

CST8207 – Linux O/S I

Creating Filesystems

Tricks and traps

- ▶ The super user of Unix/Linux works differently from Windows
- ▶ With Windows, there is a concept of an “admin” user
- ▶ With windows, regular users can be made or enhanced to an “admin” user
- ▶ With Unix/Linux, there is one superuser named “root”

Unix/Linux superuser: root

- ▶ root is the name of the one and only superuser on Linux/Unix systems
- ▶ The home directory of the one user with the name “root” is “/root”
- ▶ The root filesystem starts at what we call “the root directory”

More root...

“the root directory” is not the same concept as “root’s home directory”

The root directory is “/”

“root’s home directory” is “the home directory of the user named ‘root’” and it is usually “/root”

Sudo (better than “log in as root”)

- ▶ `/etc/sudoers` is a file that specifies which users can “become root” this way
- ▶ A good way to obtain a root prompt:
`sudo -s`
- ▶ The user who runs the command must be in `/etc/sudoers`
- ▶ The user who runs the command will be prompted for their own password

Putting your username in /etc/sudoers

- ▶ Log in as your regular user
- ▶ Open a terminal
- ▶ Become the root user with “su -l”
- ▶ You should have a root prompt #
- ▶ Type the command

`VISUAL=gedit visudo`

A text editor window will appear, go to the bottom of the file

/etc/sudoers (contd)

Add the line

```
yourownuser    ALL=(ALL)    ALL
```

to the bottom of the file, where you PUT YOUR OWN USERNAME instead of “yourownuser”

Save the file

Exit the editor

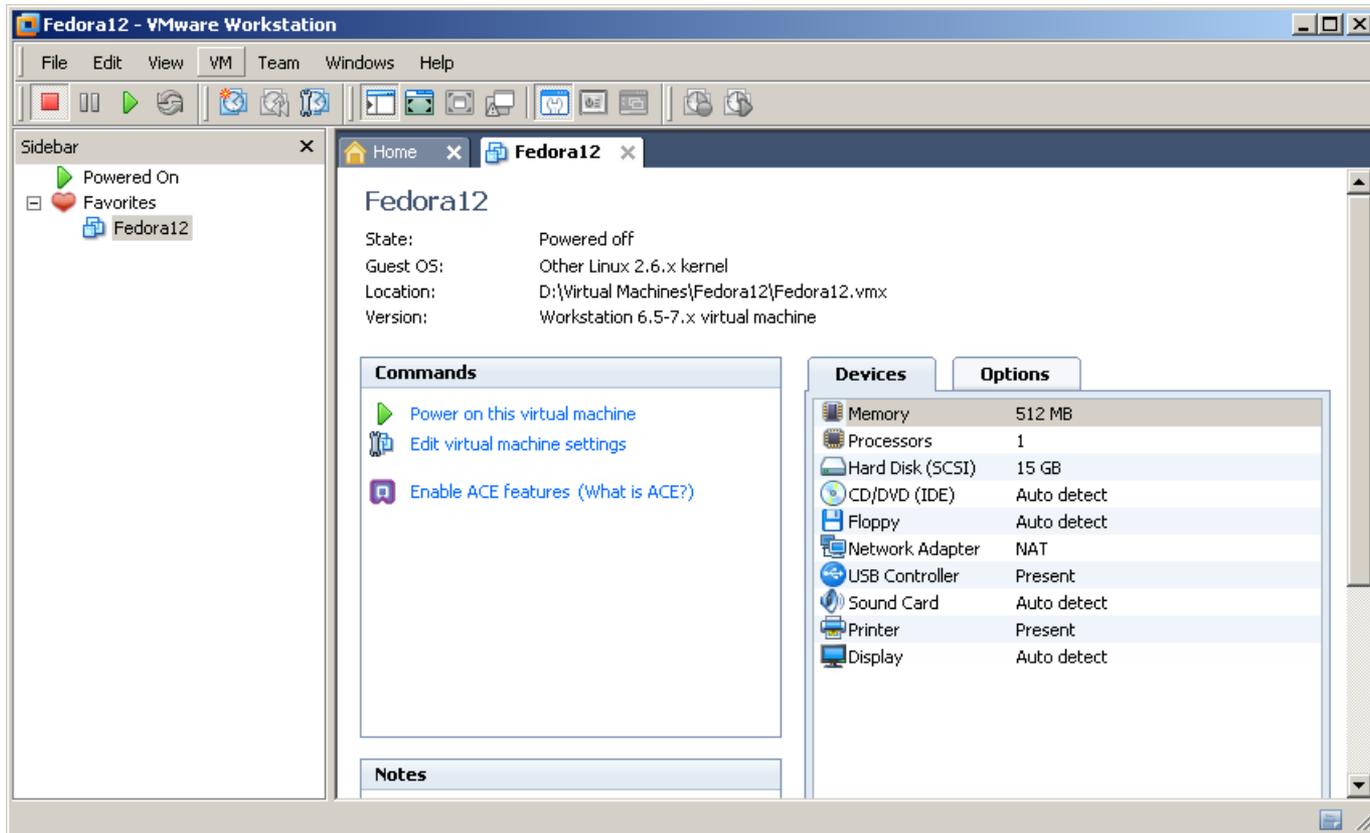
Exit from the root prompt in your terminal

Try: `sudo -s`

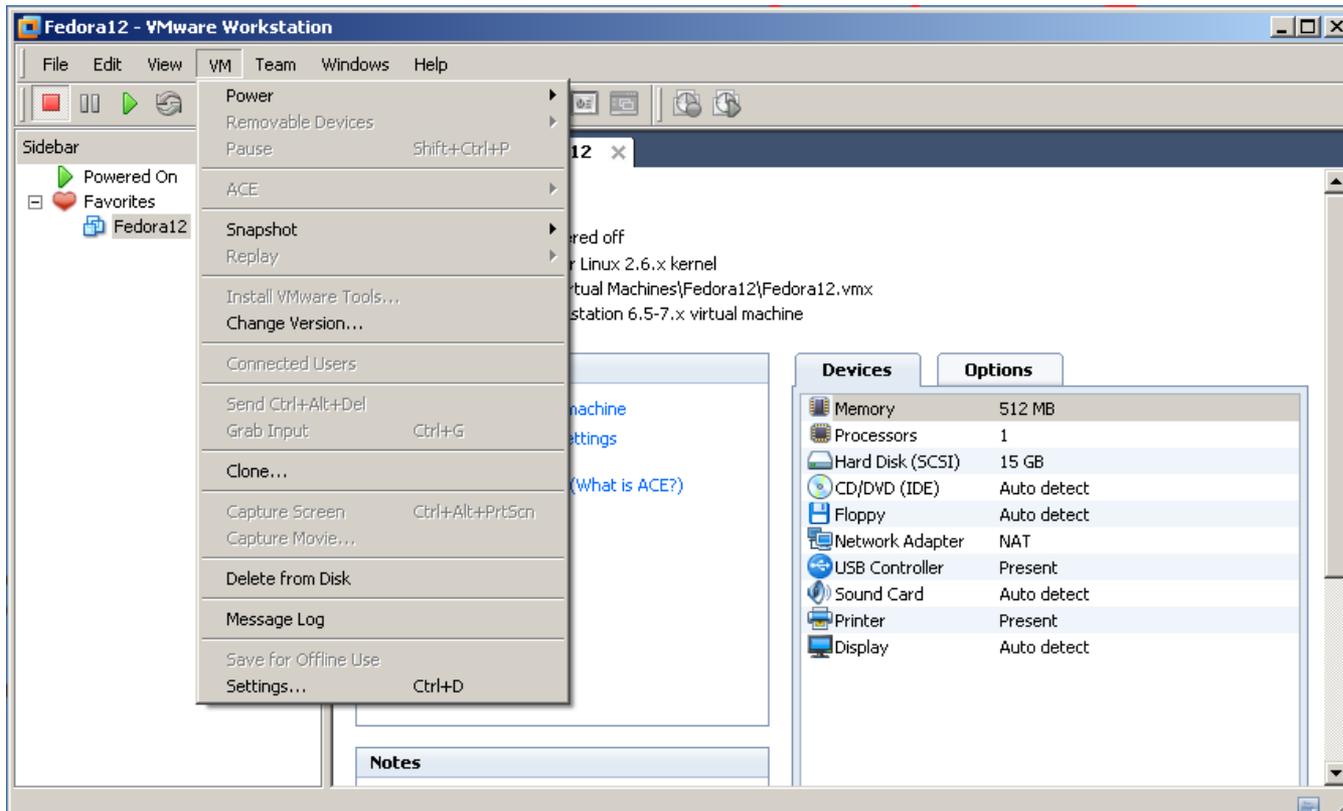
Exploring with fdisk in Lab 4

- ▶ Add disk in Vmware
- ▶ 1. Shut down Fedora and power off Virtual Machine
- ▶ 2. Virtual Machine -> Settings -> Hard Disk
- ▶ 3. Add Hard Disk (Just 1GB is enough)
- ▶ 4. boot Linux
- ▶ 5. cat /proc/partitions
- ▶ 6. be sure to use fdisk on the new 1 GB hard drive, not /dev/sda!

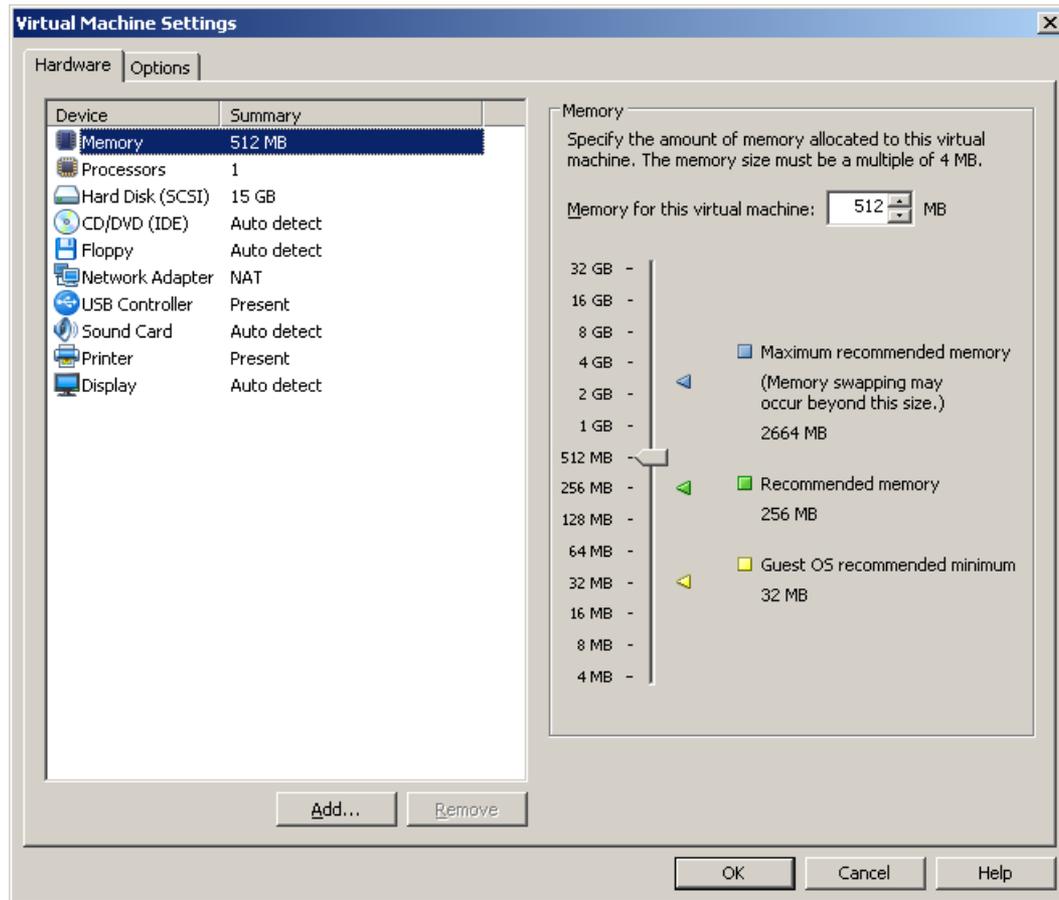
Add Hard Disk: Be sure you're dealing with your Fedora VM



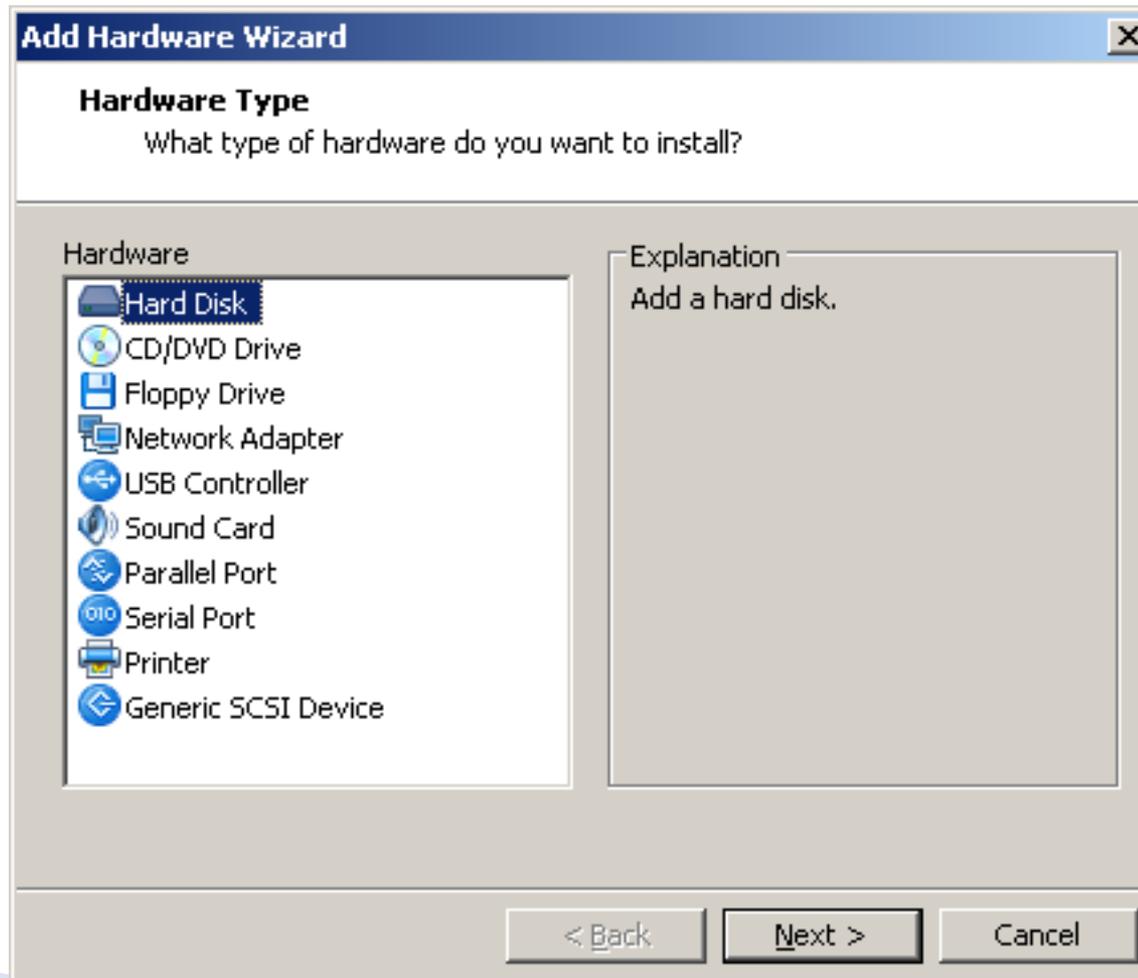
Click on the “VM” menu, select “Settings...” at the bottom



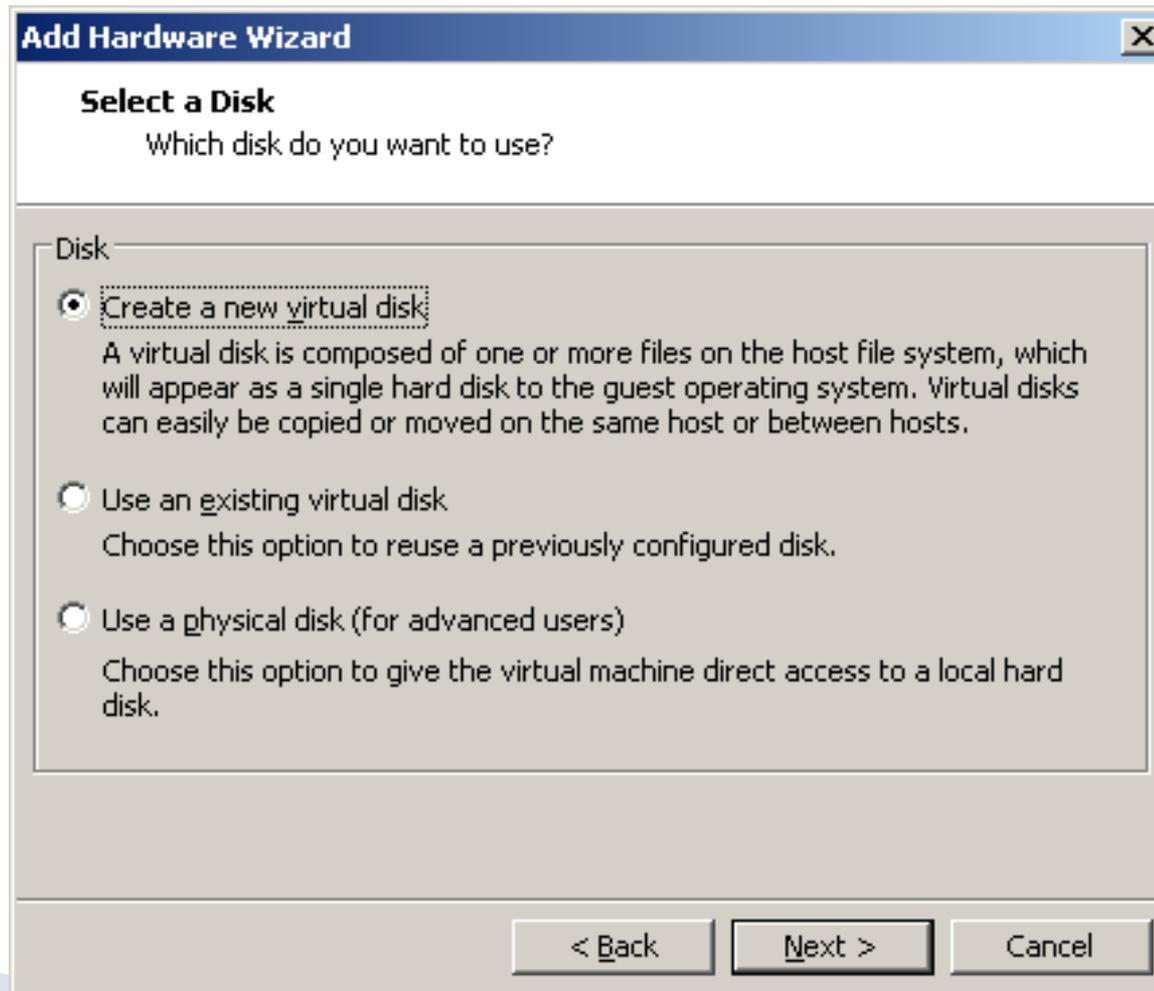
Click on the “Add...” button



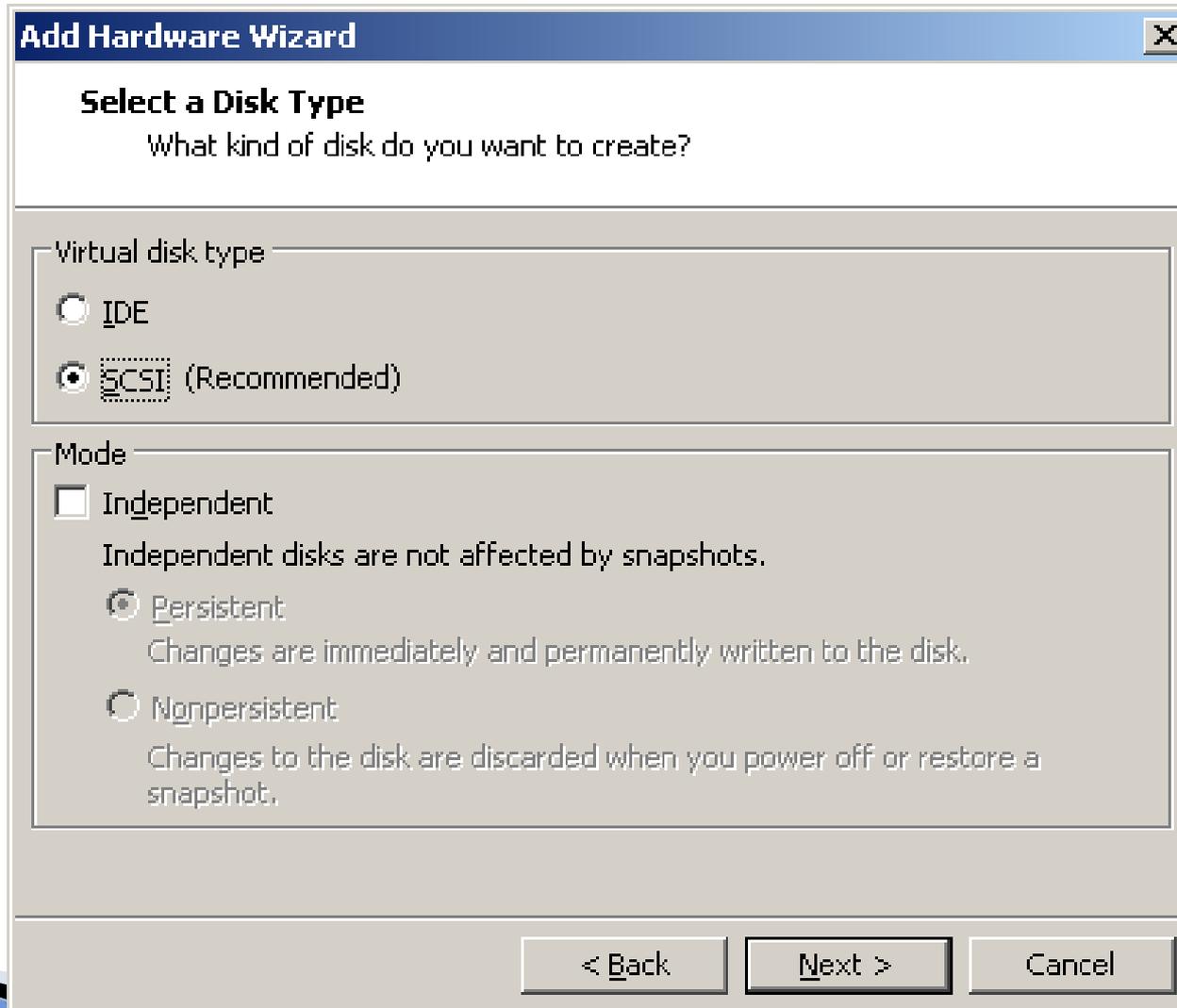
Verify “Hard Disk” is selected, click “Next”



Accept defaults, Click “Next”



Accept Defaults, Click “Next”



Add Hardware Wizard [X]

Select a Disk Type
What kind of disk do you want to create?

Virtual disk type

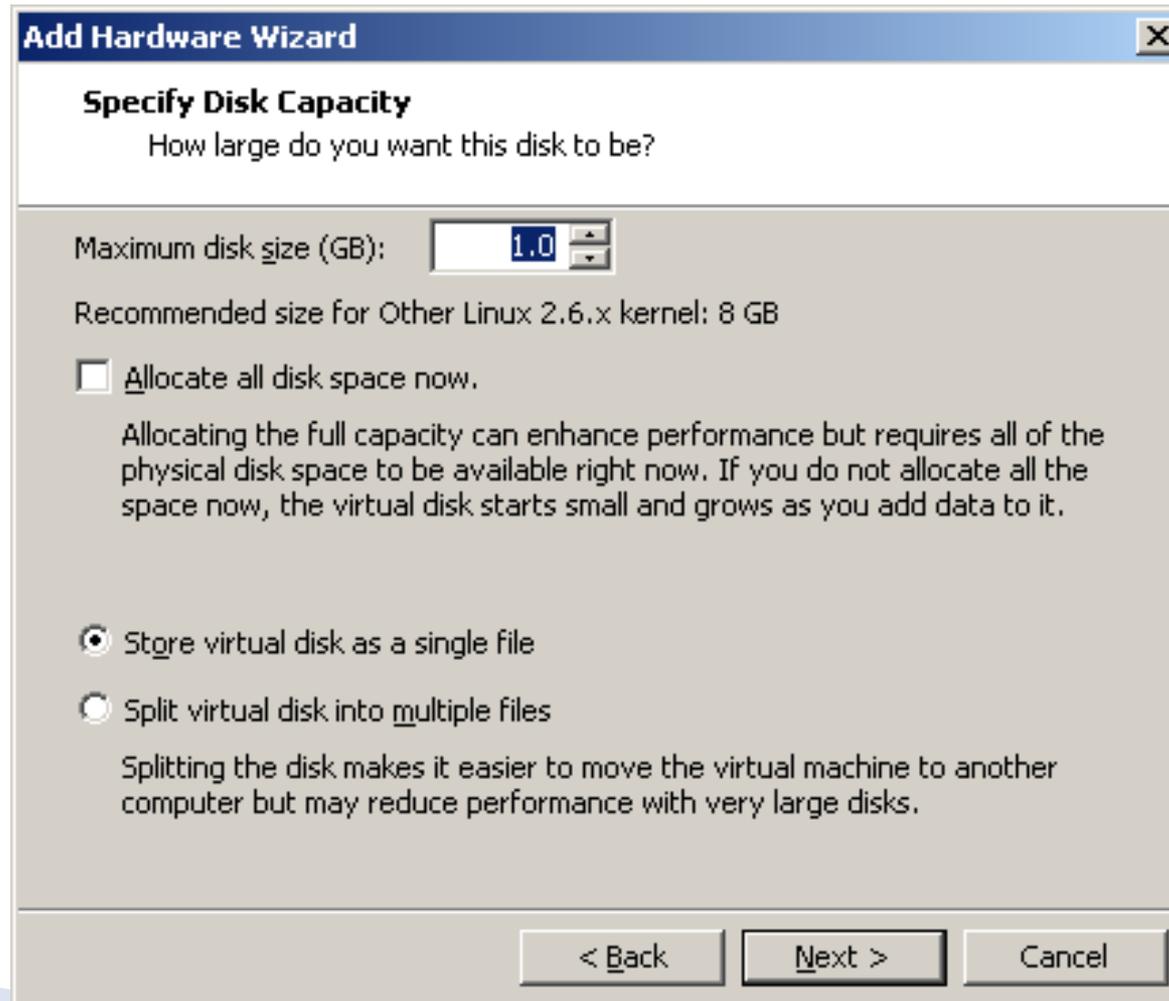
- IDE
- SCSI** (Recommended)

Mode

- Independent
Independent disks are not affected by snapshots.
- Persistent**
Changes are immediately and permanently written to the disk.
- Nonpersistent
Changes to the disk are discarded when you power off or restore a snapshot.

< Back Next > Cancel

Change size to “1.0”, click “Next”



Add Hardware Wizard [X]

Specify Disk Capacity
How large do you want this disk to be?

Maximum disk size (GB):

Recommended size for Other Linux 2.6.x kernel: 8 GB

Allocate all disk space now.

Allocating the full capacity can enhance performance but requires all of the physical disk space to be available right now. If you do not allocate all the space now, the virtual disk starts small and grows as you add data to it.

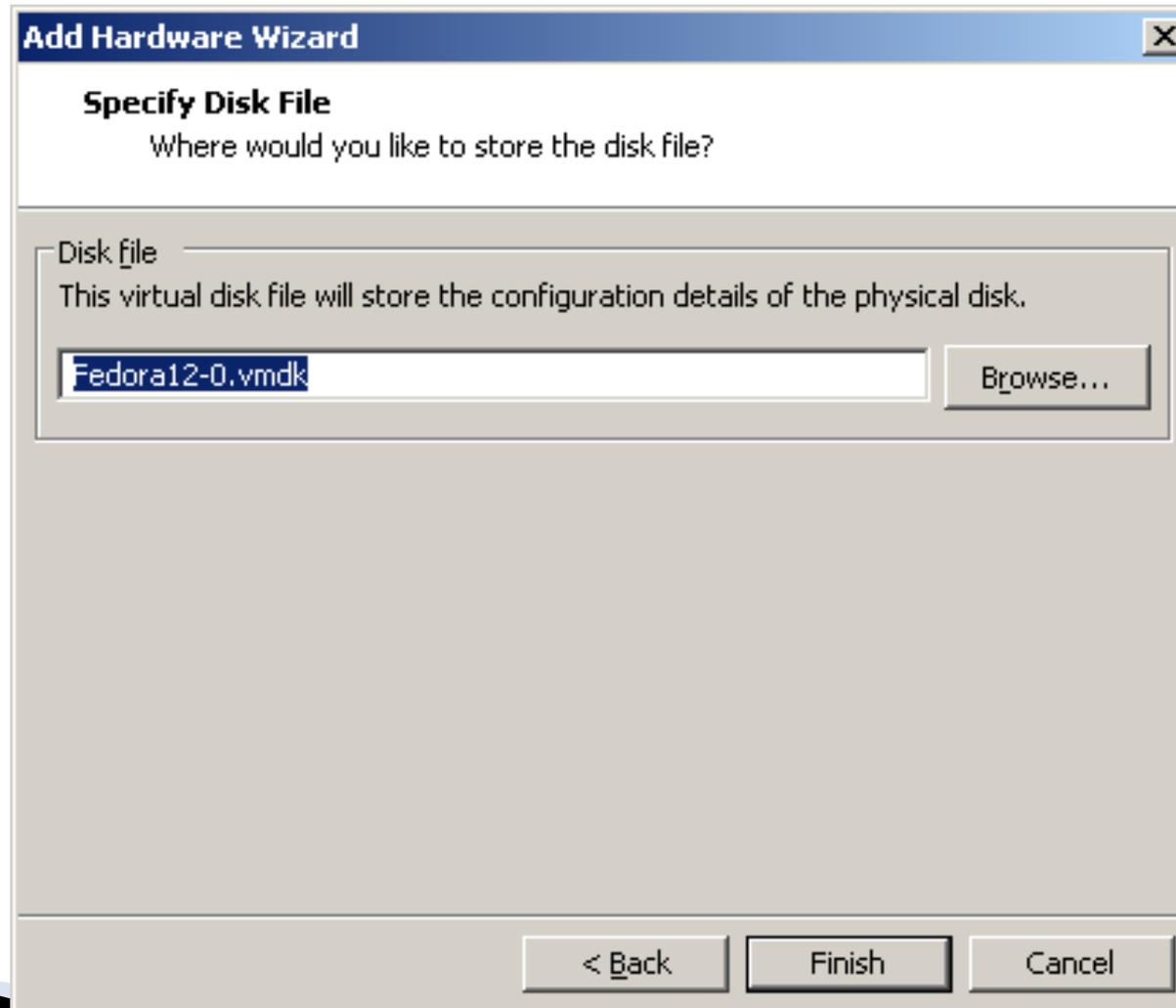
Store virtual disk as a single file

Split virtual disk into **m**ultiple files

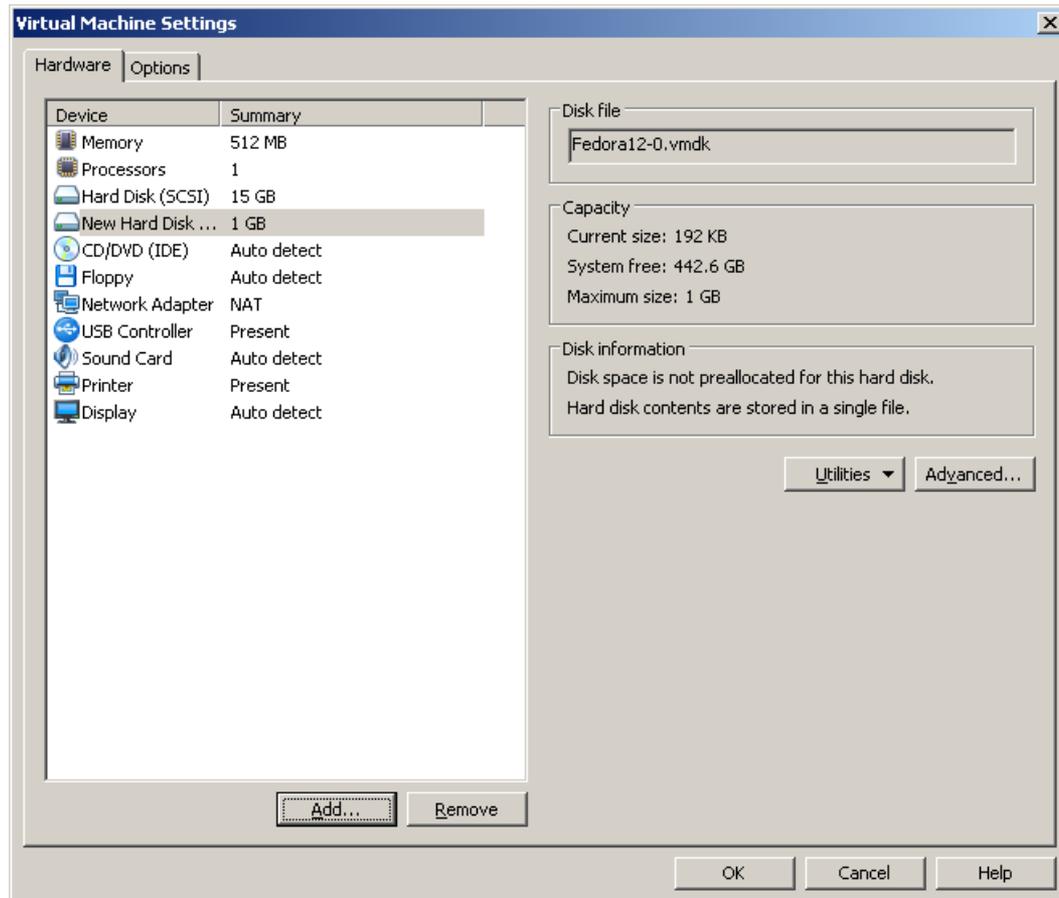
Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

< Back Next > Cancel

Accept default, Click “Next”



Notice a 1 GB Hard Disk has been added



Topics

- ▶ Prepare the drive: Partitioning
- ▶ Formatting the filesystem
- ▶ Verifying the filesystem
- ▶ Mount the filesystem

Associated Readings

- ▶ Chapter 11: page 439 (mkfs)
- ▶ Chapter 12: pages 492 (fsck)
- ▶ `man df`
- ▶ `man du`

Creating Filesystems

- ▶ There's a few steps that have to be followed to create an useable filesystem.

The steps don't change much from one operating system to another, but the specific details and utilities used vary greatly.

Creating Filesystems

- ▶ Steps include (*described for Linux*)
 - ***Preparing the device for receiving a filesystem***
 - i.e. partitioning the device
 - ***Create the filesystem needed/wanted***
 - “Format” or prepare the drive for writing the filesystem
 - ***Verify the filesystem integrity***
 - Check for flaws or errors
 - ***Mount the filesystem into the system***
 - Make it available for someone to use

Creating Filesystems in Linux

▶ Step 1: Prepare the drive

- Create, modify or change partition information
- Use **fdisk**, Disk Druid or any other partitioning tool that can manage & create Linux partitions

▶ Step 2: Create the filesystem

- “Format” or prepare the drive/partition(s) for the appropriate filesystem required
- Use appropriate **mkfs** command

Creating Filesystems in Linux

- ▶ **Step 3: Verify the filesystem**
 - Check filesystem for inconsistencies or errors
 - Use **fsck device** command
- ▶ **Step 4: Mount the filesystem for access/use**
 - Manually with all the info
 - Use appropriate **mount** command
 - **OR** add the filesystem in the **/etc/fstab** file and mount automatically or semi-automatically

Creating Filesystems in Linux

- ▶ **mkfs** [options] device
 - default is native filesystem format for version running
 - Ex. **mkfs -t ext3 /dev/hda1**
 - format IDE0 master HD, partition 1, to ext3 filesystem
 - Ex. **mkfs -t ext3 /dev/fd0**
 - format floppy to ext3 filesystem
- ▶ **mke2fs** [options] device
 - specialized version of **mkfs**
 - specific to **ext2/ext3/ext4** filesystem

Creating Filesystems in Linux

- ▶ **mkswap** [options] device
 - Makes a Linux partition a swap partition
- ▶ **swapon** device
 - activates swap partition usage
- ▶ **swapoff** device
 - de-activates swap partition usage

Creating Filesystems in Linux

▶ **fsck** [options] device

- command to help verify filesystem integrity and, if inconsistent or damaged, to try and repair the damage if at all possible
- only verifies filesystems that have been flagged as “*verifiable*” in **/etc/fstab**
- may be run at startup by kernel or manually by admin
- can **only** be run if targeted filesystem is unmounted or mounted as read-only
- will automatically execute appropriate version of **fsck** sub-scripts for filesystem type indicated at command line or by detection when accessing filesystems

Creating Filesystems in Linux

- Example: **fsck /dev/hda2**
- ▶ **e2fsck**
 - Used to check an ext2 or ext3 filesystem.
 - Accept more options than fsck

df and du commands

- ▶ **df** command
 - Used to check free space on the disk
 - Only view mounted filesystems
 - Options: **-h**
 - Example: **df -h**

- ▶ **du** command
 - Directory usage: to view the size of directory and its contents.
 - Options: **-s, -h**
 - Example: **du -hs /etc**

/etc/mtab

- ▶ This file handles the mounted devices and is automatically updated by the mount command.
- ▶ And it looks a bit similar to `/etc/fstab` but not the 100% same, and it only lists the mounted devices !
- ▶ You should not edit this file!
- ▶ To fix a broken `mtab` file:
 - `rm /etc/mtab`
 - `ln -s /proc/mounts /etc/mtab`

So... What's /proc?

- ▶ A pseudofilesystem
- ▶ From it, you can see an aspect of the Linux kernel
- ▶ You can see information on any running process including:
 - Current state
 - Memory usage
 - CPU usage
 - Terminal
 - Parent
 - Group