Università degli Studi Roma Tre
Dipartimento di Informatica e Automazione
Computer Networks Research Group

netkit lab

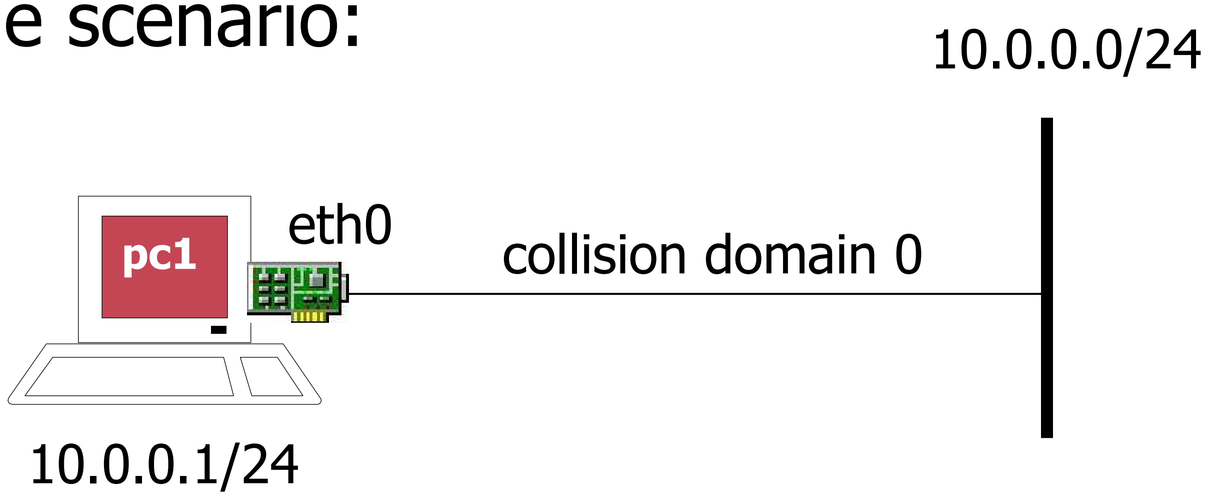
single-host

Version	2.2
Author(s)	G. Di Battista, M. Patrignani, M. Pizzonia, M. Rimondini
E-mail	contact@netkit.org
Web	http://www.netkit.org/
Description	how to set up and manage a single virtual machine

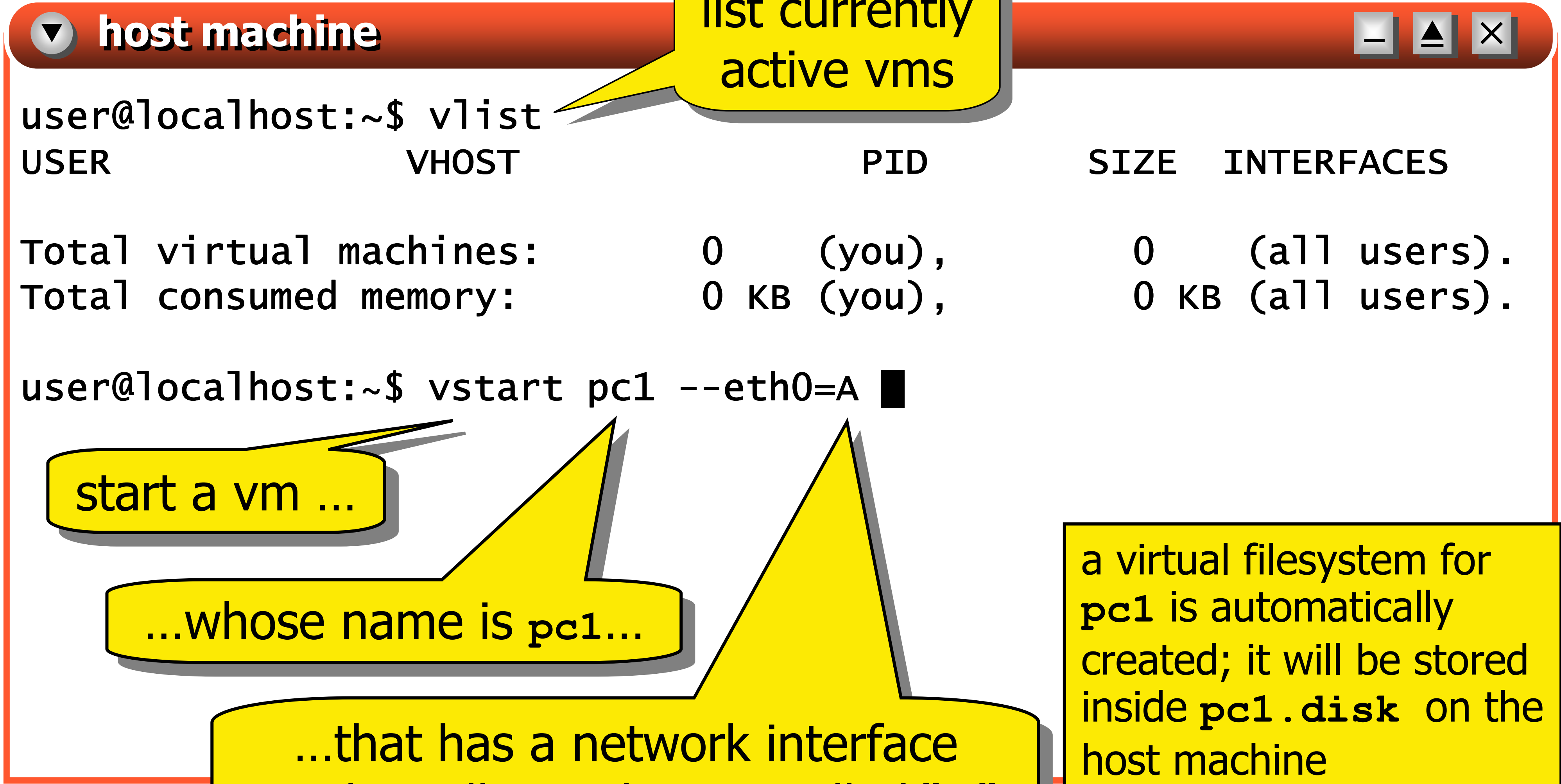
copyright notice

single host

- netkit *little by little*: just a single virtual machine
- suggestion: before setting up a netkit lab, always make a clear diagram of the scenario that you are going to emulate!
- a simple scenario:



step 1 – creating a virtual machine



host machine

```
user@localhost:~$ vlist
USER                VHOST                   PID        SIZE  INTERFACES

Total virtual machines:      0    (you),          0    (all users).
Total consumed memory:       0 KB (you),          0 KB (all users).

user@localhost:~$ vstart pc1 --eth0=A
```

list currently active vms

start a vm ...

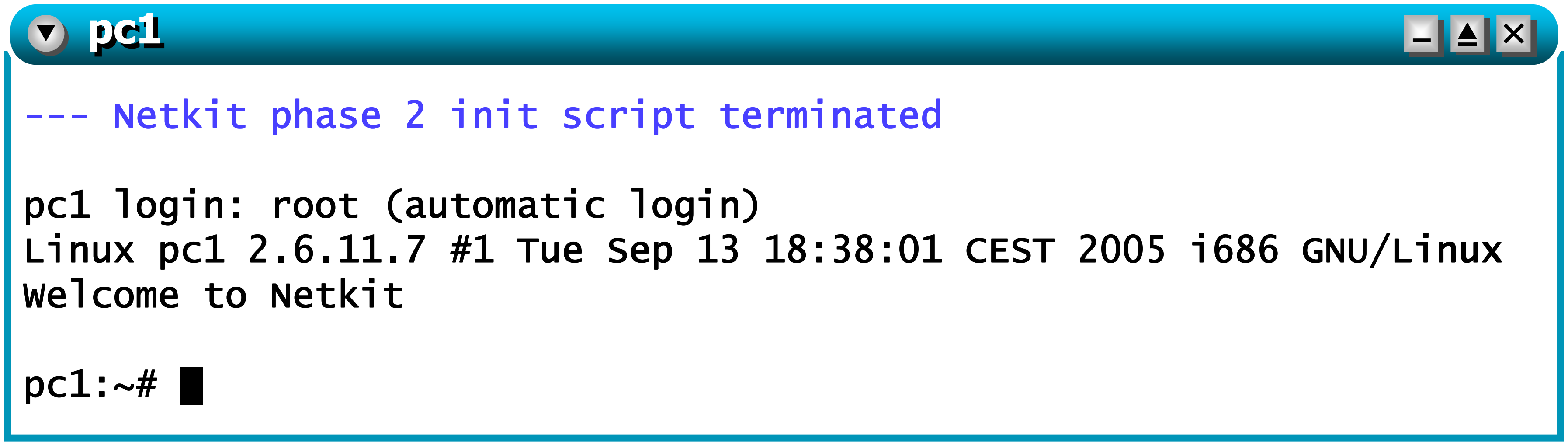
...whose name is **`pc1`**...

...that has a network interface on the collision domain called "**A**"

a virtual filesystem for **`pc1`** is automatically created; it will be stored inside **`pc1.disk`** on the host machine

step 2 – logging on **`pc1`**

- a window containing **`pc1`**'s console pops up
- once the bootstrap of **`pc1`** is terminated, a shell prompt is automatically displayed


pc1

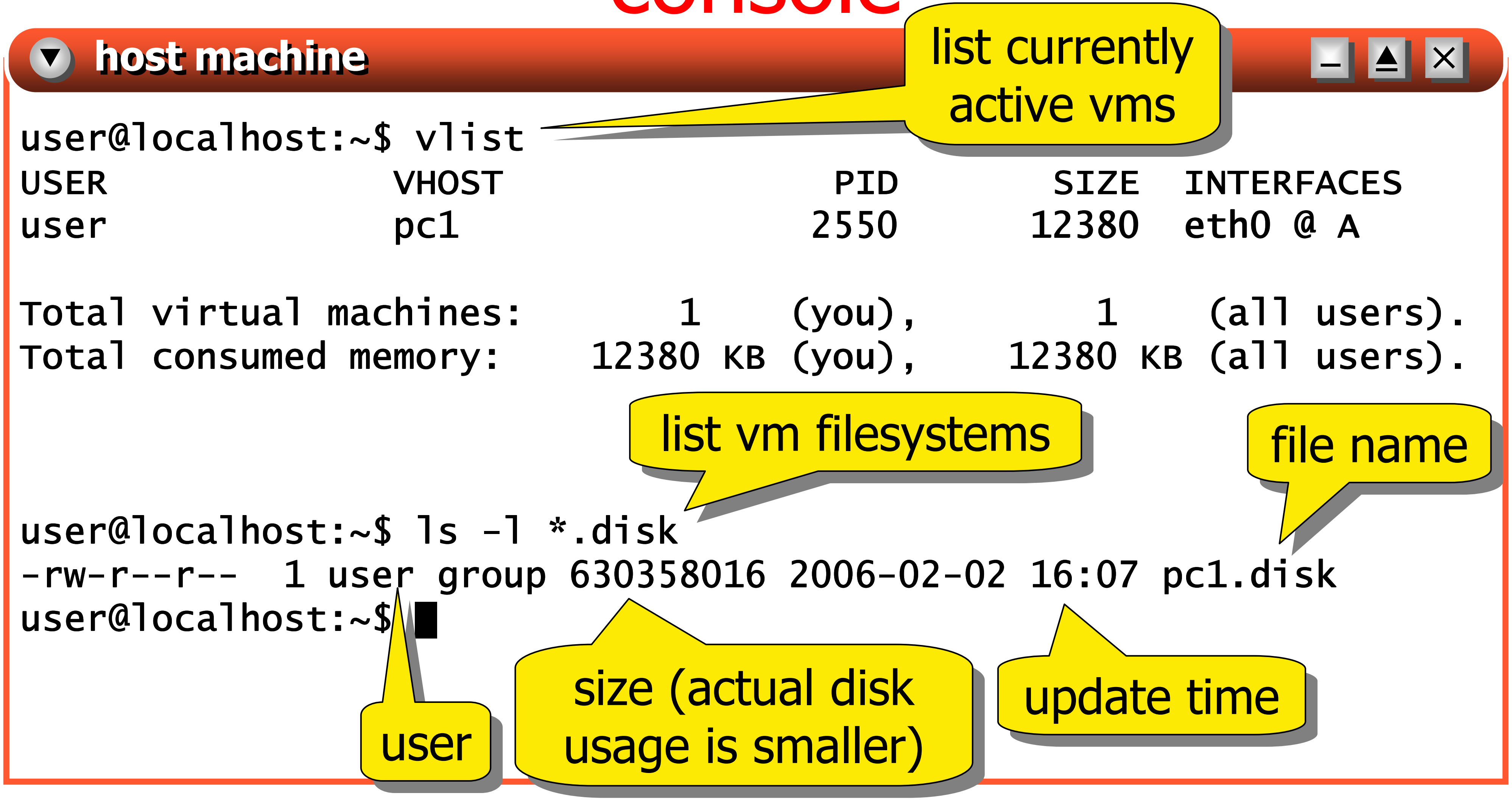
```
--- Netkit phase 2 init script terminated

pc1 login: root (automatic login)
Linux pc1 2.6.11.7 #1 Tue Sep 13 18:38:01 CEST 2005 i686 GNU/Linux
Welcome to Netkit

pc1:~# █
```

- now you are the administrator (root) of **`pc1`**

step 3 – back to the host machine console


host machine
list currently active vms
list vm filesystems
file name
user
size (actual disk usage is smaller)
update time

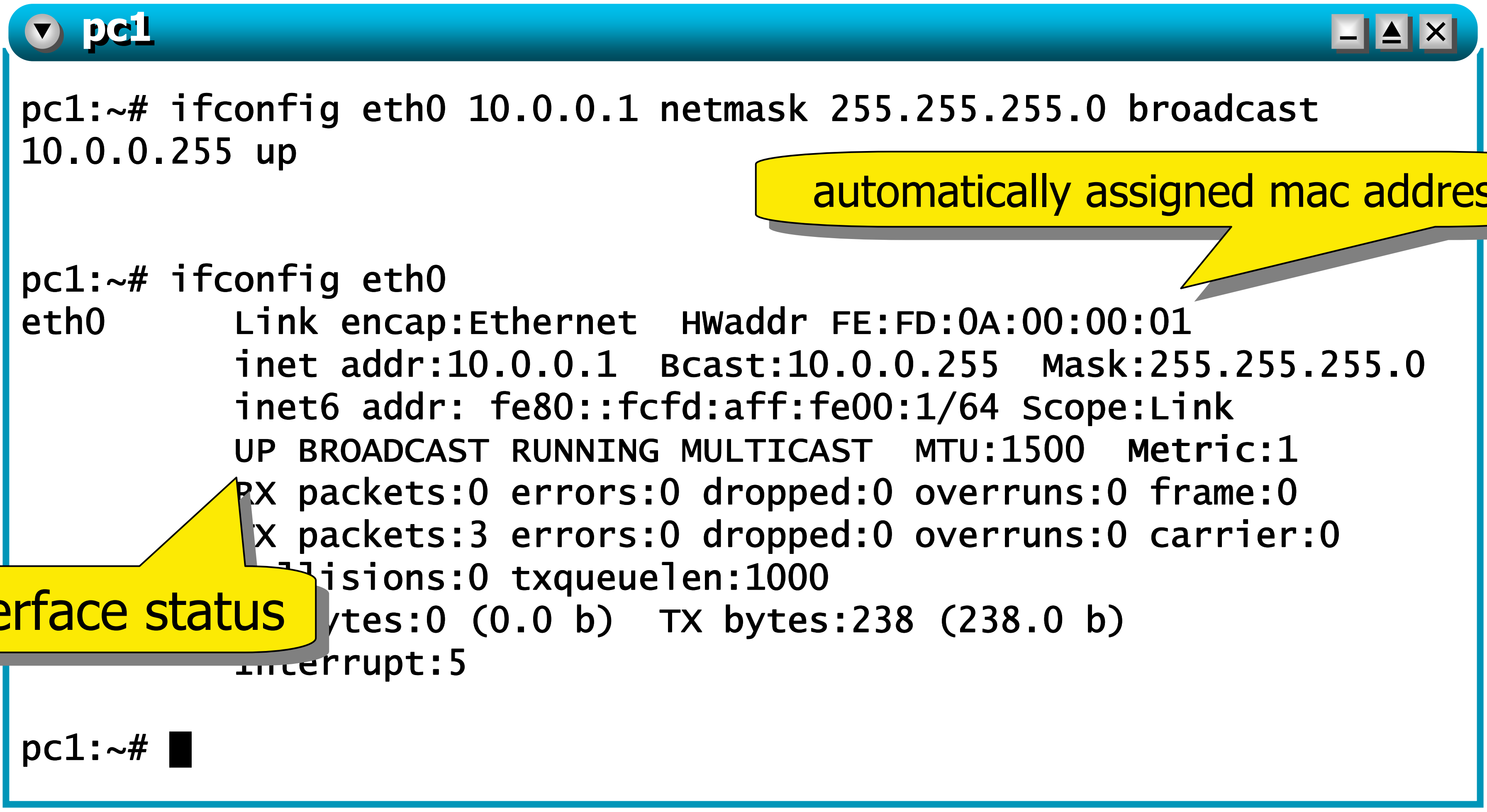
```
user@localhost:~$ vlist
USER                VHOST                          PID        SIZE  INTERFACES
user                pc1                           2550       12380  eth0 @ A

Total virtual machines:       1    (you),         1    (all users).
Total consumed memory:    12380 KB (you),     12380 KB (all users).



user@localhost:~$ ls -l *.disk
-rw-r--r--  1 user group 630358016 2006-02-02 16:07 pc1.disk
user@localhost:~$ █
```

step 4 – configuring the network interface of **`pc1`**



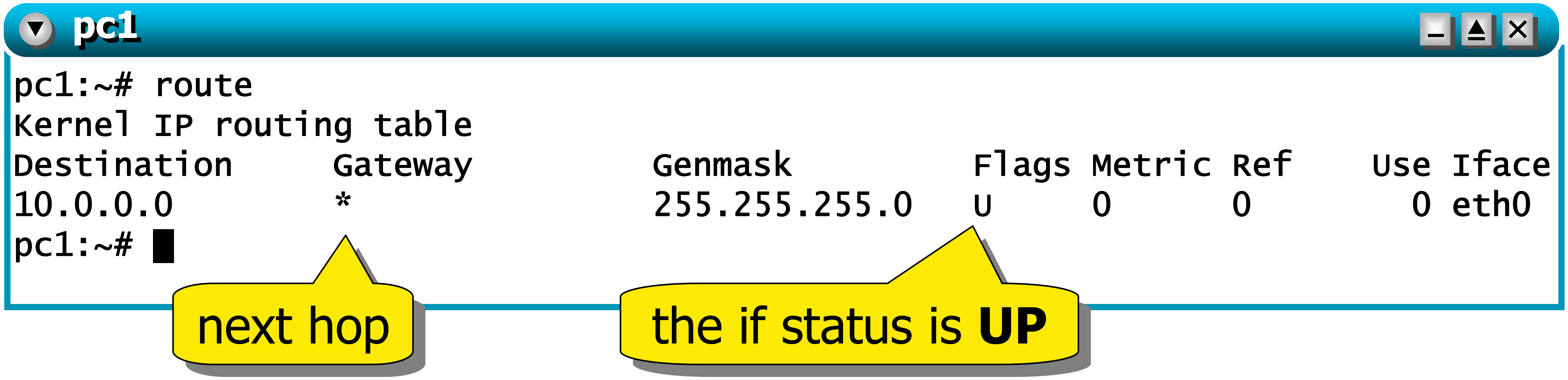
```
pc1:~# ifconfig eth0 10.0.0.1 netmask 255.255.255.0 broadcast
10.0.0.255 up


pc1:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr FE:FD:0A:00:00:01
          inet addr:10.0.0.1  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::fcfd:aff:fe00:1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
           X packets:3 errors:0 dropped:0 overruns:0 carrier:0
            llisions:0 txqueuelen:1000
              ytes:0 (0.0 b)  TX bytes:238 (238.0 b)
          interrupt:5

pc1:~# █
```

step 5 – checking the routing table

- the routing table has been automatically updated when the interface has been brought up:

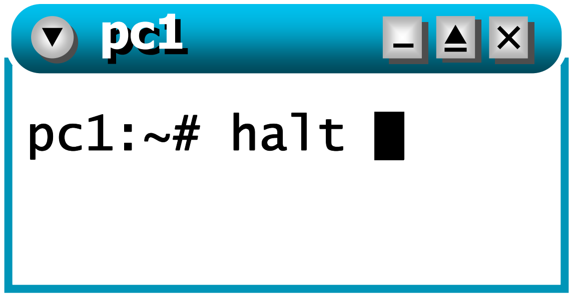


```
pc1:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        *               255.255.255.0   U     0      0        0 eth0
pc1:~# 
```

- other labs show how to manually alter the routing table

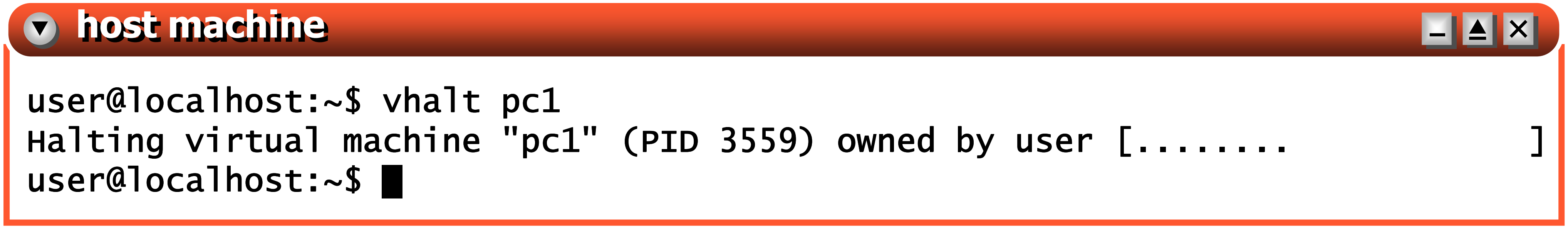
step 6 – shutting down the vm

- three possibilities
  - from inside



```
pc1:~# halt
```

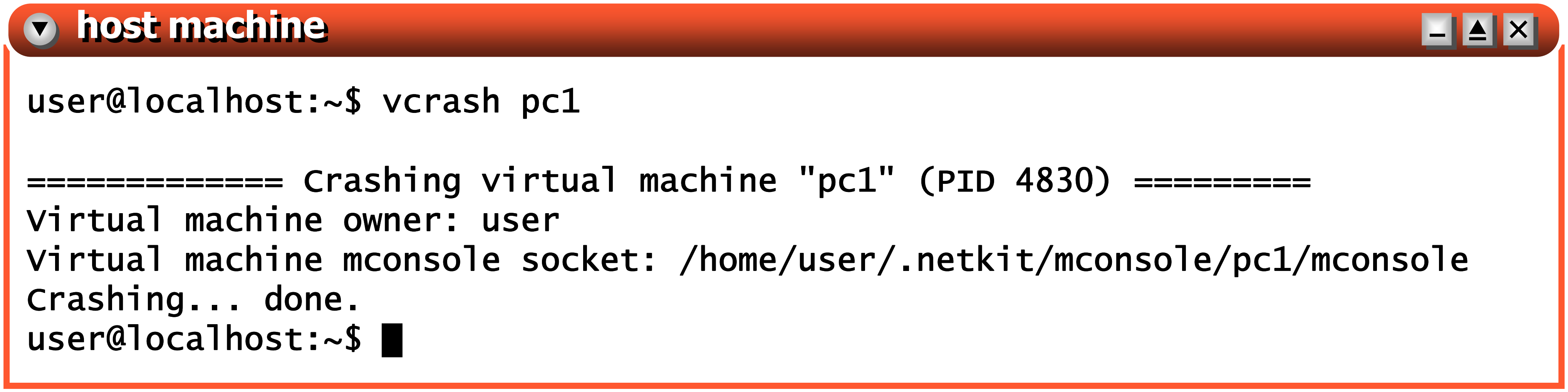
  - from outside



```
user@localhost:~$ vhalt pc1
Halting virtual machine "pc1" (PID 3559) owned by user [........        ]
user@localhost:~$
```

step 6 – shutting down the vm

- from outside, brute force



```
user@localhost:~$ vcrash pc1

============= Crashing virtual machine "pc1" (PID 4830) =========
Virtual machine owner: user
Virtual machine mconsole socket: /home/user/.netkit/mconsole/pc1/mconsole
Crashing... done.
user@localhost:~$ █
```

- unless you chose to use vcrash, **`pc1`**’s filesystem is still stored in file **`pc1.disk`**, so it will be used again when **`pc1`** is restarted

step 7 – a permanent configuration

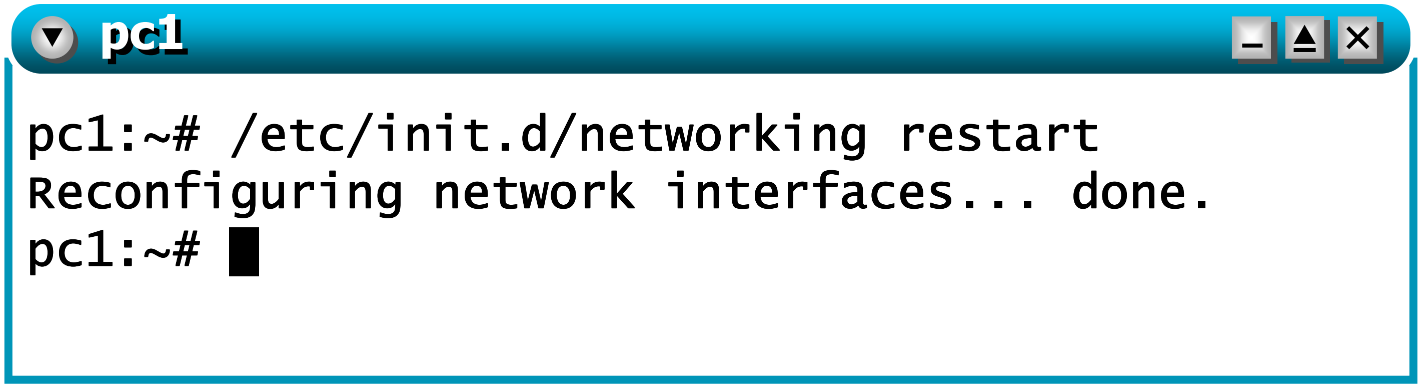
- after halting **`pc1`**, if you want to restart it you also have to reconfigure its network interface **`eth0`**
- a permanent configuration can be obtained, e.g., by editing **`/etc/network/interfaces`** (inside the vm **`pc1`**) and appending the following lines:

```
auto eth0
iface eth0 inet static
   address 10.0.0.1
   network 10.0.0.0
   netmask 255.255.255.0
```

- tips:
  - you can use an editor like **`vi`** or **`mcedit`**
  - the permanent settings of a vm can be configured inside the same files that would be used in a real linux box
- removing the vm filesystem (**`pc1.disk`**) removes any permanent configuration as well

step 8 – restarting network services

- at next boot **`pc1`** will be automatically configured by the os which will perform the suitable **`ifconfig`** and **`route`** commands based on the contents of **`/etc/network/interfaces`**
- the new configuration can also be fetched without rebooting by restarting network services:



```
pc1:~# /etc/init.d/networking restart
Reconfiguring network interfaces... done.
pc1:~# █
```