione

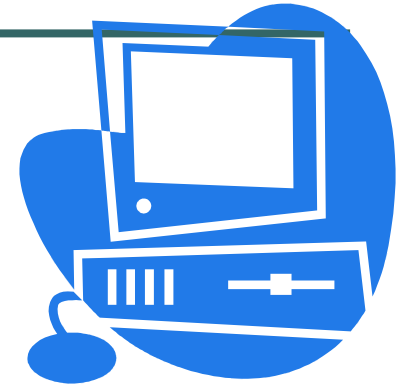
- Nel corso degli anni l'informatica ha cercato di spingere molto sul concetto di "virtualizzazione".
- Trasformare un singolo computer in una varietà di computer utilizzabili contemporaneamente e dotati di sistemi operativi.

Introduzione

- Un sogno impossibile ? No! Esattamente quello che oggi permettono i software di virtualizzazione.
- I primi esperimenti di “virtualizzazione” risalgono al 1960 presso i laboratori IBM
- i software presenti sul mercato sono in grado di emulare una o più macchine virtuali usando cpu, memoria, dvd, hard disk sulla macchina reale.

Introduzione

A cosa serve una macchina virtuale?



- Provare software rischioso
- Installare applicazioni senza mettere a rischio la stabilità del sistema
- Fare esperimenti con ambienti meno conosciuti come ad esempio:
- Linux, Solaris, FreeBSD, Netbsd, Openbsd, OpenDarwin, DragonFlyBSD.

Introduzione

Vi sono molti software per windows
distinguiamo i più celebri:

- VMware workstation (commerciale)

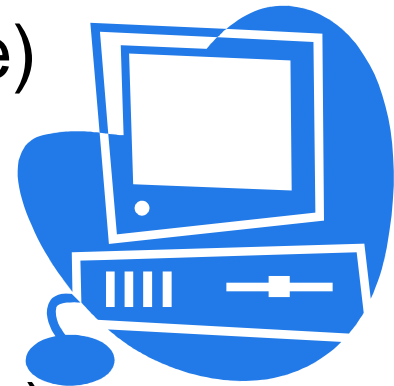
www.vmware.com

- Virtual pc 2007 (gratuito)

www.microsoft.com/italy

- Parallels workstation (commerciale)

www.parallels.com



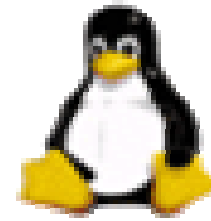
Introduzione

software per Linux (i più celebri) :

- Vmware workstation (commerciale)
- Qemu (open source)
- Bochs (open source)
- Xen (open source)
- Plex86 (open source)
- User-Mode-Linux (open source)
- UMLMON (open source)
- Vbet
- Vini (<http://vini-veritas.net/>).

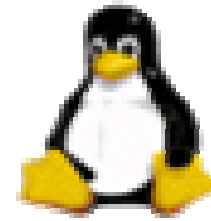


User Mode Linux (U.M.L)



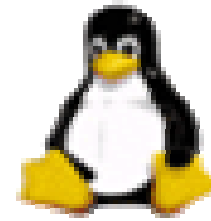
- Implementazione open source per macchine virtuali.
- A differenza di altri (vmware e plex86) è un porting di linux su se stesso.
- UML Permette di eseguire un kernel linux all'interno di un sistema operativo linux es. testare un nuovo kernel.
- Si posiziona come un normale processo utente.
- UML è nato principalmente per kernel hacking e per insegnamento didattico.

User Mode Linux (U.M.L)



- Possiamo definire U.M.L come un insieme di patch ad un kernel linux
- UML offre la possibilità di eseguire una mandrake sotto debian
- U.M.L Fornisce le anche stesse funzioni di network come una comune macchina linux
- Si possono realizzare collegamenti di rete tra le varie virtual machine attive

User Mode Linux (U.M.L)



Come funziona UML ?

- Ogni processo U.M.L corrisponde un processo sulla macchina ospitante.
- Esiste un thread speciale che fa il tracing delle chiamate ai processi U.M.L.
- Il tracing thread annulla le chiamate di sistema e reindirizza nel kernel U.M.L.

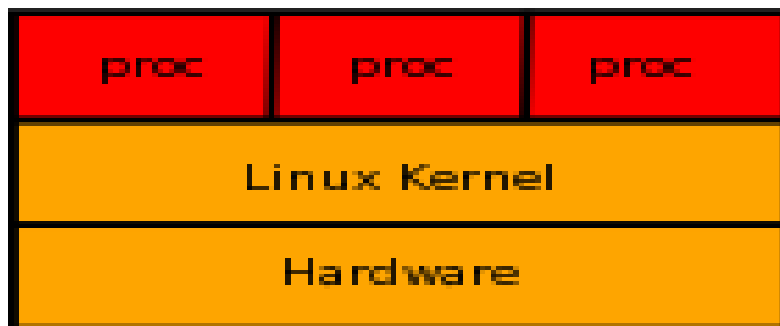
User Mode Linux (U.M.L)



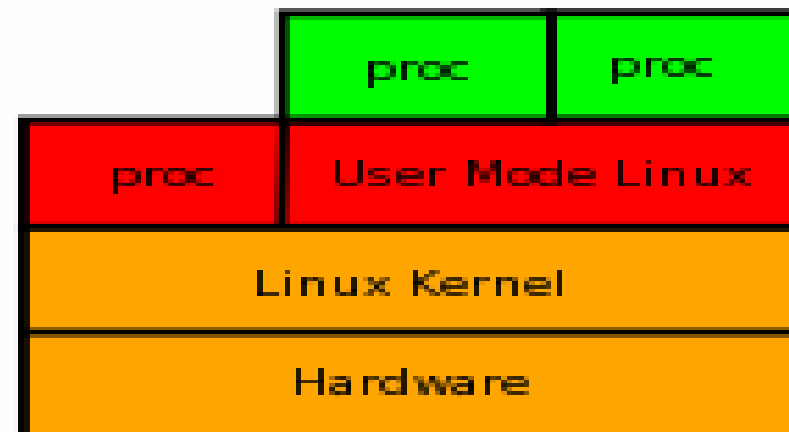
- U.M.L comunica con vero kernel

Modalita' di funzionamento

❖ id: home-works_xm1_v 1.1.2 004/11/26 22:33:39 donos. Exp ❖

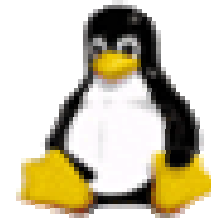


senza UML



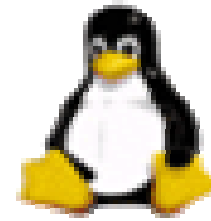
con UML

User Mode Linux (U.M.L)



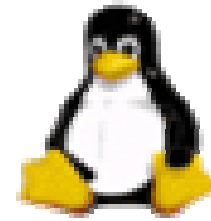
- Il kernel U.M.L risiede nello spazio di ognuno dei suoi processi , quindi scrivibile e soggetto al rischio sicurezza.
- Esiste la possibilità di coprire il rischio sicurezza con una modalità più sicura detta: U.M.L Jail , ma la soluzione è estremamente lenta.
- sono apparse le prime patch in merito : S.K.A.S (separate kernel address space) per migliorare la velocità d'esecuzione.

User Mode Linux (U.M.L)



- La compilazione di un UML uguale alla compilazione del kernel Linux.
- Scaricare i sorgenti del kernel desiderato da <http://www.kernel.org>.
- Scaricare la patch UML appropriata per il kernel da compilare da <http://user-mode-linux.sourceforge.net>
- Creare una directory per la compilazione, decomprimere i sorgenti del kernel e applicare la patch UML.
- Eseguire il comando `# make menuconfig ARCH=um` e `make linux ARCH=um`
- installare sul sistema le utilità UML, sempre scaricabili da <http://user-mode-linux.sourceforge.net>

User Mode Linux (U.M.L)

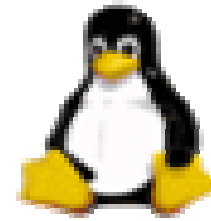



- Oltre alla soddisfazione di compilare un kernel U.M.L , c'è la possibilità di prelevare vari kernel di ridotte dimensioni già pronti all'uso
- È possibile anche scaricare una sorta di starter kit in formato rpm con incluse le utility necessarie.
- scaricabile da :

<http://user-mode-linux.sourceforge.net>

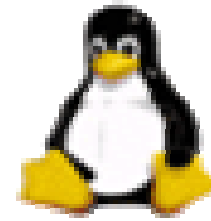


User Mode Linux (U.M.L)



- E' errore comune avviare un processo uml-linux con kernel 2.4 all'interno di una distribuzione con kernel 2.6. (system call) 
- Mentre è semplice recuperare dalla rete i numerosi kernel uml pronti, la faccenda si complica per quanto riguarda i root filesystem.
- Le dimensioni di questi filesystem già pronti posso arrivare a qualche centinaio di mega.
- Generalmente una volta espansi questi filesystem hanno poco spazio disponibile per sperimentare .
- L'utente più esperto può creare un proprio filesystem e decidere quanto spazio usare.

User Mode Linux (U.M.L)

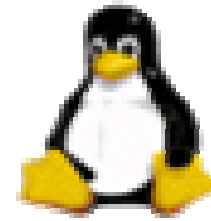


U.M.L in cinque mosse:

- Download u.m.l e filesystem
- Copia del root filesystem
- Installazione
- Sistemazione del link `root_fs`
- Lancio della virtuale machine



User Mode Linux (U.M.L)



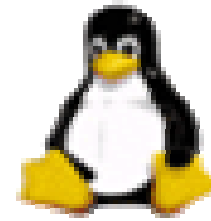
- 3 Installazione :
- `rpm -ivh usermodelinux-2. 6.19.i386.rpm`

- 4 sistemazione link
- `bunzip2 Debian-3.0r0.ext2.bz2`
- `ln -s fs/Debian-3.0r0.ext2 root_fs`

- 5 Lancio della vm
- `linux ubd0=root_fs`

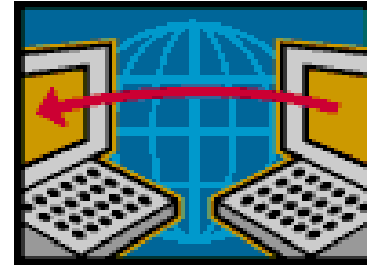


User Mode Linux (U.M.L)



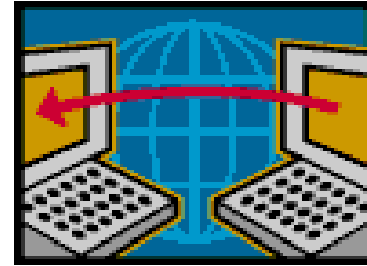
- Ricordiamo alcuni termini e convezioni che useremo nel nostra esplorazione
- Vm (indichiamo una macchina virtuale)
- Host(indichiamo la macchina reale che ospita le vm nella sua relativa ram
- Useremo `vm>` e `host>` per indicare i vari casi di inserimento comandi

La rete virtuale U.M.L



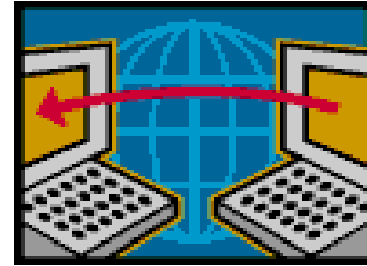
- Per poter usare una vm con interfaccia di rete dobbiamo passare il parametro `eth <n>` nel comando avvio del vm stessa (cioè passati al kernel uml).
- U.M.L mette a disposizione 5 parametri

La rete virtuale U.M.L



- 1) Ethertap tun/tap : connettività direzionale tra vm e host
- 2) switch-daemon, multicast : solo con le altre vm
- 3) slip, slirp : se non è disponibile ethertap , tun/tap
- 4) pcap : interfaccia di rete in modalità read-only usa la libreria libcap e risulta utile per lo sniffer dentro una vm

La rete virtuale U.M.L

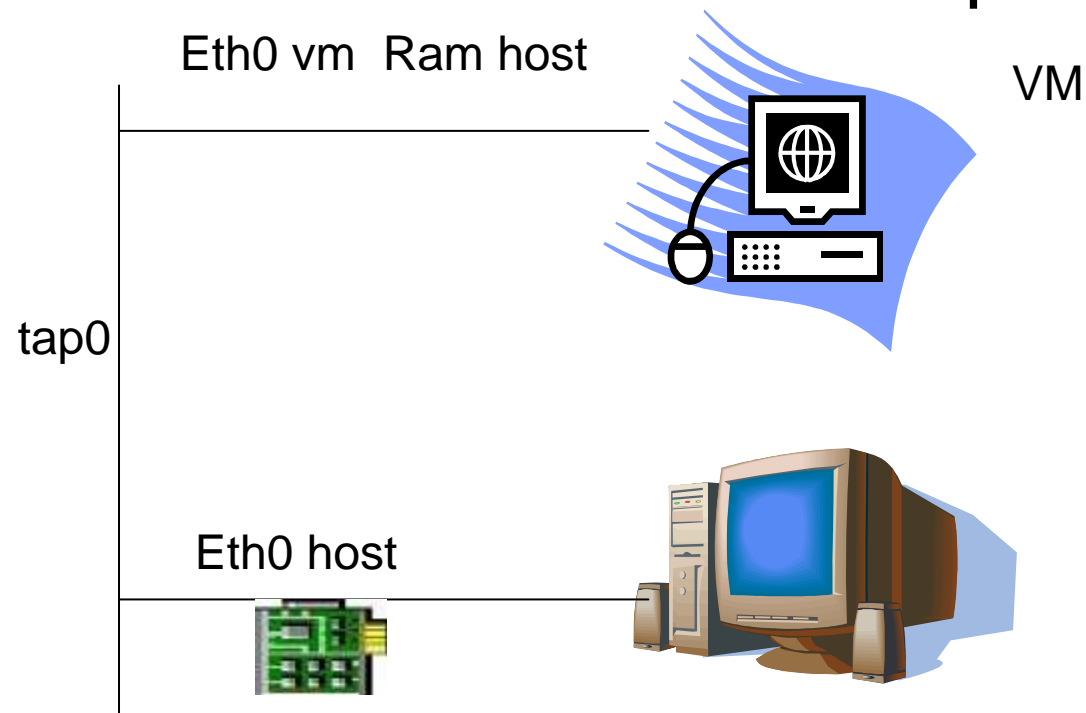


- Il modo più semplice per impostare un networking di base è l'uso del modo ethertap.
- Per assegnare un'interfaccia ethernet ad una vm dobbiamo indicare il device interessato e il traspont scelto tramite il comando `eth <n>`

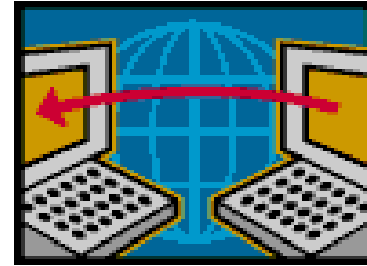
La rete virtuale U.M.L



- Schema di funzionamento ethertap

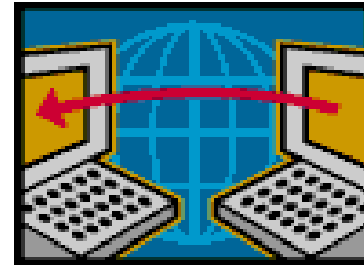


La rete virtuale U.M.L



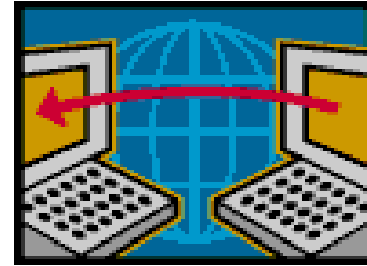
- Il formato generale per ethertap è il seguente :
- `eth<n> = ethertap,<device>,<indirizzo>,<ip>`
- Se un host ha eth0 con ip 192.168.0.1
- `# linux eth0=ethertap,tap0,192.168.0.1`

La rete virtuale U.M.L



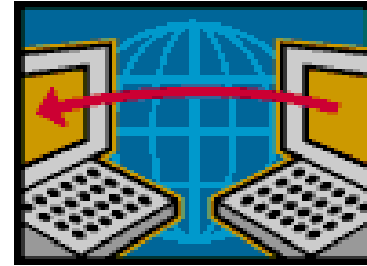
- Eseguiamo sulla vm i seguenti comandi:
- Vm> ifconfig eth0 192.168.0.144 up
- Vm> route add default gw 192.168.0.1
- Vm> ping 192.168.0.1
- Infine proviamo ad accedere alla nostra virtual machine da host
- Host>ping 192.168.0.144

La rete virtuale U.M.L



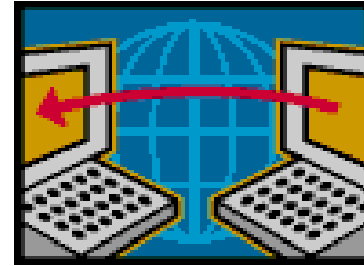
- Complichiamo leggermente la nostra prima rete virtuale U.M.L parlando di Switch.
- Switch è un dispositivo di rete che inoltra selettivamente i frame ricevuti verso una porta di uscita.
- Ricorriamo alle `uml.utililes` e al tool `uml_switch`: in realtà è un demone in grado di simulare uno switch ethernet
- La sua sintassi è :
- `Uml_switch (opzioni)`

La rete virtuale U.M.L



- Le opzioni possibili sono tre
- hub (forza lo switch ad essere un hub)
- Tap (serve a connettere la rete delle virtual machine con l'host e quindi con l'esterno)
- Unix (indica il tipo di socket solitamente unix)

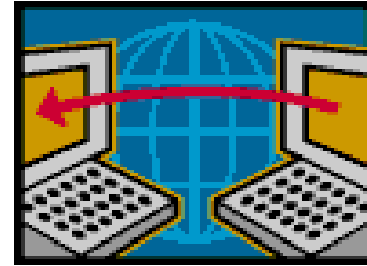
La rete virtuale U.M.L



- Host> `uml_switch -unix /tmp/switch`
- L'unica avvertenza da tenere a mente è che lo switch deve essere creato prima di collegarci le virtual machine.
- Avendo cambiato il trasporto da ethertap a switch ovviamente la sintassi sarà diversa del parametro eth

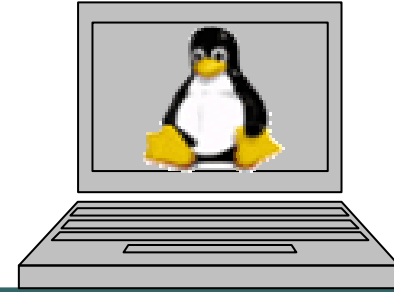
```
switch-1  
armin@partner:~$ uml_switch -unix /tmp/switch-1  
uml_switch attached to unix socket '/tmp/switch-1'  
New connection  
Addr: fe:fd:c0:a8:01:01 New port 5
```

La rete virtuale U.M.L



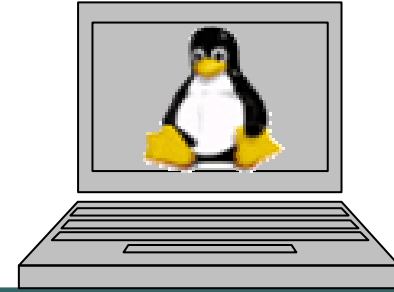
- `eth<n>=daemon ,<eth address>,<tipo socket>,<nome socket>`
- Tipo e nome socket hanno gli stessi valori dello switch virtuale.
- Il nostro studio con U.M.L si conclude qua, sono molte le opzioni possibili per creare una lan virtuale con uml ,tra cui aggiunta di router, domini di collisione ,proxy arp ecc

Il progetto NetKit



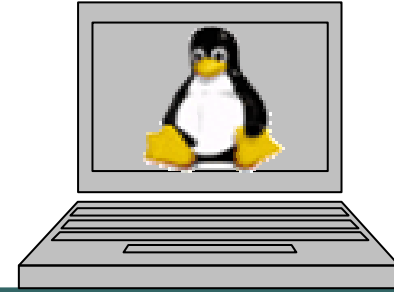
- Il progetto Netkit nasce in Italia e più esattamente all'università Roma tre e mantenuto dal L.U.G (linux user group) di Roma.
- Netkit è un laboratorio virtuale con le funzionalità di UML, possiamo definirlo come un front-end a UML.
- E' un insieme predefinito di comandi per le macchine virtuali.
- Possiede un filesystem preinstallato.

Il progetto NetKit



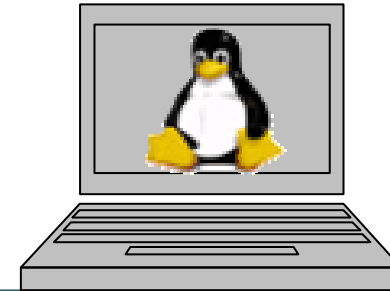
- Il progetto è reperibile all'indirizzo
- www.netkit.org
- E' presente molto materiale in linea disponibile per l'utente.
- Disponibile sia per Linux e per Windows.
- C'è una distribuzione Live CD con incluso il netkit (netkit4tic) .
- Netkit4tic E' una versione di Knoppix elaborata per ottenere migliori prestazioni con UML

Il progetto NetKit



- Per chi non vuole installare linux sulla propria macchina può eseguire in emulazione il WinNetkit (reperibile sul sito di netkit) in ambiente windows tramite VMWare Player:
- www.vmware.com/download/player

Il progetto NetKit



Il winNetkit contiene il programma già preconfigurato.

The screenshot shows the WinNetkit application window. At the top, there are icons for Firefox Web Browser, Shell, and File Manager. The main area contains several terminal windows. The top window shows the root user at an Ubuntu machine, running commands to start two virtual machines, 'pc1' and 'pc2'. Below this, there are two separate terminal windows, one for 'pc1' and one for 'pc2'. Both show the initialization process of the virtual machines, including setting up networking, initializing random number generators, and starting the Netkit phase 1 and 2 scripts. The 'pc1' terminal shows the user logging in as 'root' and the system date as 'Tue Sep 13 18:38:01 CEST 2005'. The 'pc2' terminal shows a similar process for the second virtual machine.

```
root@ubuntu: /mnt/netkit/netkit2/bin ...
Model fs: /mnt/netkit/netkit2/fs/netkit-fs
Filesystem: /mnt/netkit/netkit2/bin/pc1.disk
Hostfs at: /root

Running ==> /mnt/netkit/netkit2/kernel/netkit-kernel modules=/mnt/netkit/netkit2
/kernel/modules name=pc1 title=pc1 umid=pc1 mem=12M ubd0=/mnt/netkit/netkit2/bin
/pc1.disk /mnt/netkit/netkit2/fs/netkit-fs root=98:1 uml_dir=/root/.netkit/mcone
ole hosthome=/root quiet con0=term con1=null
root@ubuntu:/mnt/netkit/netkit2/bin# vstart pc2 --new

===== Starting virtual machine "pc2" =====
Kernel: /mnt/netkit/netkit2/kernel/netkit-kernel
Modules: /mnt/netkit/netkit2/kernel/modules
Memory: 8 MB
Model fs: /mnt/netkit/netkit2/fs/netkit-fs
Filesystem: /mnt/netkit/netkit2/bin/pc2.disk
Hostfs at: /root

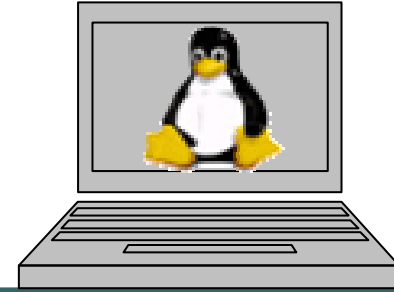
Running ==> /mnt/netkit/netkit2/kernel/netkit-kernel modules=/mnt/netkit/netkit2
/kernel/modules name=pc2 title=pc2 umid=pc2 mem=12M ubd0=/mnt/netkit/netkit2/bin
/pc2.disk /mnt/netkit/netkit2/fs/netkit-fs root=98:1 uml_dir=/root/.netkit/mcone
ole hosthome=/root quiet con0=term con1=null
root@ubuntu:/mnt/netkit/netkit2/bin#

pc1
Setting up networking...done.
Initializing random number generator...done.
INI: Entering runlevel: 2
--- Starting Netkit phase 1 startup script
Setting terminal title to "pc1"
Mounting host home (/root) on /hosthome...
--- Netkit phase 1 init script terminated
Starting kernel log daemon: klogd.
Configuring network interfaces...done.
Starting system log daemon: syslogd.
--- Starting Netkit phase 2 startup script
Virtual host pc1 ready.
--- Netkit phase 2 init script terminated
pc1 login: root (automatic login)
Linux pc1 2.6.11.7 #1 Tue Sep 13 18:38:01 CEST 2005 1686 GNU/Linux
Welcome to Netkit
pc1:~#

pc2
Setting up networking...done.
Initializing random number generator...done.
INI: Entering runlevel: 2
--- Starting Netkit phase 1 startup script
Mounting host home (/root) on /hosthome...
--- Netkit phase 1 init script terminated
Starting kernel log daemon: klogd.
Configuring network interfaces...done.
Starting system log daemon: syslogd.
--- Starting Netkit phase 2 startup script
Virtual host pc2 ready.
--- Netkit phase 2 init script terminated
pc2 login: root (automatic login)
Linux pc2 2.6.11.7 #1 Tue Sep 13 18:38:01 CEST 2005 1686 GNU/Linux
Welcome to Netkit
pc2:~#
```

Figura 1 - WinNetkit in ambiente windows con vmware player -

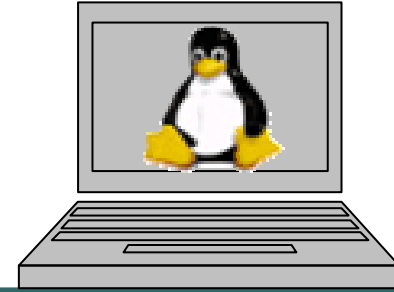
Il progetto NetKit



- Il netkit include :
- Kernel user Mode (UML)
- Root filesystem debian
- Uml utilities
- Vari Script in python



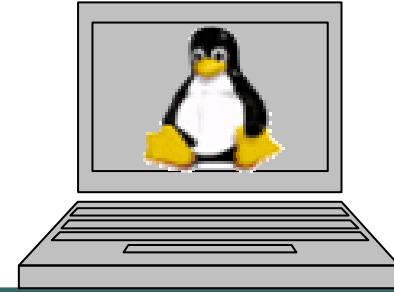
Il progetto NetKit



- L'installazione del pacchetto in tre fasi:
- Il prelievo dei seguenti file:
- Netkit-2.4.tar.bz2
- Netkit-filesystem-F2.2.tar.bz2
- Netkit-Kernel-k2.2.tar.bz2



Il progetto NetKit



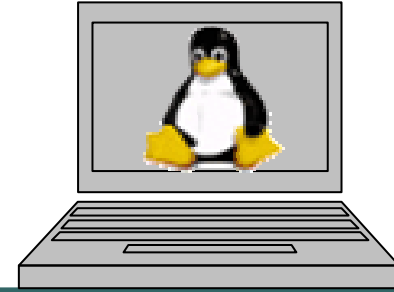
- Host> tar xjf Netkit-2.4.tar.bz2
- Host> tar xjf Netkit-filesystem-F2.2.tar.bz2
- Host> tar xjf Netkit-Kernel-k2.2.tar.bz2

- Da www.netkit.org

- Path user :
- set NETKIT_HOME to the path where you installed
- Netkit
- set PATH to the string
- "\$PATH:\$NETKIT_HOME/bin"
- set MANPATH to the string ":\$NETKIT_HOME/man"



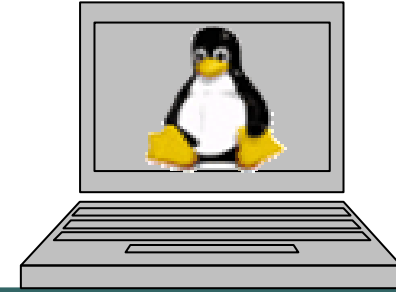
Il progetto NetKit



- Terminata la seconda fase procediamo con il comando :
- `./check_configuration.sh`
- Oltre a controllare la presenza di tutto quello che serve ,indica l'impostazione della variabile d'ambiente `vlab-home` che dovrà essere inclusa nel path dell'utente



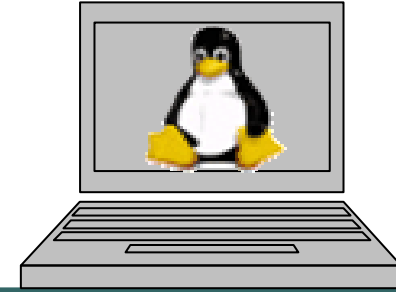
Il progetto NetKit



- Netkit offre all'utente seguenti comandi:
- Vstart (avvia una macchina virtuale)
- Vconfig (attacca un'interfaccia di rete a una vm)
- Vlist (esegue una lista di macchine avviate)
- Vhalt (termina la macchina virtuale)
- Vcrash (sopprime la macchina virtuale)
- Vclean ("bottono del panico" sopprime tutte le macchine virtuali, rimuove tunnel e altro)



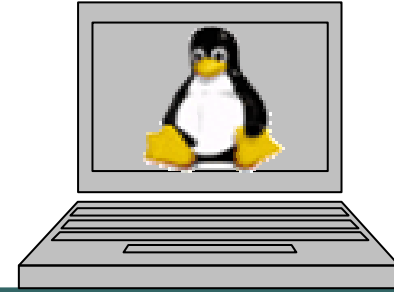
Il progetto NetKit



- A questo punto possiamo avviare una macchina virtuale con Netkit, che chiameremo “virtuale” tramite il comando `vstart`.
- `Host> vstart virtuale -- new`
- Verrà aperto un terminale che rappresenta la console “virtuale”



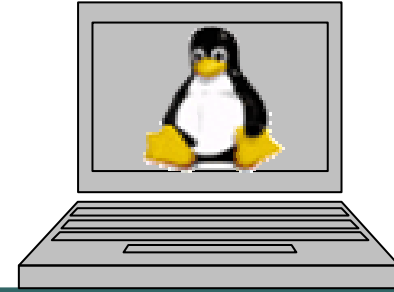
Il progetto NetKit



- --- Starting Netkit phase 2 startup script
- Virtual host virtuale ready.
- --- Netkit phase 2 init script terminated
- virtuale login: root (automatic login)
- Linux virtuale 2.6.11.7 #1 Tue Sep 13 18:38:01 CEST 2005 i686
- GNU/Linux
- Welcome to Netkit
- virtuale:~# █



Il progetto NetKit



- L'approccio consigliato è quello di disegnare la topologia della rete che si desidera emulare, definendo il ruolo delle macchine virtuali, dare i nomi ai dispositivi di rete ecc ecc
- Iniziamo a capire meglio con una semplice configurazione il progetto netkit . Quello che proponiamo di fare è una piccola rete con quattro macchine virtuali sul medesimo network 10.0.0.0/24

Il progetto NetKit

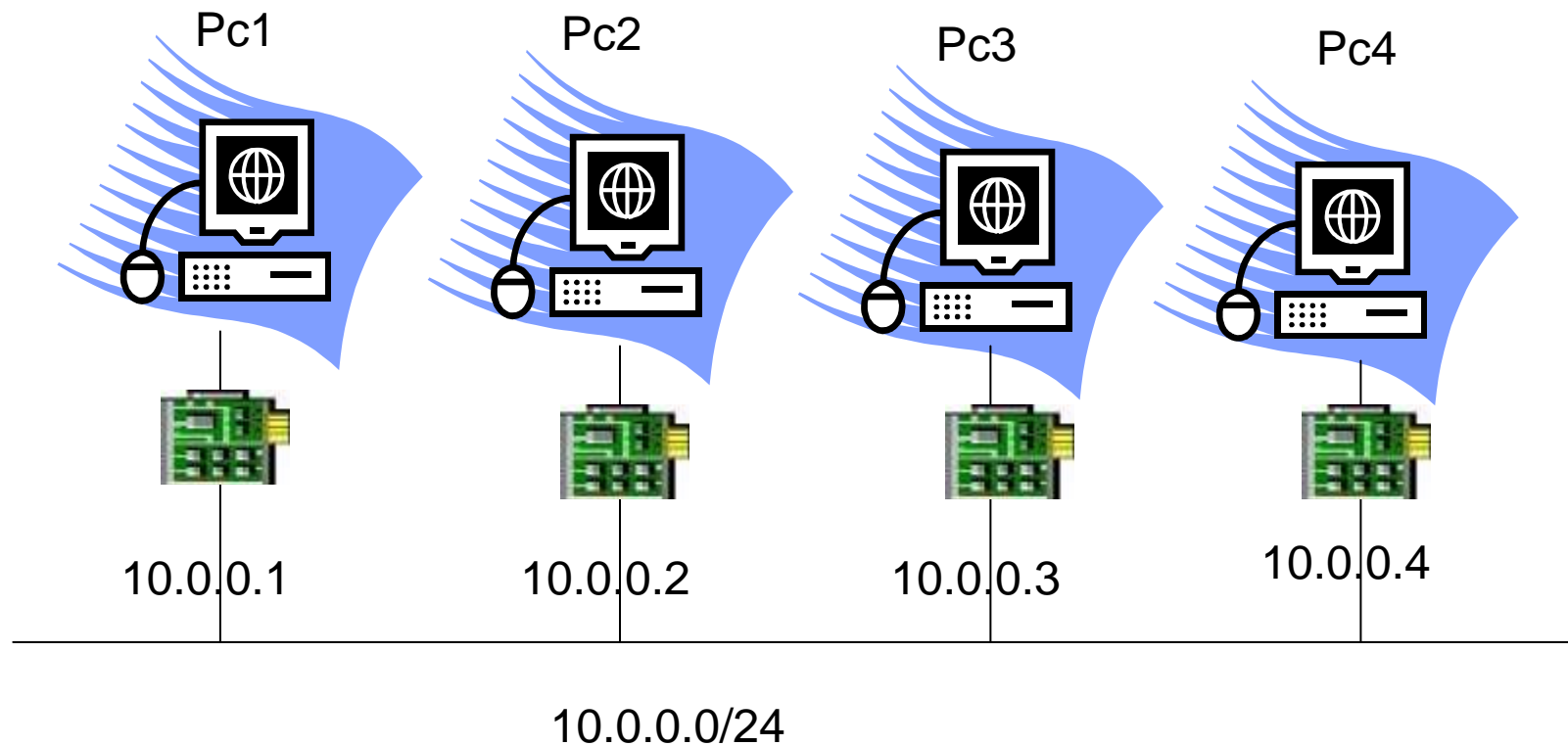
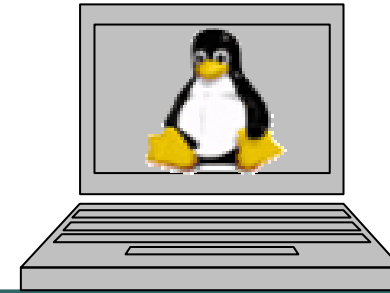
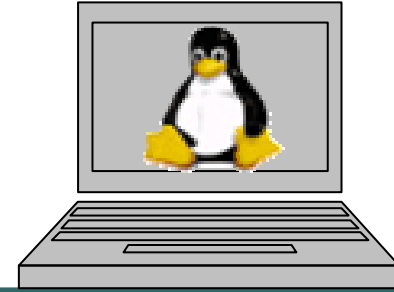


FIGURA 2 - quattro macchine virtuali sullo stesso network

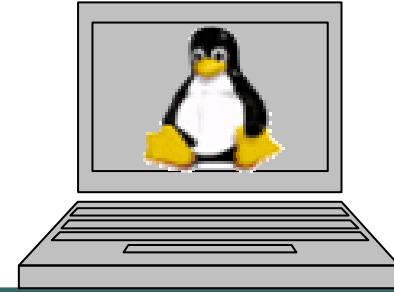
Il progetto NetKit



- Passiamo con i comandi necessari
- Host> Vstart Pc1 -eth0=A1 - new
- Host> Vstart Pc2 -eth0=A1 - new
- Host> Vstart Pc3 -eth0=A1 - new
- Host> Vstart Pc4 -eth0=A1 - new



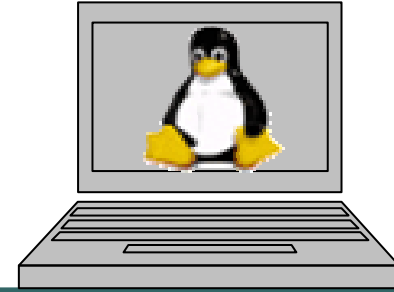
Il progetto NetKit



- Pc1# `ifconfig eth0 10.0.0.1 netmask 255.255.255.255 up`
- Pc2# `ifconfig eth0 10.0.0.2 netmask 255.255.255.255 up`
- Pc3# `ifconfig eth0 10.0.0.3 netmask 255.255.255.255 up`
- Pc4# `ifconfig eth0 10.0.0.4 netmask 255.255.255.255 up`



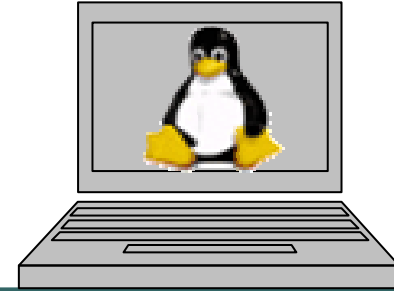
Il progetto NetKit



- Per verificare la rete basterà lanciare una serie di ping dal Pc2 al Pc4 per esempio .
- Pc2 # ping -c 5 10.0.0.4
- Pc1 # ping -c 5 10.0.0.2



Il progetto NetKit



- Proseguiamo con una tipologia di lan più complessa per lo studio del protocollo ARP specificato dalla RFC826.
- Il protocollo ARP è usato da tecnologie basate sul broadcast ad accesso condiviso quale è ethernet.
- I frame in broadcast aggiornano la cache arp di tutti i nodi connessi al medesimo segmento.
- Esempio reperibile su www.netkit.org

Il progetto NetKit

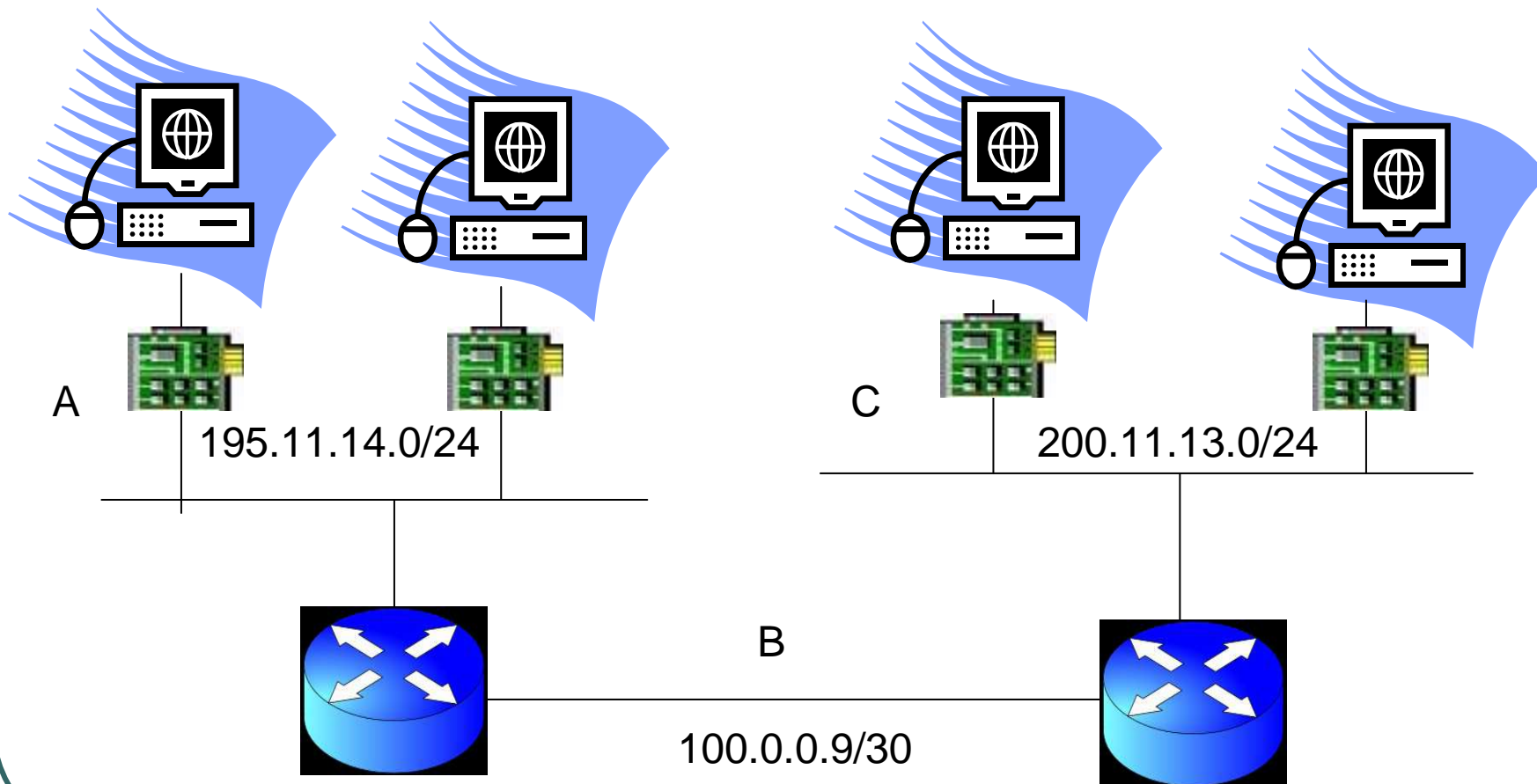
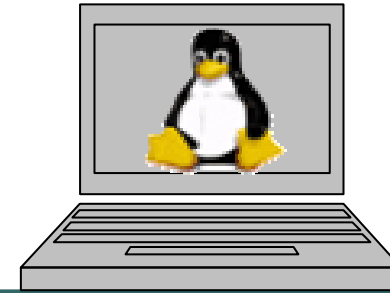
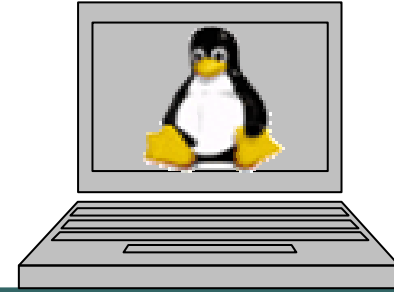


FIGURA 3 – PROTOCOLLO ARP -

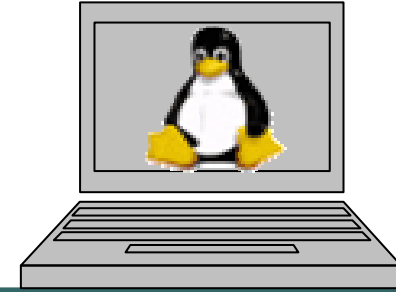
Il progetto NetKit

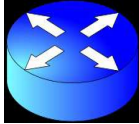


- Procediamo con la creazione dei dispositivi.
- `Vstart pc1 -eth0=A --new`
- `Vstart pc2 -eth0=A --new`
- `Vstart pc3 -eth0=C --new`
- `Vstart pc4 -eth0=C --new`



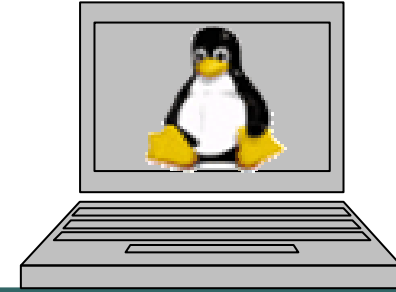
Il progetto NetKit



- I due router 
- `Vstart router1 -eth0=A -eth0=B -new`
- `Vstart router2 -eth0=C -eth0=B -new`
- Dopo aver definito i relativi router non ci resta che entrare in una singola macchina e passare ai comandi.



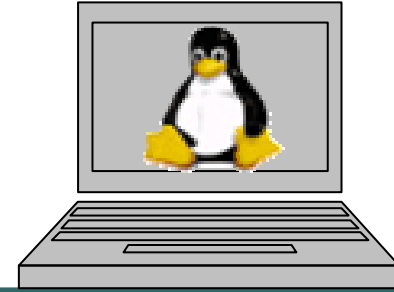
Il progetto NetKit



- Router1 # ifconfig eth0 195.11.14.1 up
- Router1# ifconfig eth1 100.0.0.9 netmask 255.255.255.255 broadcast 100.0.0.11 up
- Router1# route add -net 200.1.1.0 netmask 255.255.255.0 gw 100.0.0.10 eth1
- Router2# ifconfig eth0 200.1.1.1 up
- Router2# ifconfig eth1 100.0.0.9 netmask 255.255.255.255 broadcast 100.0.0.11 up
- Router2# route add -net 195.11.14.0 netmask 255.255.255.0 gw 100.0.0.9 eth1



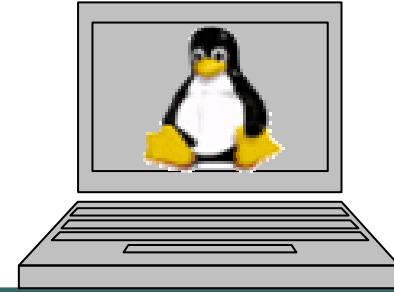
Il progetto NetKit



- Pc1# ifconfig eth0 195.11.14.5 up
- Pc1#route add default gw 195.11.14.1
- Pc2# ifconfig eth0 195.11.14.6 up
- Pc2#route add default gw 195.11.14.1
- Pc3 # ifconfig eth0 200.1.1.7 up
- Pc3# route add default gw 200.1.1.1
- Pc4 # ifconfig eth0 200.1.1.8 up
- Pc3# route add default gw 200.1.1.1



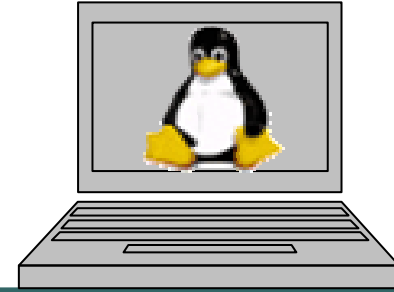
Il progetto NetKit



- Dopo aver configurato il tutto è il momento di usare ARP.
- Pc2# arp -n
- Pc3#arp -n
- Anche in questa fase il ping ci porge aiuto
- Pc3# ping -c 1 200.1.1.7
- Data la cache arp vuota pc3 dovrà mandare il messaggio arp-request affinché sia risolto il suo mac address.

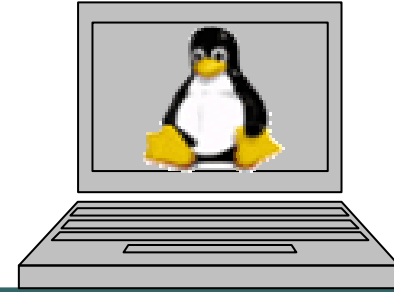


Il progetto NetKit



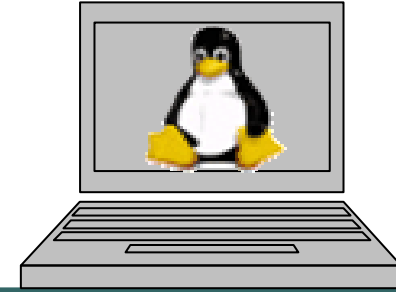
- Configurare tutto a mano è istruttivo ma molto laborioso.
- Si ricorre quindi a determinati script che si accollano di effettuare il setup al posto nostro.
- Questi script (LAB SCRIPT) sono successioni di comandi necessari alla creazione della rete che vogliamo emulare.
- Il sito www.netkit.org offre alcuni esempi già pronti all'uso.

Il progetto NetKit



- Possiamo scindere due tipologie di script accettabili per netkit:
- Bash script
- Netml
- Il primo è un interprete di comandi per sistema operativo linux ,al contrario netml è nato per favorire l'utente nell'uso di netkit

Il progetto NetKit



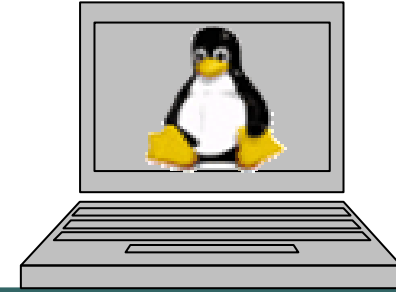
Es. Bash script (Tratto da www.netkit.org)

- -----
- `#!/bin/bash`
- `...`
- `if [$HOSTNAME = $PC1]; then`
- `/sbin/ifconfig eth0 10.0.0.1 up`
- `/sbin/ifconfig eth0 hw ether 00:00:00:00:10:00`
- `fi`

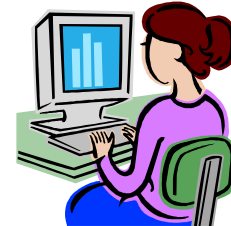


- -----
- Una volta compiuto il proprio bash script(.netkit),sarà sufficiente avviare la propria rete virtuale con netkit.

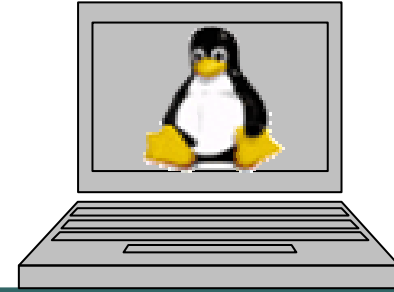
Il progetto NetKit



- Per poter usare gli script lab, il progetto mette a disposizione il comando ltool con alcuni parametri :
- Lstart (avvia uno script lab)
- Ltest (avvia in test mode)
- Lhalt (termina lo script lab)
- Lcrash (termina una virtual machine)
- Linfo (mostra le informazioni sullo script lab)
- Lclean (toglie temporaneamente uno script lab)

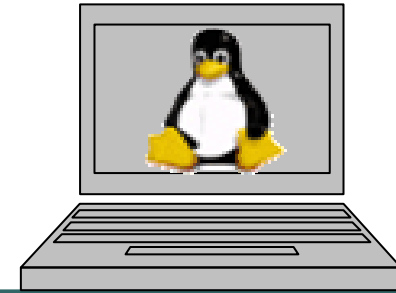


NetKit e NetML

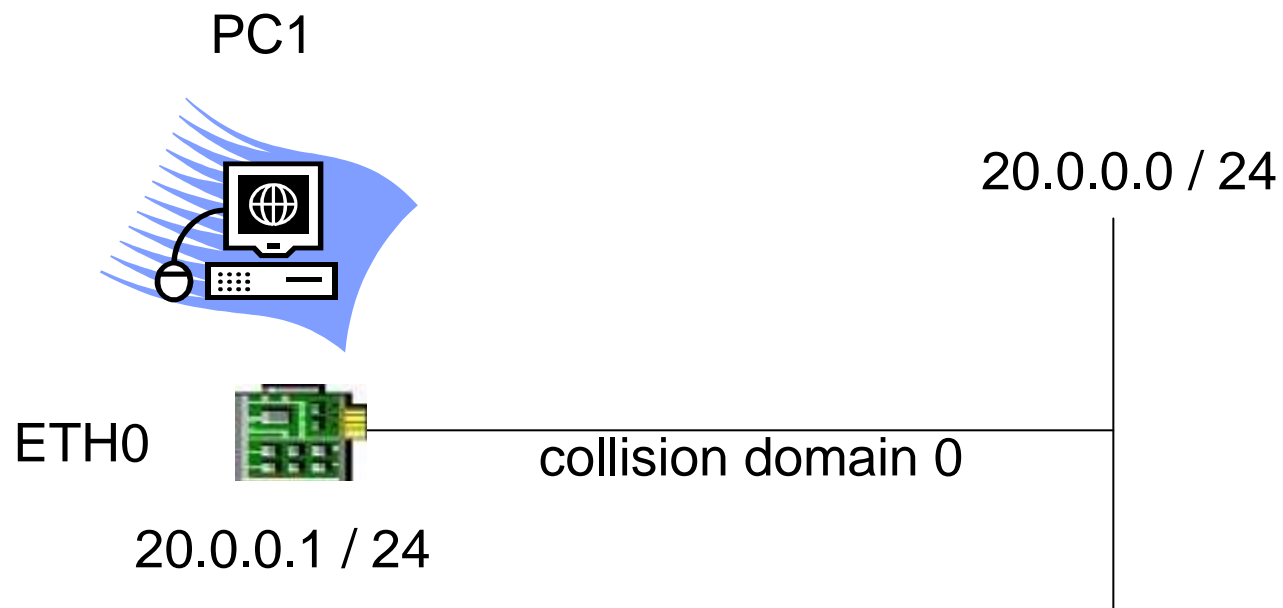


- Netml (Network Modelling Language).
- Il linguaggio Netml si basa su XML, una breve descrizione è disponibile su:
 - <http://cs.sci.usq.edu.au/netml/netml>
 - <http://cs.sci.usq.edu.au/netml2>
 - <http://www.w3.org/TR/xmlschema-0/>
- netkit propone in più un tool :
- NetML-2005-12-10.tar.gz per la sintassi netml.

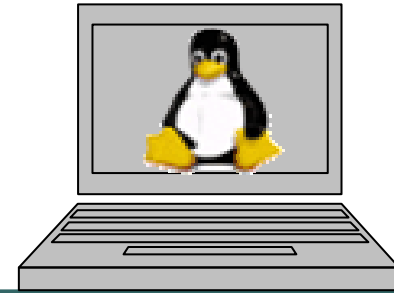
NetKit e NetML



- Continuiamo con un piccolo esempio di sintassi netml con il seguente modello:



NetKit e NetML



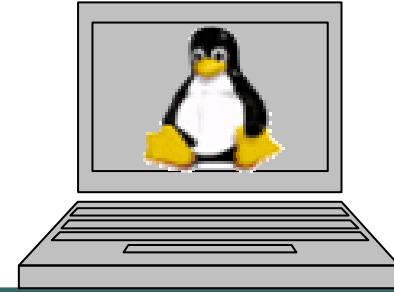
- Aggiungiamo il seguente codice standard :

- `<?xml version="1.0" encoding="ISO-8859-1" ?>`



- `<NetML xmlns="http://www.xmlnetwork.org"`
- `xmlns:xsi="http://www.w3.org/2001/XMLSchema-`
- `instance"`
- `xsi:schemaLocation="http://www.xmlnetwork.org`
- `netml.xsd">`

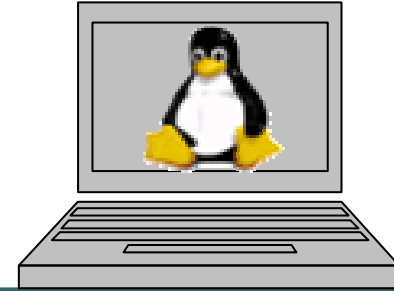
NetKit e NetML



- `<Networks>//inizio`
- `<n id="n-A">`
- `<networkAddress>20.0.0.0/24`
- `</networkAddress> //fine`
- `<CollisionDomains>// inizio`
- `<c id="C0">`
- `<iface r_id="r_1" if="eth0"/>`
- `</c>`
- `</CollisionDomains> //fine`



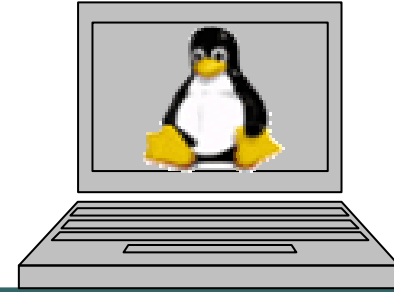
NetKit e NetML



- `</n>`
- `</Networks> //fine network`
- `<Routers> //dichiaro un router`
- `<RouterConf id="r_1" Hostname="pc1"> // pc1 è router1`
- `<Interface name="eth0"> //nome interfaccia eth0`
- `<ipAddress>20.0.0.1/24`
- `</ipAddress> // fine ipaddress`
- `</Interface> //fine interfaccia`
- `</RouterConf> //fine routerConf`
- `</Routers> //fine routers`
- `</NetML> //fine script netml`



NetKit e NetML



- Conclusioni
- Netkit è stato concepito come un ambiente a basso costo per sperimentazioni di rete.
- All'interno del suo ambiente possono essere creati e congiunti router, switch e host.