

Using VMWare to Capture Linux Installation Screen Shots

How do you capture screen shots of an operating system installation? There are a couple of ways. One is to take digital photographs of the screen. Blech. Another way is to install a piece of software called VMWare on an existing OS. Then, use VMWare to create a shell in which the OS in question can be installed - and then use the screen capture facilities of the existing OS to grab the screens of the OS being installed. Here's a detailed explanation of how this works.

Ask the "How do you capture screen shots?" question on any forum and you'll immediately get one word responses from a number of folks: VMWare. It's a bit more involved than that. While VMWare is designed for a wide variety of purposes, I'm going to explain how to use VMWare specifically to capture screen shots of a Linux install. I've installed Fedora Core on a machine, and will install VMWare and then a second copy of Fedora Core on the same machine, using VMWare to get the screen shots of the installation process. Finally, I'll uninstall the copy of Fedora Core, and uninstall VMWare as well.

What is VMWare?

VMWare is a piece of software that acts as a shell into which other operating systems can be installed and run. It is a commercial application, which means you have to pay for it. However, you can download a trial version that works just like the real product for 30 days.

VMWare is installed on an operating system that is then referred to as the "host". Once installed, VMWare is used to create objects called 'virtual machines' that are contained in large disk files on the host computer. Additional operating systems - called 'guest operating systems' - are installed inside these virtual machines, one OS per virtual machine. The user can then use any of the guest operating systems just as if they were installed on the machine natively, but they run entirely within VMWare, and thus don't interfere with anything else on the machine.

An Overview of the Process

First, the host operating system must be installed with enough spare disk space to hold the disk files for each virtual machine. How much disk space is required for any specific virtual machine varies according to the operating system you're going to install in it. For example, if you're going to install a copy of Linux, you should have 3 to 6 GB of free space, depending on how much software you want to install along with the OS itself. Note that this disk space has to be available inside the host OS, not unpartitioned space.

Second, install VMWare on the host OS. This can be more complicated than it would first seem. VMWare has some strict requirements about displays that cause it not to work on some

systems. I couldn't get it to work on a Thinkpad connected to a ViewSonic flat panel display, for instance, and had irregular results on a couple of other machines.

VMWare comes with a number of standard 'vmmon' modules that are designed to work with many typical machine configurations. However, one effect of the exacting display requirements is that the vmmon modules that are included with the standard VMWare installation are not always sufficient to run VMWare on your machine. As a result, you may need to build a custom vmmon module that works specifically with your operating system. Fortunately, VMWare comes with tools in the installation process to easily build a new module.

Once installed, VMWare is ready for action. The third step is to create a virtual machine specifically configured for the guest operating system you want to install. When creating a virtual machine, you can choose which operating system you want to install in it, how much disk space to reserve for the virtual machine, how much RAM of the computer to reserve for the virtual machine, and some other parameters.

Once the virtual machine is installed, you still have to install the OS itself - the virtual machine is simply a shell that's customized for that type of OS. The final step is to insert the first installation CD for the guest operating system and start the virtual machine from within the VMWare interface. Doing so starts the installation of the guest OS within the VMWare interface. VMWare has a mechanism to easily and reliably capture screenshots during the installation.

After the guest OS is installed, you can use it inside of VMWare just like it was the only OS on the machine.

Installing VMWare on the Host OS

VMWare comes versions that can be installed on either Windows or Linux hosts. The Linux version comes in both a Compressed Tar Archive (25.0) MB and an RPM (25.2 MB). See Figure 1.

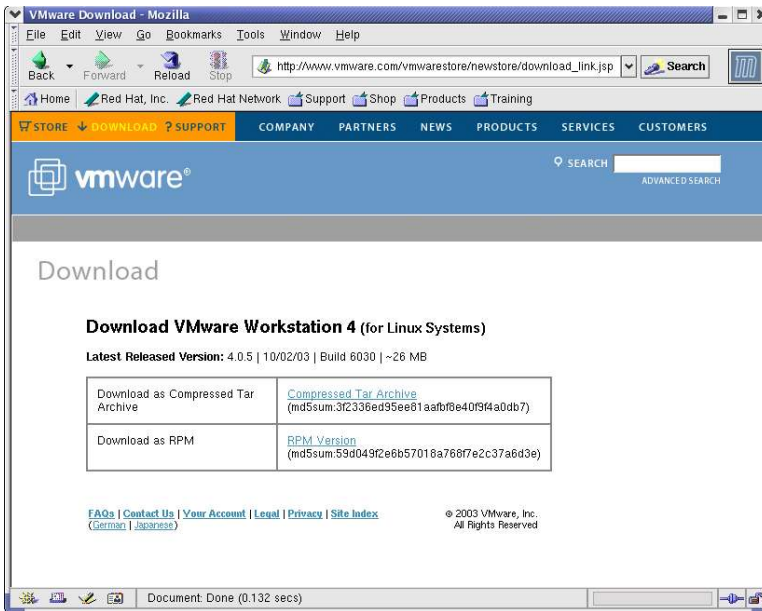


Figure 1. VMWare comes as a Compressed Tar archive and as an RPM.

1. Log on to your machine as a regular user.
2. Download VM ware onto the target machine from www.vmware.com.

As of this writing, the links to follow are Products, then VMWare Workstation. I downloaded both just to see the difference, and saved to . Saved to /home/alex

3. Download the PDF user manual.

Might as well do it now, so that you' rnot scrambling around trying to find it when you need to look up the answer to a question during the install or use.

4. Open Nautilus on your machine.
5. Double click on the .rpm file.

You' llbe prompted for root access.

5. Once you provide the root password, it' lldqueue the RPM for installation and you' llget the following dialog. See Figure 2.



Figure 2. VMWare Workstation queued for installation.

6. Click on Continue.

The installation process will run and you'll see a thermo bar in the progress dialog as shown in Figure 3.

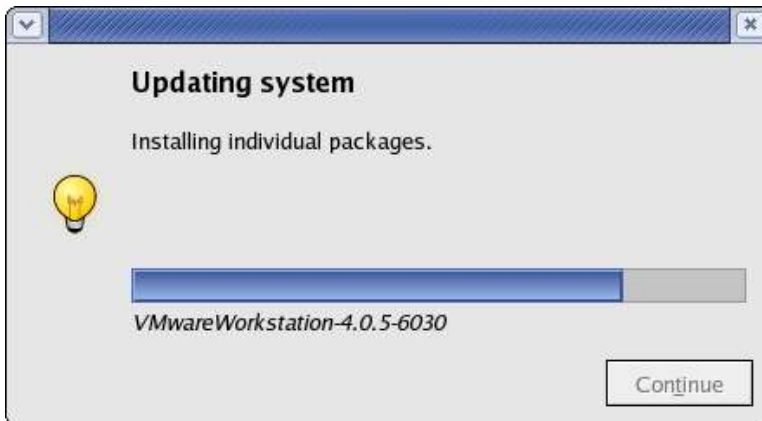


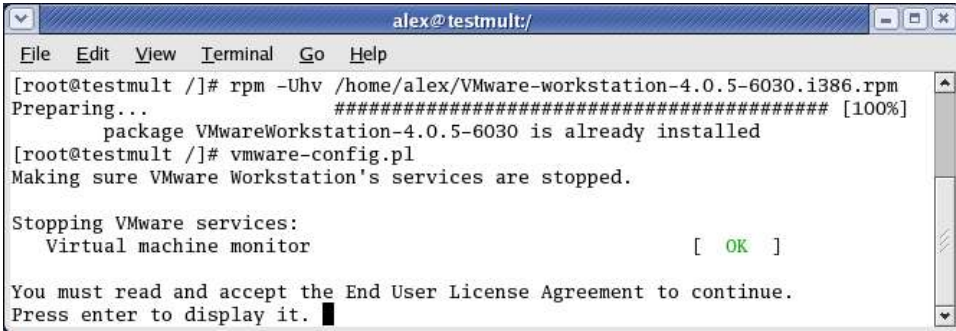
Figure 3. The VMWare installation process executing.

Note: Once the installation is complete, nothing 'happens' In other words, no completion message displays, no application dialog is launched. So now what? At first I didn't think it was installed, so I ran the rpm command from a terminal window, as shown in Figure 4. It told me that VMWare was already installed, so I reluctantly pulled out the documentation and found the next step was to run the config script, as described in Step 7.

7. Open a terminal window and become root:

8. Run the `vmware-config.pl` script, as shown in the next line in Figure 4.

The script stops the VMWare services, then displays the prompt regarding the EULA (this is commercial software, remember).



```
alex@testmult:/
File Edit View Terminal Go Help
[root@testmult /]# rpm -Uhv /home/alex/VMware-workstation-4.0.5-6030.i386.rpm
Preparing... ##### [100%]
package VMwareWorkstation-4.0.5-6030 is already installed
[root@testmult /]# vmware-config.pl
Making sure VMware Workstation's services are stopped.

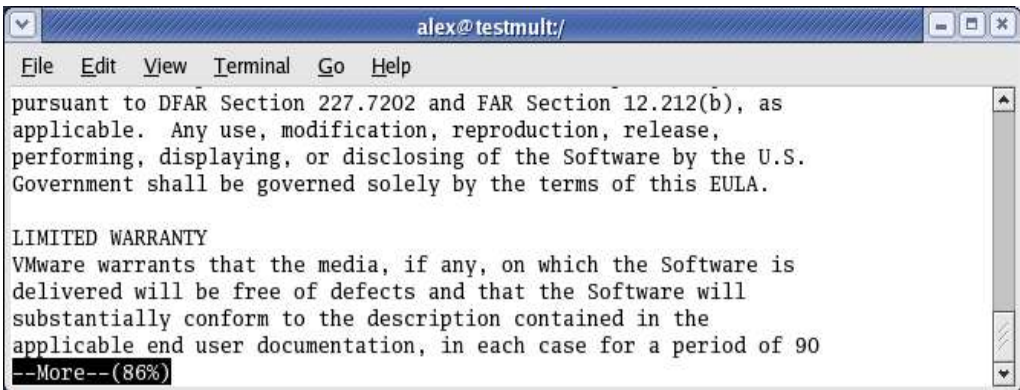
Stopping VMware services:
  Virtual machine monitor          [ OK ]

You must read and accept the End User License Agreement to continue.
Press enter to display it. █
```

Figure 4. Running the `vmware-config.pl` script from a terminal window.

9. Advance through the EULA as shown in Figure 5.

You can press the Enter key to read the EULA line by line or the space bar to read a screen full at a time. (You do read these word for word, don't you?)



```
alex@testmult:/
File Edit View Terminal Go Help
pursuant to DFAR Section 227.7202 and FAR Section 12.212(b), as
applicable. Any use, modification, reproduction, release,
performing, displaying, or disclosing of the Software by the U.S.
Government shall be governed solely by the terms of this EULA.

LIMITED WARRANTY
Vmware warrants that the media, if any, on which the Software is
delivered will be free of defects and that the Software will
substantially conform to the description contained in the
applicable end user documentation, in each case for a period of 90
--More-- (86%)
```

Figure 5. The space bar scrolls the EULA a screen at a time.

10. Accept or decline the EULA at the "Do you accept?" prompt, as shown in Figure 6.

When I installed VMWare on a SuSE 9.0 Personal host, the Enter key and the space bar scrolled through the EULA as it did on Fedora Core, but Konsole would not display the "Do you accept?" prompt. The shells for other GUIs included with SuSE wouldn't display the "Do you accept?" prompt either. I was forced to abandon SuSE as a host operating system.

11. If the `vmmon` module included with VMWare isn't suitable for your machine and host operating system, you'll get a message as shown in the middle of Figure 6. (If the `vmmon` module is OK, you can go on to Step 12.)

11a. The vmmon modules included with VMWare aren't suitable for Fedora Core 1, I had to build it myself. This was a bit trying. First, the script asked if it can try to build the module itself using information from the currently running kernel, as shown in the middle of Figure 6.

11b. I answered yes, and VMWare gave it the old college try.

11c. VMWare's attempt was unsuccessful, and it displays an obscure error message about versions of gcc, as shown in the bottom of Figure 6.

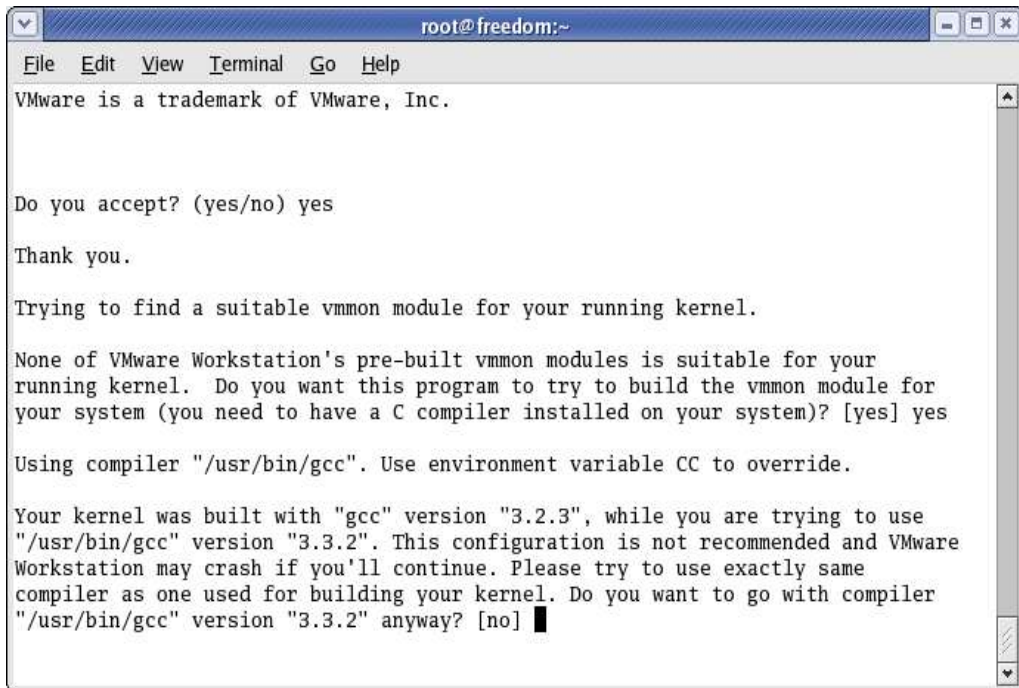


Figure 6. After accepting the EULA, VMWare goes to work on the vmmon module.

11d. You may find that the error message in Figure 8 is obscure. It took me a couple of times to read through it to catch on. Here's the deal. The current box, FC1, has the following versions of gcc installed:

| | | |
|--------|-------------|-------------|
| gcc | Oct 23 2003 | 83668 bytes |
| gcc296 | Feb 11 2003 | 83216 bytes |
| gcc32 | June 2 2003 | 79452 bytes |

The kernel on the host operating system was built using gcc 3.2.3 (which is in the file gcc32). However, when accepting ' yes' in the middle of Figure 6, VMWare tried to use gcc, which is a newer version - gcc 3.3.2. So VMWare's vmmon module would be built with a different version of gcc, and that might not work with the kernel. While you're given the choice to try anyway, it's probably not a good idea. (Actually, it's definitely not a good idea - I tried it just for fun, and it didn't work.)

So what do you do? There are several options.

First, ignore the warning, and try to build with the newer (incompatible) version. As I mentioned, I tried that and it didn't wrk. Your mileage may vary, but it probably won't. So answer 'no' and get some suggestion for help, as shown in Figure 7.

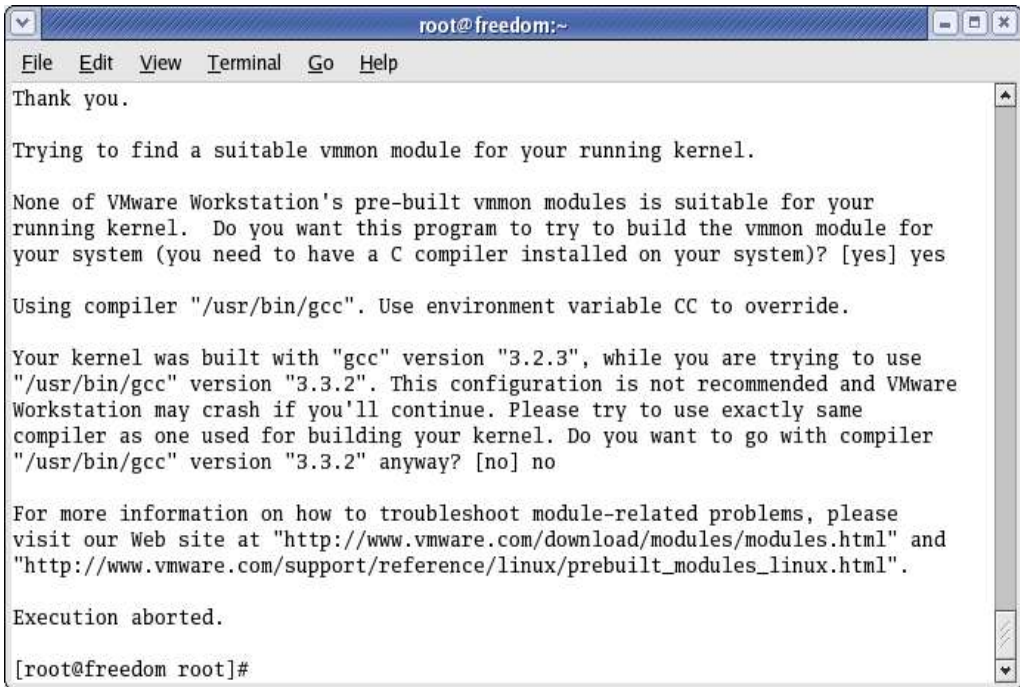


Figure 7. Information on troubleshooting module-related problems.

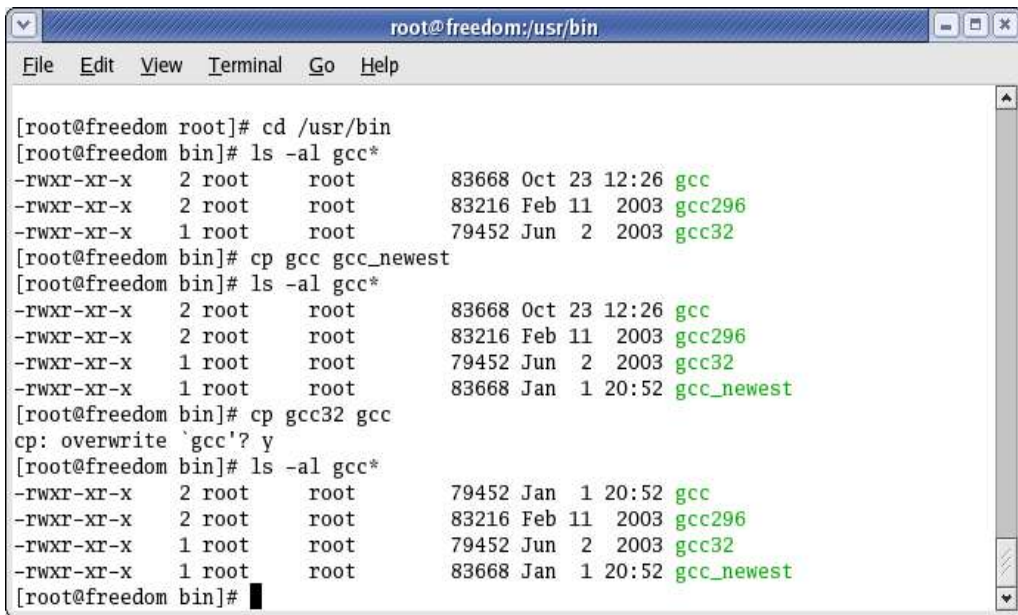
11e. So the second option is to use the correct version of gcc. You need to tell it to use the file gcc32 when building vmmon. How?

You could use the environment variable CC, or you could call gcc32 manually. I opted for the third method - to copy gcc32 on top of gcc temporarily.

Specifically, I renamed gcc to gcc_newest, copied gcc32 to gcc so it (the correct, compatible, albeit older version of gcc) will be called by VMWare, do the build, then replace gcc with the original again. The following code performs these steps except for the final replace.

```
cd /usr/bin
cp gcc gcc_newest
cp gcc32 gcc
```

The results are shown in Figure 8.



```
root@freedom:/usr/bin
File Edit View Terminal Go Help

[root@freedom root]# cd /usr/bin
[root@freedom bin]# ls -al gcc*
-rwxr-xr-x  2 root    root      83668 Oct 23 12:26 gcc
-rwxr-xr-x  2 root    root      83216 Feb 11  2003 gcc296
-rwxr-xr-x  1 root    root      79452 Jun  2  2003 gcc32
[root@freedom bin]# cp gcc gcc_newest
[root@freedom bin]# ls -al gcc*
-rwxr-xr-x  2 root    root      83668 Oct 23 12:26 gcc
-rwxr-xr-x  2 root    root      83216 Feb 11  2003 gcc296
-rwxr-xr-x  1 root    root      79452 Jun  2  2003 gcc32
-rwxr-xr-x  1 root    root      83668 Jan  1 20:52 gcc_newest
[root@freedom bin]# cp gcc32 gcc
cp: overwrite `gcc'? y
[root@freedom bin]# ls -al gcc*
-rwxr-xr-x  2 root    root      79452 Jan  1 20:52 gcc
-rwxr-xr-x  2 root    root      83216 Feb 11  2003 gcc296
-rwxr-xr-x  1 root    root      79452 Jun  2  2003 gcc32
-rwxr-xr-x  1 root    root      83668 Jan  1 20:52 gcc_newest
[root@freedom bin]#
```

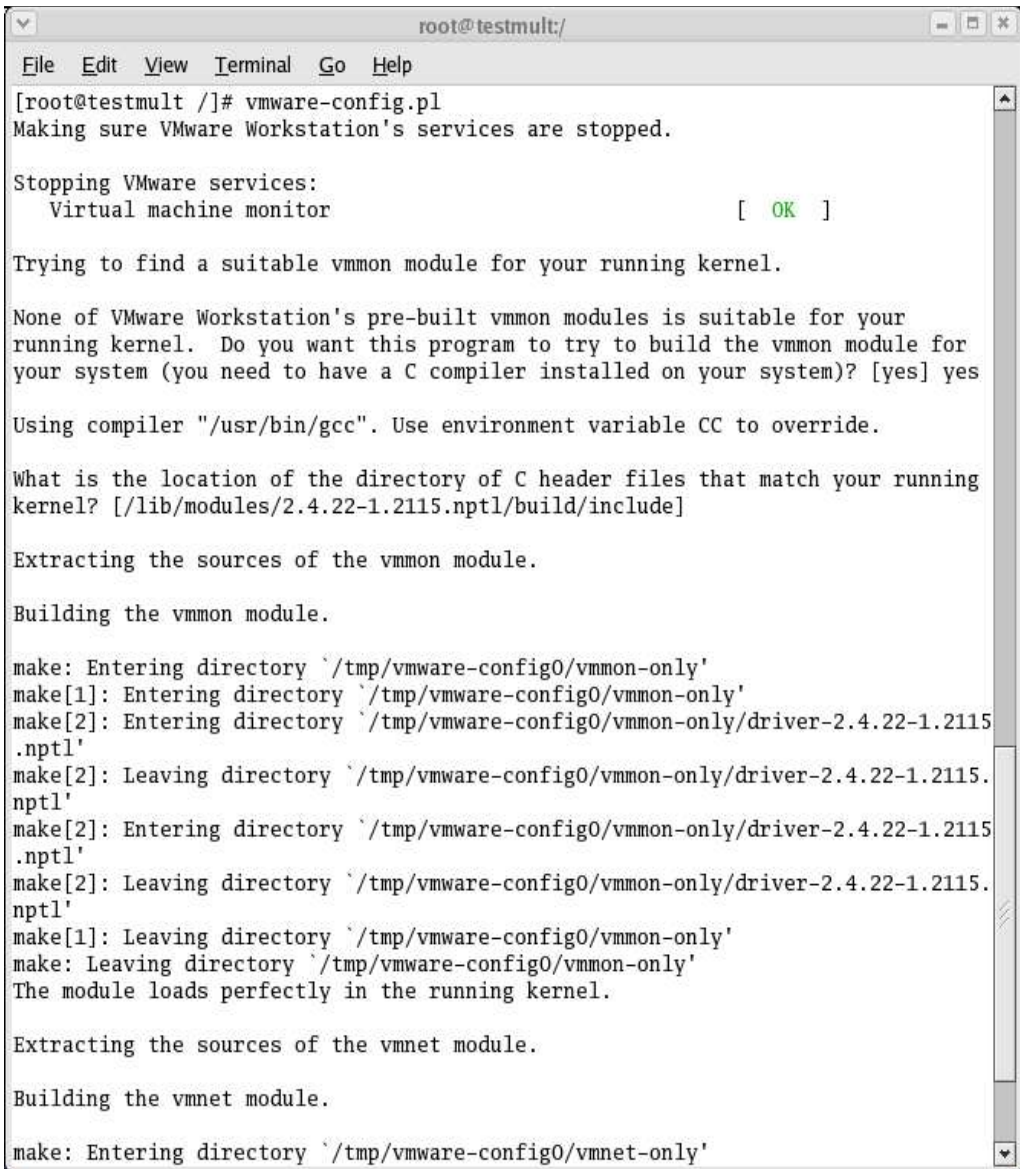
Figure 8. Temporarily replacing the current version of gcc with an older one for the vmmon build.

Now we're ready to try to build vmmon again.

11f. Enter

vmware-config.pl

as super user, as shown in Figure 9.



```

root@testmult:/
File Edit View Terminal Go Help
[root@testmult /]# vmware-config.pl
Making sure VMware Workstation's services are stopped.

Stopping VMware services:
  Virtual machine monitor          [ OK ]

Trying to find a suitable vmmon module for your running kernel.

None of VMware Workstation's pre-built vmmon modules is suitable for your
running kernel. Do you want this program to try to build the vmmon module for
your system (you need to have a C compiler installed on your system)? [yes] yes

Using compiler "/usr/bin/gcc". Use environment variable CC to override.

What is the location of the directory of C header files that match your running
kernel? [/lib/modules/2.4.22-1.2115.nptl/build/include]

Extracting the sources of the vmmon module.

Building the vmmon module.

make: Entering directory `/tmp/vmware-config0/vmmon-only'
make[1]: Entering directory `/tmp/vmware-config0/vmmon-only'
make[2]: Entering directory `/tmp/vmware-config0/vmmon-only/driver-2.4.22-1.2115
.nptl'
make[2]: Leaving directory `/tmp/vmware-config0/vmmon-only/driver-2.4.22-1.2115
.nptl'
make[2]: Entering directory `/tmp/vmware-config0/vmmon-only/driver-2.4.22-1.2115
.nptl'
make[2]: Leaving directory `/tmp/vmware-config0/vmmon-only/driver-2.4.22-1.2115
.nptl'
make[1]: Leaving directory `/tmp/vmware-config0/vmmon-only'
make: Leaving directory `/tmp/vmware-config0/vmmon-only'
The module loads perfectly in the running kernel.

Extracting the sources of the vmnet module.

Building the vmnet module.

make: Entering directory `/tmp/vmware-config0/vmnet-only'

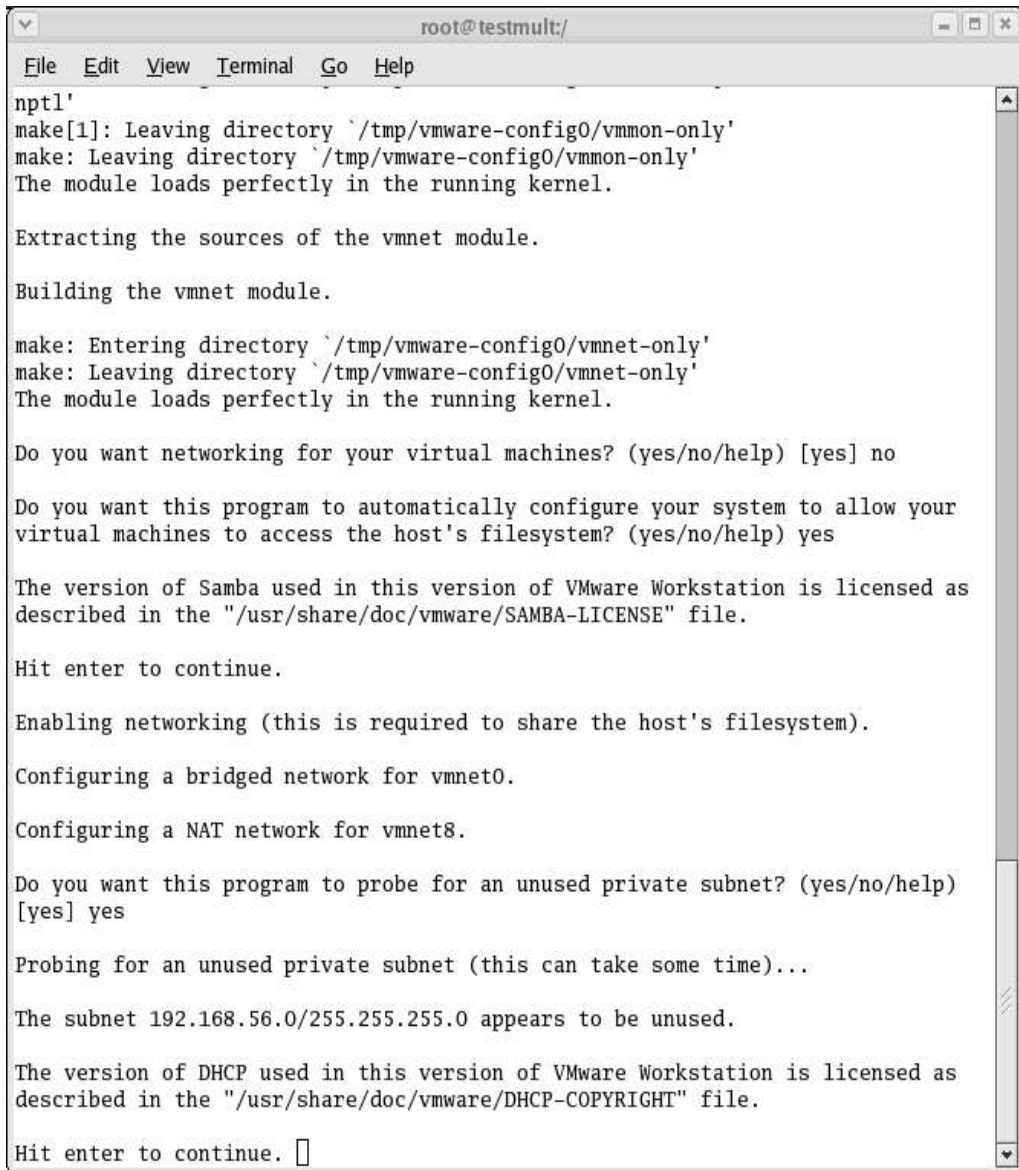
```

Figure 9. Building the vmmon module with the correct version of gcc.

11g. You'll see that this time it tells you to use /usr/bin/gcc and launches right into asking about the C header files - no more business about gcc versions not matching, so it must have accepted my little renaming trick.

11h. Press Enter at the prompt for the C header files (I didn't type the location), and off it will go. The entire process took about 30 seconds on a 1.4 GH machine with a GB of RAM. The screen in Figure 10 shows the completion of the build process (through the "The module

is running perfectly" line), and continues on with the configuration steps also executed if this step, building a new vmmon module, wasn't required.

A screenshot of a terminal window titled 'root@testmult:/'. The window contains the following text:

```
nptl'  
make[1]: Leaving directory `/tmp/vmware-config0/vmmon-only'  
make: Leaving directory `/tmp/vmware-config0/vmmon-only'  
The module loads perfectly in the running kernel.  
  
Extracting the sources of the vmnet module.  
  
Building the vmnet module.  
  
make: Entering directory `/tmp/vmware-config0/vmnet-only'  
make: Leaving directory `/tmp/vmware-config0/vmnet-only'  
The module loads perfectly in the running kernel.  
  
Do you want networking for your virtual machines? (yes/no/help) [yes] no  
  
Do you want this program to automatically configure your system to allow your  
virtual machines to access the host's filesystem? (yes/no/help) yes  
  
The version of Samba used in this version of VMware Workstation is licensed as  
described in the "/usr/share/doc/vmware/SAMBA-LICENSE" file.  
  
Hit enter to continue.  
  
Enabling networking (this is required to share the host's filesystem).  
  
Configuring a bridged network for vmnet0.  
  
Configuring a NAT network for vmnet8.  
  
Do you want this program to probe for an unused private subnet? (yes/no/help)  
[yes] yes  
  
Probing for an unused private subnet (this can take some time)...  
  
The subnet 192.168.56.0/255.255.255.0 appears to be unused.  
  
The version of DHCP used in this version of VMware Workstation is licensed as  
described in the "/usr/share/doc/vmware/DHCP-COPYRIGHT" file.  
  
Hit enter to continue. ☐
```

Figure 10. VMWare prompts for networking requirements.

12. Answer a number of configuration questions to complete the installation of VMWare.

The second half of Figure 10 and Figures 11 and 12 show the prompts and my answers. As I primarily wanted to install VMWare so that I could grab screen shots, I didn't have a lot

of networking requirements. You may want to make different choices depending on your situation. The warning about 'this can take some time' ended up needing about 5 seconds on my ten machine network.



```

root@testmult:/
File Edit View Terminal Go Help
described in the "/usr/share/doc/vmware/DHCP-COPYRIGHT" file.

Hit enter to continue.

Creating a host-only network on vmnet1. (this is required to share the host's
filesystem).

Configuring a host-only network for vmnet1.

Do you want this program to probe for an unused private subnet? (yes/no/help)
[yes] yes

Probing for an unused private subnet (this can take some time)...

The subnet 192.168.128.0/255.255.255.0 appears to be unused.

This system appears to have a CIFS/SMB server (Samba) configured for normal use.
If this server is intended to run, you need to make sure that it will not
conflict with the Samba server setup on the private network (the one that we use
to share the host's filesystem). Please check your /etc/samba/smb.conf file so
that:

. The "interfaces" line does not contain "192.168.128.1/255.255.255.0"
. There is a "socket address" line that contains only your real host IP address

Hit enter to continue.

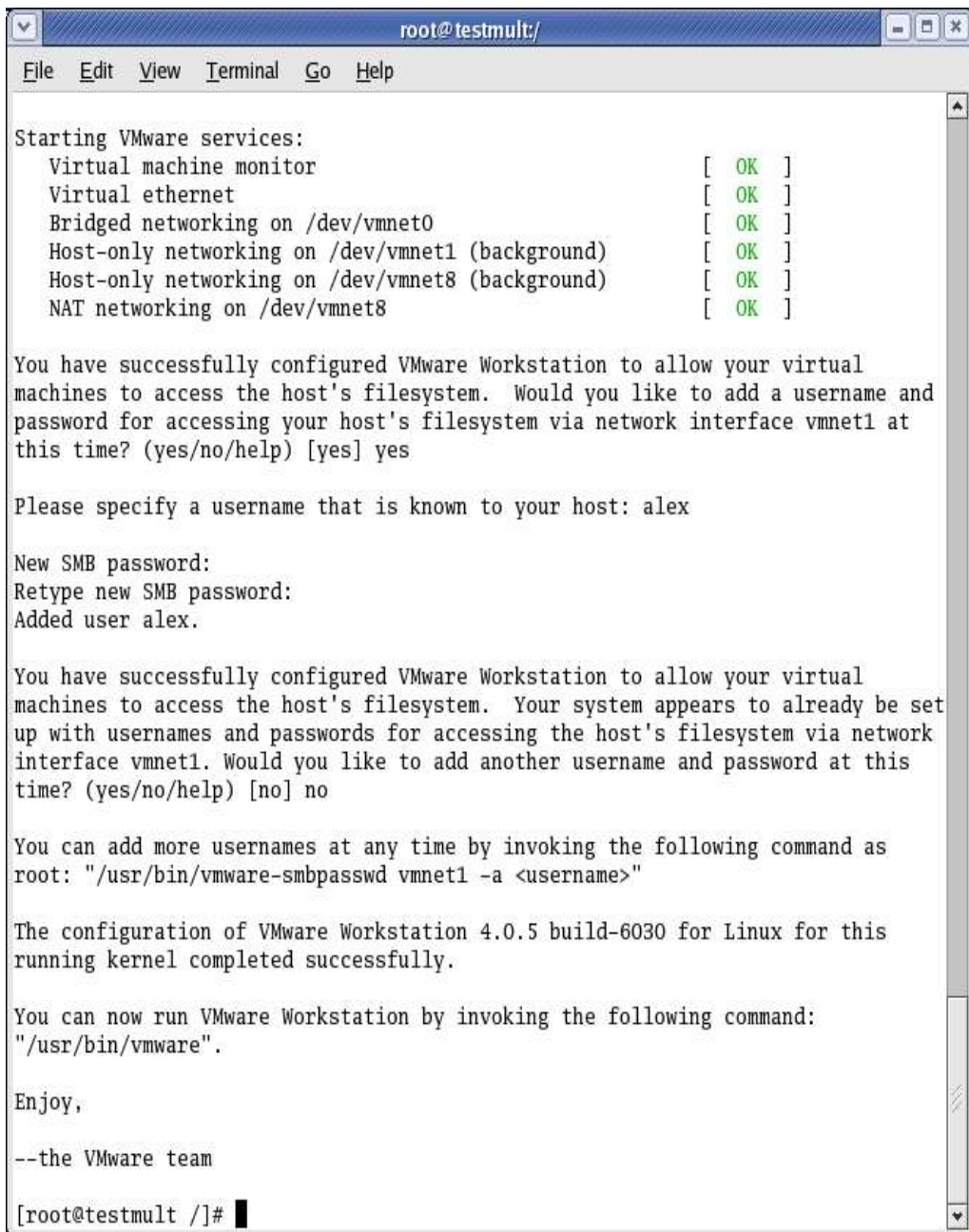
Starting VMware services:
Virtual machine monitor          [ OK ]
Virtual ethernet                 [ OK ]
Bridged networking on /dev/vmnet0 [ OK ]
Host-only networking on /dev/vmnet1 (background) [ OK ]
Host-only networking on /dev/vmnet8 (background) [ OK ]
NAT networking on /dev/vmnet8    [ OK ]

You have successfully configured VMware Workstation to allow your virtual
machines to access the host's filesystem. Would you like to add a username and
password for accessing your host's filesystem via network interface vmnet1 at
this time? (yes/no/help) [yes]

```

Figure 11. VMWare prompts for subnet requirements.

I've installed VMWare about a half dozen times, and each time the "interfaces" line address has never conflicted with the subnet VMWare grabbed.



```
root@testmult:/
File Edit View Terminal Go Help

Starting VMware services:
  Virtual machine monitor          [ OK ]
  Virtual ethernet                 [ OK ]
  Bridged networking on /dev/vmnet0 [ OK ]
  Host-only networking on /dev/vmnet1 (background) [ OK ]
  Host-only networking on /dev/vmnet8 (background) [ OK ]
  NAT networking on /dev/vmnet8    [ OK ]

You have successfully configured VMware Workstation to allow your virtual
machines to access the host's filesystem. Would you like to add a username and
password for accessing your host's filesystem via network interface vmnet1 at
this time? (yes/no/help) [yes] yes

Please specify a username that is known to your host: alex

New SMB password:
Retype new SMB password:
Added user alex.

You have successfully configured VMware Workstation to allow your virtual
machines to access the host's filesystem. Your system appears to already be set
up with usernames and passwords for accessing the host's filesystem via network
interface vmnet1. Would you like to add another username and password at this
time? (yes/no/help) [no] no

You can add more usernames at any time by invoking the following command as
root: "/usr/bin/vmware-smbpasswd vmnet1 -a <username>"

The configuration of VMware Workstation 4.0.5 build-6030 for Linux for this
running kernel completed successfully.

You can now run VMware Workstation by invoking the following command:
"/usr/bin/vmware".

Enjoy,

--the VMware team

[root@testmult /]#
```

Figure 12. VMWare prompts for SMB data.

The SMB password does not have to be the same as the user's regular password - purists would argue that it should be different. For those who are just using this as a demo machine, it's probably more work to create and remember a separate SMB password. It's your call.

And we're done - back at the shell prompt - and ready to run VMWare. you're ready to create your first virtual machine.

Creating a new Virtual Machine ("VM")

The next step is to create a new virtual machine. Log on to your machine as the user who will be using VMWare (not as root). Then, to start VMWare, type

```
vmware &
```

into a terminal window. You'll get the main VMWare window as shown in Figure 13.

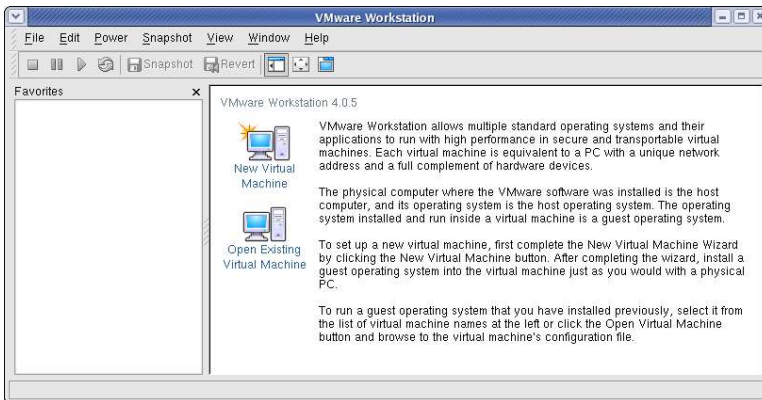


Figure 13. The main VMWare interface window allows you to create new virtual machines as well as use existing virtual machines.

Click on the New Virtual Machine icon to start the New Virtual Machine wizard as shown in Figure 14.



Figure 14. The first step in the New Virtual Machine Wizard is to select the type of configuration.

When faced with the "typical" vs. "custom" choices, I always choose the custom choice, even if I don't think I'll make any custom selections, just so I can see what the potential alternatives are going to be.

The next step is to select the Guest Operating System as shown in Figure 15. The Guest Operating System is the one you're going to install inside the new virtual machine you're creating. It's interesting that there are about a dozen choices for various flavors of Windows while just one choice for Linux, regardless of which distribution you plan on installing.



Figure 15. The second step in the New Virtual Machine Wizard is to select the guest operating system.

The third step is to select a name for the virtual machine, and the location on disk for the files that will contain the virtual machine, as shown in Figure 16.



Figure 16. The third step in the New Virtual Machine Wizard is to select the location of the files that will contain the virtual machine.

If you intend on creating more than one virtual machine for the same guest operating system (say, one VM for SuSE and another for Mandrake), you should use a more descriptive name for the virtual machine. Else, you'll end up with a list of virtual machines that all say "linux".

Similarly, consider where you want the virtual machine files stored. A separate directory will be created automatically for each virtual machine, and the default location is under the user's home directory.

I use Nautilus to navigate through my file system a lot, and so I like to keep my home directory clear of stuff I'm not going to navigate through. Since the virtual machine directories don't contain files that I'm going to manipulate directly, I renamed the `/home/whil/linux` directory to `/home/whil/zzzlinux` so that it falls to the bottom of the tree view in Nautilus - out of sight and out of bother. Your preferences may differ, of course.

The fourth step in the New Virtual Machine Wizard is to configure memory, as shown in Figure 17. VMWare likes lots of memory, so the more RAM your physical machine has, the better a time you'll have with VMWare.

VMWare reserves a chunk of the available memory for each virtual machine that is running - and the more memory that can be allotted to a virtual machine, the better it will perform. Obviously, you can't reserve more memory than is available on the entire machine, and you'll need to leave some for the host operating system. The amount of memory you allocate to a virtual machine depends on (1) as well as how much memory is available on the machine, (2) the requirements of the guest OS you're installing in the virtual machine, and (3) your tolerance for pokey performance.

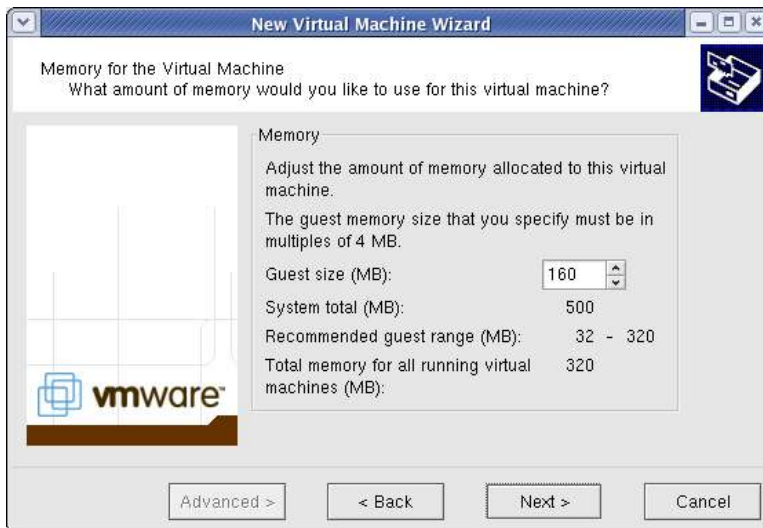


Figure 17. The fourth step in the New Virtual Machine Wizard is to select how much memory to allocate to the virtual machine.

I've found 160 MB of RAM just for an installation of Fedora Core was tolerable, but I wouldn't want to use it much with that little.

The fifth step in the New Virtual Machine Wizard is to choose the type of networking, as shown in Figure 18.

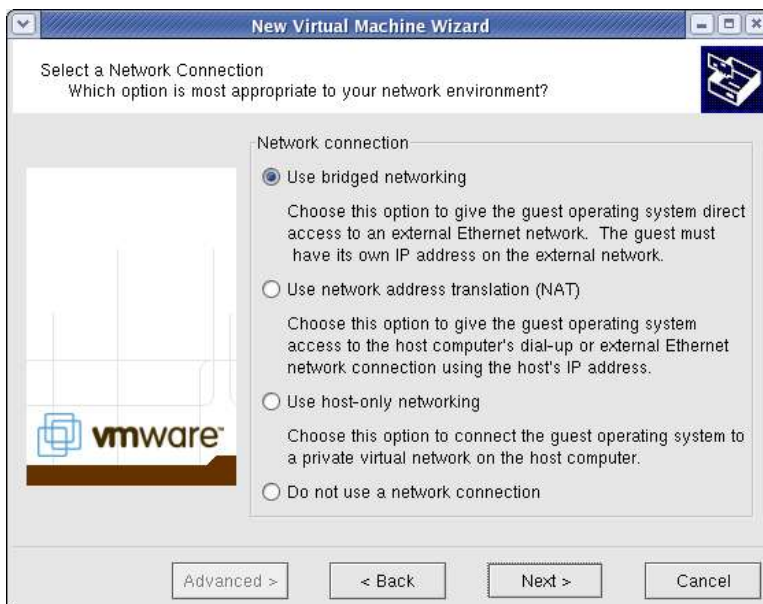


Figure 18. The fifth step in the New Virtual Machine Wizard is to choose the type of networking.

Your personal configuration requirements will drive the choice you make here.

The sixth step is to choose which disk should be used by the virtual machine as shown in Figure 19.



Figure 19. The sixth step in the New Virtual Machine Wizard is to choose the type of networking.

Since this is your first shot, keep the default of "Create a new virtual disk." The seventh step is to choose disk capacity for this virtual machine, as shown in Figure 20.



Figure 20. The seventh step in the New Virtual Machine Wizard is to choose the disk capacity.

You need to specify a minimum size that will allow the guest operating system to be installed and run, including space for temporary files. For Fedora Core, with a minimum install, I've found 4 gigabytes was enough.

While the wizard will allow you to specify a disk size that is larger than the available disk space on the hard disk, if VMWare runs out of room during installation or use of the guest operating system, it will simply die a hot white death and you'll be stuck with reconfiguring the physical machine to allow the host operating system to access more disk space (if it's available) or to find another machine with more disk space.

The final step is to specify the name of the file that will hold the virtual disks themselves, shown in Figure 21.

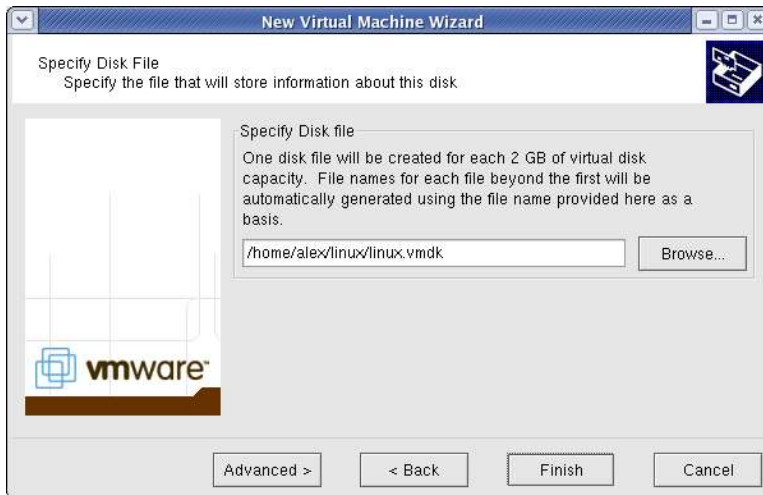


Figure 21. The last step in the New Virtual Machine Wizard is to choose the disk file names.

Clicking "Finish" will create the virtual machine and then display the main virtual machine screen as shown in Figure 22. If you create more than one VM on this computer, you'll see nodes for each VM in the tree view on the left, with the name of the virtual machine chosen in Step 3 of the wizard (see Figure 16.)

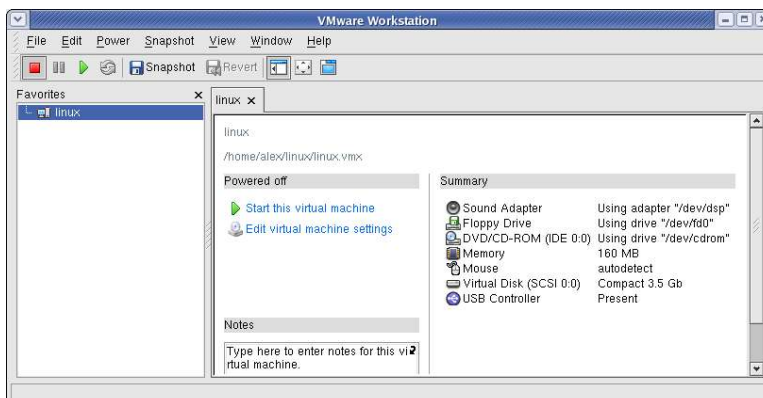


Figure 22. The virtual machine main screen.

It's worth a moment to point out a few features of the virtual machine main screen.

Components

There are four primary components - the main menu, the toolbar, the Favorites window and the Virtual Machine window. The main menu and toolbar operate just like in other standard GUI applications while the Favorites and Virtual Machine windows have specialized purposes.

Main Menu

The main menu has all of the standard pads you'd expect in a GUI application, including File, Edit, View, Window and Help, and the menu options underneath them are pretty much what you'd expect as well.

The Power menu allows you to turn a virtual machine on or off. The Snapshot menu doesn't have anything to do with screen shots; rather, it has options to take a 'snapshot' of the current state of the virtual machine. This is useful for being able to roll back to a given state, say, before a particular Windows service pack was installed or a Linux kernel was upgraded, or an application was installed.

The File | Capture Screen menu option is where you can grab screen shots of your Linux installation.

Toolbar

The toolbar provides buttons for some of the functions available on the main menu; namely Power Off, Suspend, Power On, and taking and reverting to a snapshot.

Favorites window

The Favorites window displays a list of all virtual machines installed in VMWare, Virtual machines that are running have a small green arrow included as part of the icon that is left of the virtual machine's name in the list.

Virtual Machine window

The Virtual Machine window is the window that displays the guest operating system. If more than one virtual machine is running at the same time, the tabs on the top of the Virtual Machine window identifies the name of the guest operating system.

Activation

Before you start using a virtual machine to install a guest operating system, though, you'll need to activate your copy of VMWare if you haven't already. You can actually do this as soon as you've installed VMWare if you choose, but you need to do it before using a virtual machine, even if you're running a demo copy of VMWare.

Log on to your machine as the user who will be using VMWare (not as root). Then, to start VMWare, type

```
vmware &
```

into a terminal window. You'll get the main VMWare window as shown in Figure 13.

Click on Help | Enter Serial Number as shown in Figure 23.

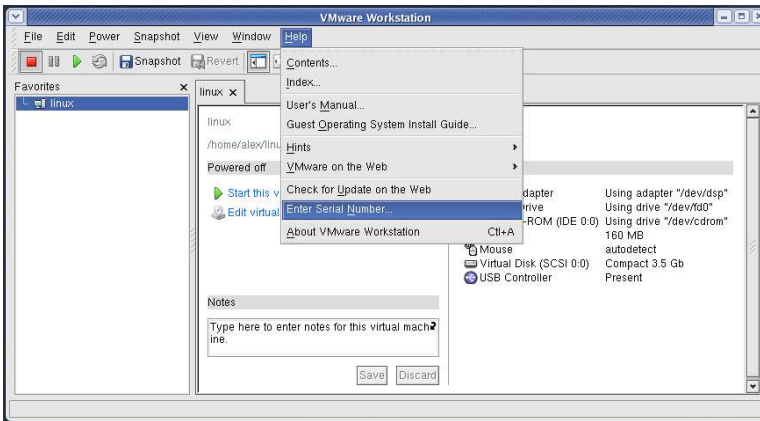


Figure 23. Entering the VMWare serial number.

Once you've entered a valid serial number, the confirmation dialog shown in Figure 24 will display.



Figure 24. Serial number entry confirmation dialog.

Now click OK and you'll be ready to use a VMWare virtual machine.

Using VMWare to install a second copy of an OS

The next step is a little tricky because the VMWare interface lets you do a bunch of things, but there's really only one function you should be performing.

You've got a virtual machine installed, but it's an empty shell - there's nothing in it. It's just like having a machine without an operating system on it - which, after all, is the situation we've got in our virtual machine. You want to install your guest operating system - in this case, a second copy of Fedora Core - into this empty virtual machine shell.

Do this by putting the first installation CD for the guest operating system you're installing into the computer's CD-ROM drive (yes, the physical drive).

Then, power on the virtual machine, in one of three ways:

1. Select the Power | Power On menu option,

2. Click the Power on button (the green right-pointing arrow) in the toolbar, or
 3. Select the the Start the Virtual Machine link in the main window.
- See Figure 25 to see where these items are.

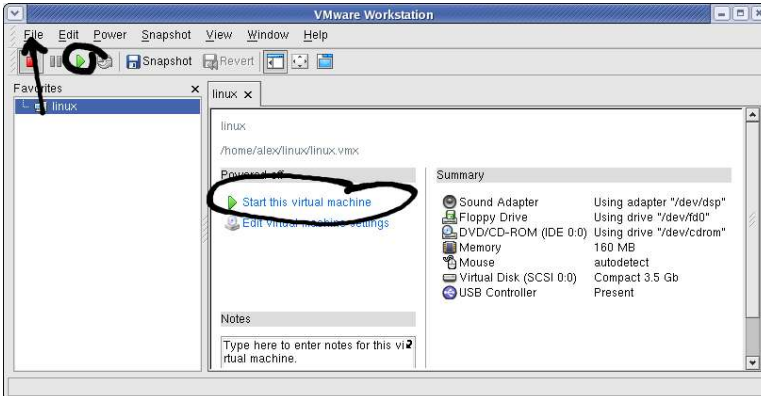


Figure 25. Three different ways to power on a virtual machine.

Performing any of these three actions is the software equivalent of having a CD in a box that's off, and then turning the box on. The virtual machine really acts just like a physical machine in this regard; if you attempt to power on a virtual machine without a bootable CD in the CD-ROM drive, VMWare will complain, as shown in Figure 26.

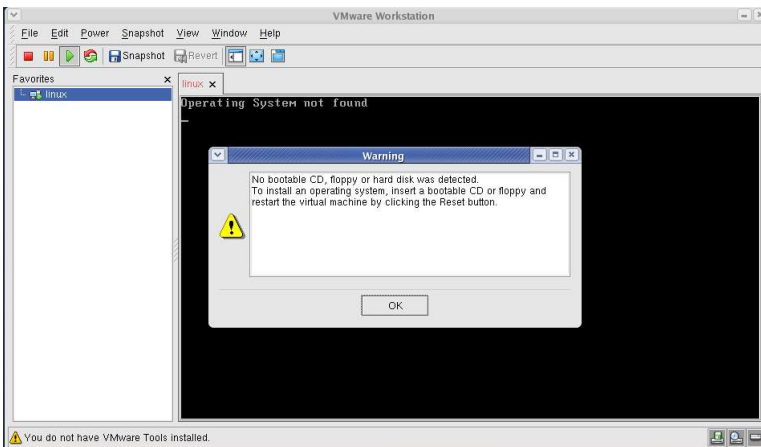


Figure 26. Powering on a virtual machine without a bootable CD generates an error.

Once you've powered on the virtual machine (and there's a bootable CD in the drive), the entire installation process will display in the virtual machine window. The first thing you'll see in the virtual machine window will be the VMWare log, as shown in Figure 27.

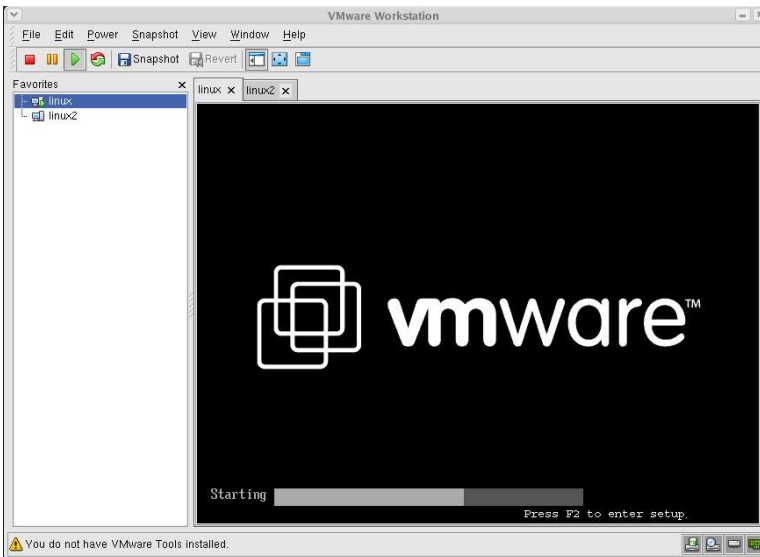


Figure 27. Powering on a virtual machine starts with the display of the VMWare logo.

Shortly thereafter, the boot screen of the Fedora Core installation process will display, as shown in Figure 28.

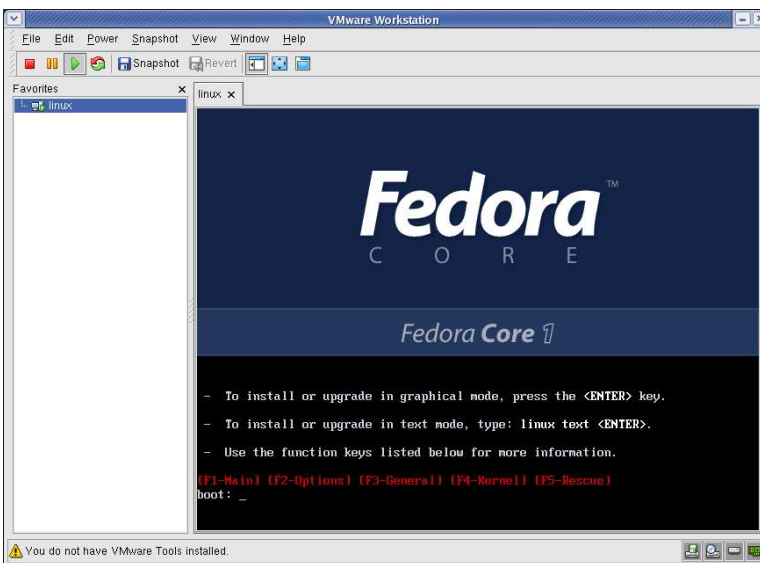


Figure 28. The Fedora Core installation process in a virtual machine starts with the standard boot screen.

Pressing Enter at this point runs the media test, as shown in Figure 29.

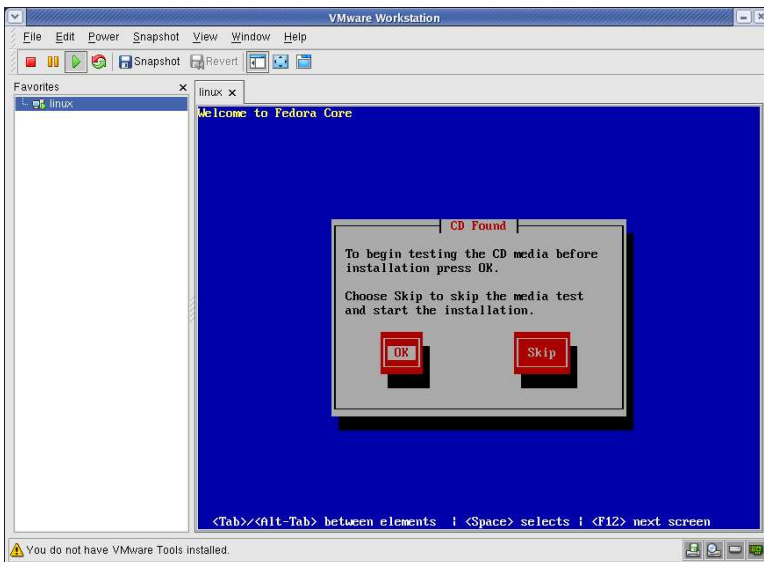


Figure 29. The media test is the second step in the Fedora Core installation process.

At this point, it's important to point out one key operational technique. The mouse doesn't work in the media test screen - you have to use the keyboard to arrow back and forth between the two choices.

Once you're done with the media test, the installation process continues as shown in Figure 30 as anaconda, the Fedora Core system installer, begins checking the hardware.

Note: The screens in Figures 30, 31 and 32 are all still displayed in the virtual machine window in the virtual machine main screen, as the media test screen was shown in Figure 29. They're simply shown full screen in these figures for better detail.

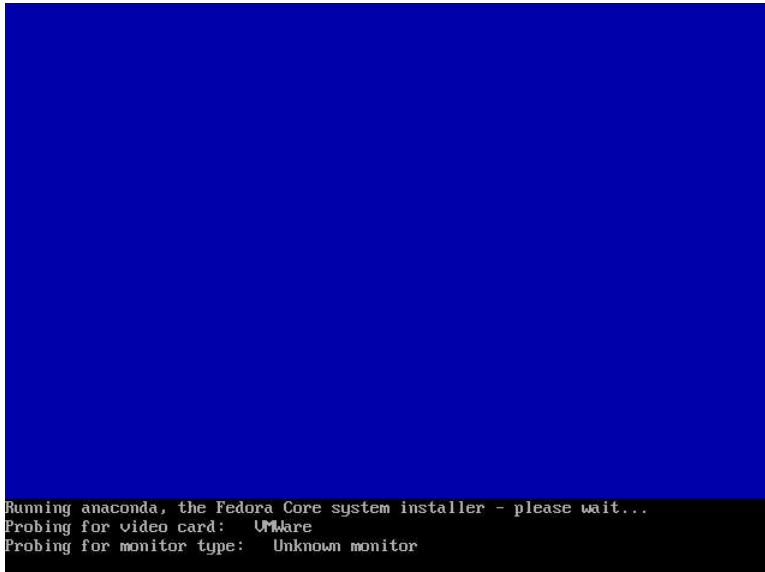


Figure 30. Once the media test screen has been passed, anaconda starts probing for hardware.

If the hardware check goes well, the Fedora Core logo screen displays, which tells you that X has started successfully, and you' re going to be able to use the mouse from now on.



Figure 31. Display of the Fedora Core logo screen means that X has started successfully and that the mouse will work from now on.

The Fedora Core logo screen will disappear after a few moments and the Welcome screen will take its place, as shown in Figure 32.

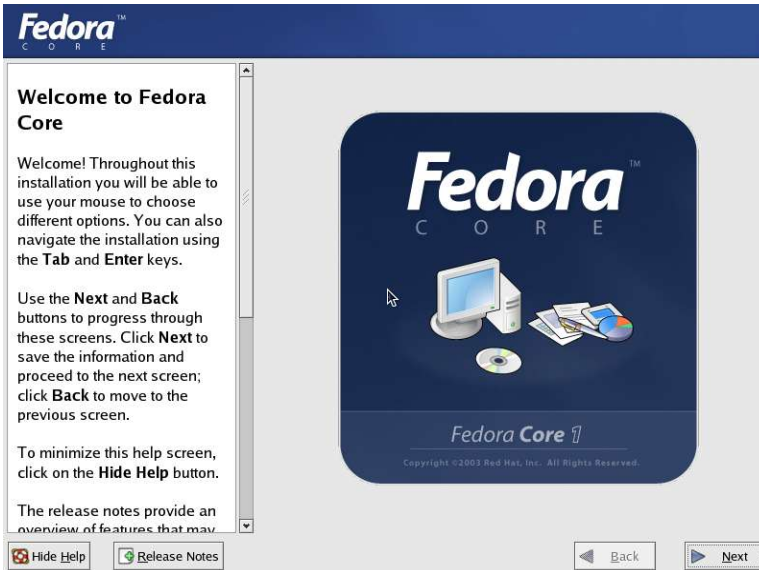


Figure 32. The Welcome screen of the Fedora Core installation routine.

At this point, the installation routine works the same as it normally does when Fedora Core is installed on a machine as the host operating system. What is different, however, is how the mouse interacts with the virtual machine and the VMWare virtual machine main screen, and this is true after installation during regular use of the guest operating system.

Working with VMWare

There are three interrelated topics involved in working with VMWare in order to make screen shots. I'll use a copy of Fedora Core running inside a VMWare virtual machine to demonstrate; the same techniques are applicable during the installation where you want to take screen shots.

Using a Mouse

During installation as well as during general use, the mouse is in one of two modes. The first mode is regular mode, where you can work with VMWare as well as other applications as usual, or in restricted mode. The mouse pointer is displayed as an arrow outside the virtual machine window, and a closed hand with the index finger extended when the mouse is moved over the virtual machine window. The only limitation is that you can't use the mouse to manipulate anything inside the virtual machine window in VMWare.

The other mode is restricted, where the mouse is limited to the borders of the virtual machine window inside the VMWare application - in other words, only with the guest operating system. The mouse pointer is displayed as an arrow.

To get from regular mode to restricted, so that you can manipulate the mouse in the guest operating system, move the mouse over the virtual machine window, and then click either the left or right mouse button. The mouse pointer will change to an arrow.

To get from restricted mode to regular again, so that you can work with the VMWare interface or another application, hold the Ctrl and Alt keys down at the same time. The mouse pointer will change to the hand. Note that when you change back to restricted mode, the mouse cursor will be back at the place it was when you left restricted mode, not where the mouse was when you re-entered restricted mode.

Taking screen shots

In order to take screen shots during installation, then, you're going to be flipping back and forth between regular and restricted modes. As you start up the installation process, you'll power on the virtual machine, and then enter restricted mode in order to run the install.

When it comes time to take the first screen shot, you'll press Ctrl-Alt in order to move into regular mode, and then select the File | Capture screen menu option. You'll be prompted to save the image as a PNG file as shown in Figure 33.

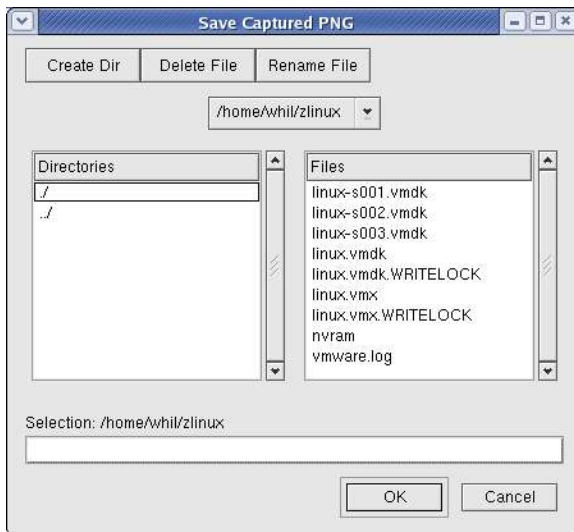


Figure 33. VMWare prompts you to save a captured screen shot in the virtual machine's home directory.

Unfortunately, the default location is the virtual machine's home directory. Depending on your work habits, you're probably going to want to save your image files somewhere else. This is particularly true if you're creating and deleting virtual machines regularly. When you delete a virtual machine, the directory is deleted as well - and so are all of your carefully captured screen shots!

I say 'unfortunately' in the previous paragraph because VMWare doesn't seem to be able to remember any other location, and it's a pain to have to navigate up and down directory trees to get them put into the directory of your choosing.

Another inconvenience I encountered when capturing screen shots is that the name is not automatically generated; in fact, it isn't even given a PNG extension. If you just type "image1" or "install_01" as the file name, you'll end up a bunch of extensionless files, which can cause problems for some applications.

Conclusion

I've just scratched the surface of VMWare's capabilities in order to show you how to grab screen shots during an operating system installation. VMWare is intended to be used not just for screen shot capturing, but for full use of an operating system when you don't have the capability or inclination to install multiple OSs on separate machines (or on the same machine.) The 300+ page manual is chock full of information I haven't even mentioned.

For updates to this whitepaper as well as other HOWTO whitepapers, please visit www.hentzenwerke.com.

Copyright 2004 Whil Hentzen. All rights reserved.