

UML

[Sito del corso di Laboratorio di Reti di Calcolatori](#)

[Binari User Mode Linux](#)

[Immagine sistema installato](#)

[Immagine sistema installato \(2 dischi\)](#)

Eseguibili:

- linux (il kernel)
- uml_moo, uml_mkcow (utility per la gestione dei dischi in modalità Copy On Write)
- uml_switch (gestione della rete virtuale)
- altre utility...

Scaricare e scompattare i binari di UML e le immagini del sistema nella directory corrente

- **Lanciare lo switch (in un console separata)**

```
$ ./uml_switch
```

- **Lanciare user mode linux**

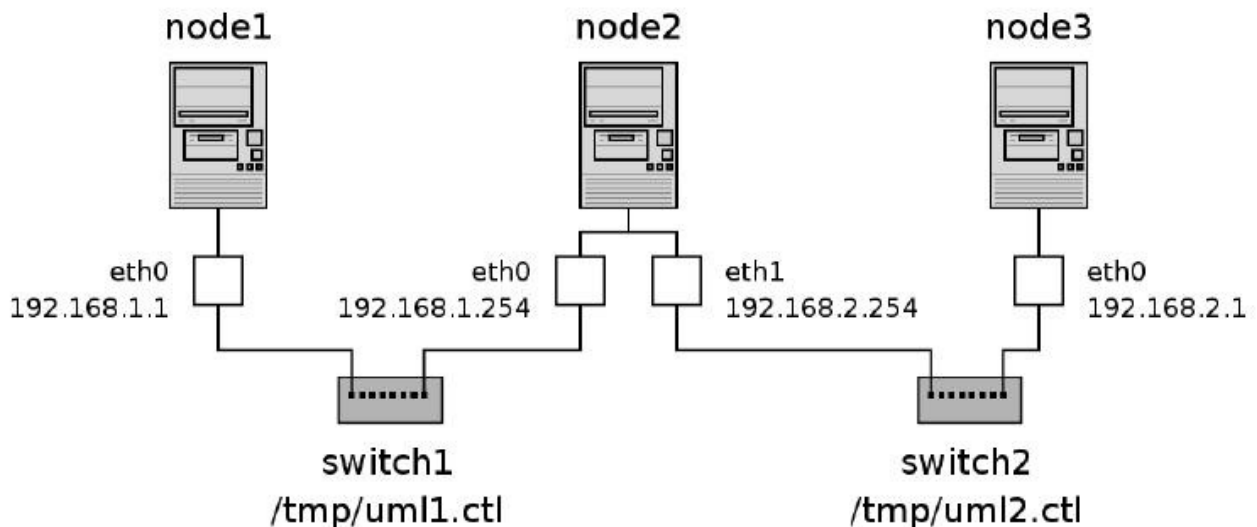
```
$ ./linux ubd0=installato.cow,installato.ext2 eth0=daemon,,unix,/tmp/uml.ctl
```

interfaccia di rete: eth0 su switch1 (/tmp/uml.ctl)

```
$ ./linux ubd0=root1.cow,root.ext2 ubd1=usr1.cow,usr.ext2 eth0=daemon,,unix,/tmp/uml.ctl
```

2 dischi: ubd0 e ubd1

MAI chiudere le finestre di UML, dare il comando shutdown dalla linea di comando del sistema UML



Fare partire il supporto per la rete (i due switch virtuali)

- switch1
\$./uml_switch -unix /tmp/uml1.ctl
- switch2
\$./uml_switch -unix /tmp/uml2.ctl

Se serve usare degli hub usare il comando:

```
$ ./uml_switch -hub -unix /tmp/uml1.ctl
```

Lanciare i sistemi virtuali

- node1
\$./linux ubd0=installato1.cow,installato.ext2 eth0=daemon,,unix,/tmp/uml1.ctl
- node3
\$./linux ubd0=installato3.cow,installato.ext2 eth0=daemon,,unix,/tmp/uml2.ctl
- node2
\$./linux ubd0=installato2.cow,installato.ext2 eth0=daemon,,unix,/tmp/uml1.ctl \
eth1=daemon,,unix,/tmp/uml2.ctl

Ogni volta che si lancia un nodo dovrebbe apparire un'indicazione nelle schermate dove girano gli switch.

Configurare i nodi

Si fa il login su ciascuno dei nodi e si lavora da linea di comando sui nodi virtuali

- **Login:**
Username: root
Password nulla (basta battere invio)
- **Dare un nome ai nodi:**
Editare il file /etc/hostname in modo che contenga il nome del nodo

Inserire le giuste informazioni di rete per risolvere i nomi

Si agisce sul file /etc/hosts

Alla fine ogni nodo dovrebbe avere un file fatto in questo modo:

127.0.0.1	localhost
192.168.1.1	node1
192.168.2.1	node3
192.168.1.254	node2
192.168.2.254	node2

Configurare le interfacce di rete

Direttiva iface nel file /etc/network/interfaces

- <modalità> vale:
dhcp = inizializzazione dell'interfaccia in modo automatico col protocollo dhcp
loopback = interfaccia locale (127.0.0.1)
static = seguono parametri per configurare l'interfaccia
- Configurazione statica di un'interfaccia
address = indirizzo IP dell'interfaccia (192.168.X.Y)
netmask = 255.255.255.0
network = indirizzo della rete (192.168.X.0)
broadcast = indirizzo di broadcast (192.168.X.255)
gateway = solo per node1 e node3, l'indirizzo dell'interfaccia di node2 che si affaccia sulla sottorete

Esempio

```
# File di configurazione per node1
# The loopback network interface
auto lo
iface lo inet loopback
```

```
# Interface eth0
auto eth0
# iface eth0 inet dhcp
iface eth0 inet static
    address 192.168.1.1
    netmask 255.255.255.0
    broadcast 192.168.1.255
    network 192.168.1.0
    gateway 192.168.1.254
```

Possiamo riavviare tutti e tre i sistemi

```
$ poweroff
```

e poi farli ripartire

Perché node 1 e 3 non comunicano?

Serve che node 2 diventi un vero router e passi il traffico tra le sue due interfacce (eth0 e eth1)

```
# echo "1" >/proc/sys/net/ipv4/ip_forward
```

Ora dovremmo essere in grado di far comunicare anche node1 e node3 attraverso node2

Salvare il lavoro fatto

Bisogna preservare le date di accesso:

Opzione -p del comando tar (preserve)

Esempio:

```
$ tar czpvf mie_immagini.tgz *.cow *.ext2
```

Questo comando serve per aggiungere altri gateway a un nodo:

```
route add -net 192.168.x.0 netmask 255.255.255.0 gw 192.168.x.x
```

dovrebbe essere utile per il compito postato sul forum