The Structure of ESX Server

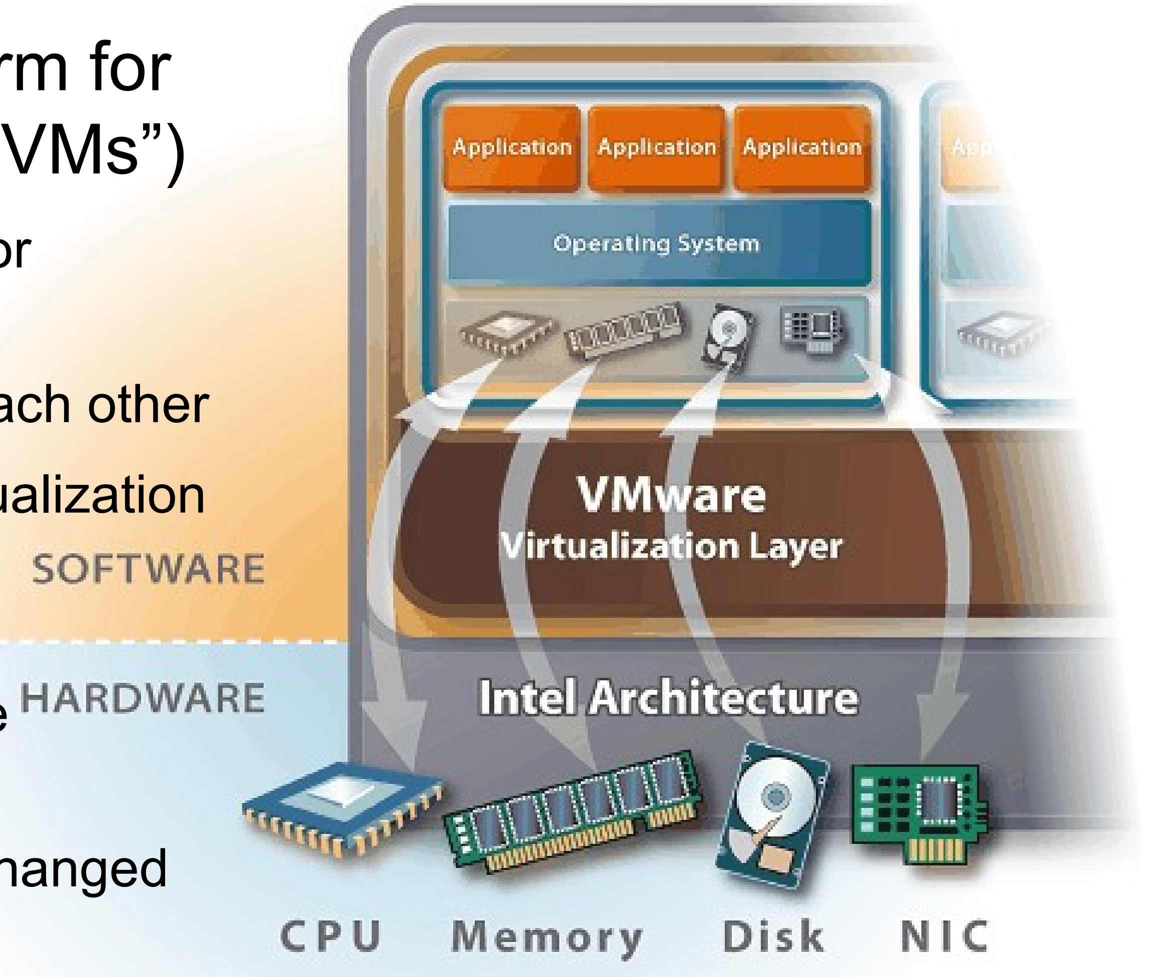
- VMnix and the VMkernel
- Virtual Machines
- VMware Services

ESX Server System Management II
Module 2

For ESX Server 2.0.1 2003-10-28

What is ESX Server?

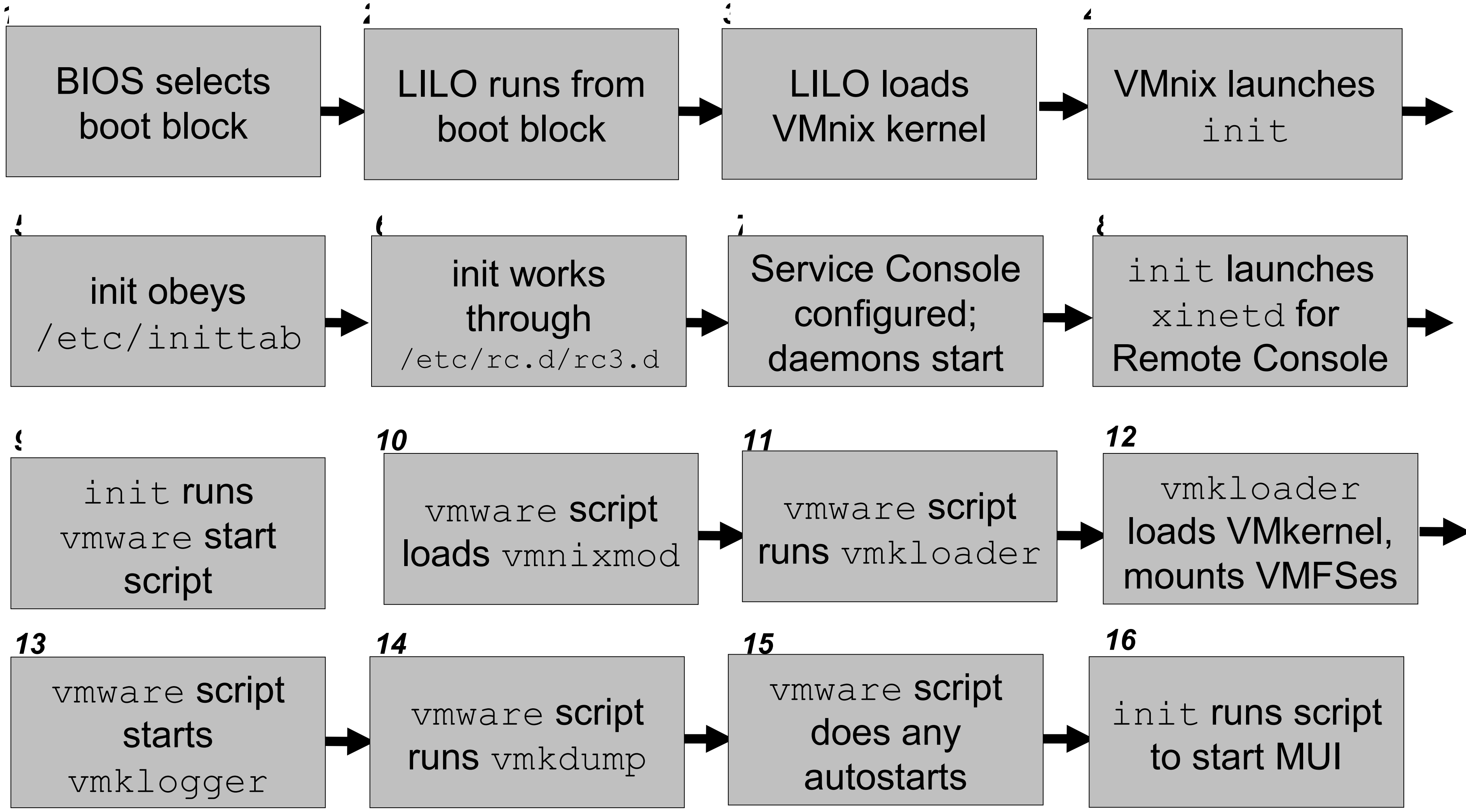
- A dedicated platform for virtual machines ("VMs")
  - VMs encapsulated for portability
  - VMs isolated from each other
  - ❖ Lowest possible virtualization overhead
  - ❖ VMs have minimum guaranteed resource allocations
  - ❖ Allocations can be changed without rebooting



The components of ESX Server

- ESX Server consists of
  - An installer package, based on Red Hat Linux 7.2
  - A lightly modified Red Hat Linux 7.2 installation
    - Called the **Service Console**, or **VMnix**
    - Serves as bootstrap loader for the VMkernel
    - Offers an administrative interface for VMkernel
    - Gives virtual machines access to non-virtualizable devices
  - A proprietary OS for managing virtual machines
    - Called the **VMkernel**

ESX Server Boot Process



1. BIOS selects boot block
2. LILO runs from boot block
3. LILO loads VMnix kernel
4. VMnix launches `init`
5. init obeys `/etc/inittab`
6. init works through `/etc/rc.d/rc3.d`
7. Service Console configured; daemons start
8. `init` launches `xinetd` for Remote Console
9. `init` runs `vmware` start script
10. `vmware` script loads `vmnixmod`
11. `vmware` script runs `vmkloader`
12. `vmkloader` loads VMkernel, mounts VMFSes
13. `vmware` script starts `vmklogger`
14. `vmware` script runs `vmkdump`
15. `vmware` script does any autostarts
16. `init` runs script to start MUI

Step 1: The BIOS and LILO

- BIOS contains a boot order
  - Among removable media, all disk controllers, network boot
- When ESX Server is properly set up, boot order points to a local hard drive
- Its first physical block contains LILO, the Linux Loader

Steps 2 and 3: LILO and VMnix

- LILO has been previously configured according to `/etc/lilo.conf`

```
prompt
timeout=50
boot=/dev/cciss/c0d0
map=/boot/map
install=/boot/boot.b
linear
default=vmnix

image=/boot/vmlinuz-2.4.9-vmnix2
        label=vmnix
        root=/dev/cciss/c0d0p2
        initrd=/boot/initrd-2.4.9-vmnix2.img
        append="mem=192M cpci=0:0,13,15,18,20;"
```

LILO gets installed in first block here

VMnix (Service Console)

Service Console memory size

Step 3: VMnix, the Service Console

- Service Console is a uniprocessor Linux 2.4 kernel, in a lightly modified Red Hat 7.2 environment
  - Uses same IOAPIC architecture for mapping interrupts as does VMkernel
- Service Console's role:
  - Bootstraps the VMkernel onto the system
  - Supports VMware Management Interface ("MUI"), Remote Console, SNMP, APIs

Steps 4 and 5: VMnix and init

- `init` is the first process spawned by any Linux kernel
- Its operation is controlled by `/etc/inittab`

```
# tag:runlevels:mode:command
id:3:initdefault:
si::sysinit:/etc/rc.d/rc.sysinit
…
l3:3:wait:/etc/rc.d/rc 3
…
1:2345:respawn:/usr/sbin/vmkstatus tty1
2:2345:respawn:/sbin/mingetty tty2
…
```

Steps 6-8: `/etc/rc.d/rc3.d`

- rc3.d contains links to start scripts, to be run in numerical order
  - `S` scripts get invoked with a `start` argument, `K` with a `stop`

`S00vmkstart`	log the change in system status
`S10network`	configures Service Console networking
`S12syslog`	starts system logger daemon
`S55sshd`	starts secure shell daemon
`S56xinetd`	starts services super-daemon
`S90vmware`	loads VMkernel, other ESX components
`S91httpd.vmware`	starts MUI server

Service Console networking setup (`S10network`)

- Service Console's NIC determined during install
- Correct driver gets associated with Service Console Ethernet by `/etc/modules.conf`

```
alias eth0 e100
```

- IP address and netmask are set in `/etc/sysconfig/network-scripts/ifcfg-eth0`

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=static
IPADDR=192.168.130.200
NETMASK=255.255.255.0
```

Service Console networking (cont'd)

- Hostname and default gateway set in `/etc/sysconfig/network`

```
NETWORKING=yes
HOSTNAME=myesx.example.com
GATEWAY=192.168.130.1
```

- DNS servers set in `/etc/resolv.conf`

```
search example.com
nameserver 192.168.130.2
nameserver 192.168.2.1
```

- Hostname needs an entry in `/etc/hosts`

```
192.168.130.200 myesx.example.com myesx
```

Service Console services (`S12syslog, S55sshd`)

- `syslogd` is governed by `/etc/syslog.conf`

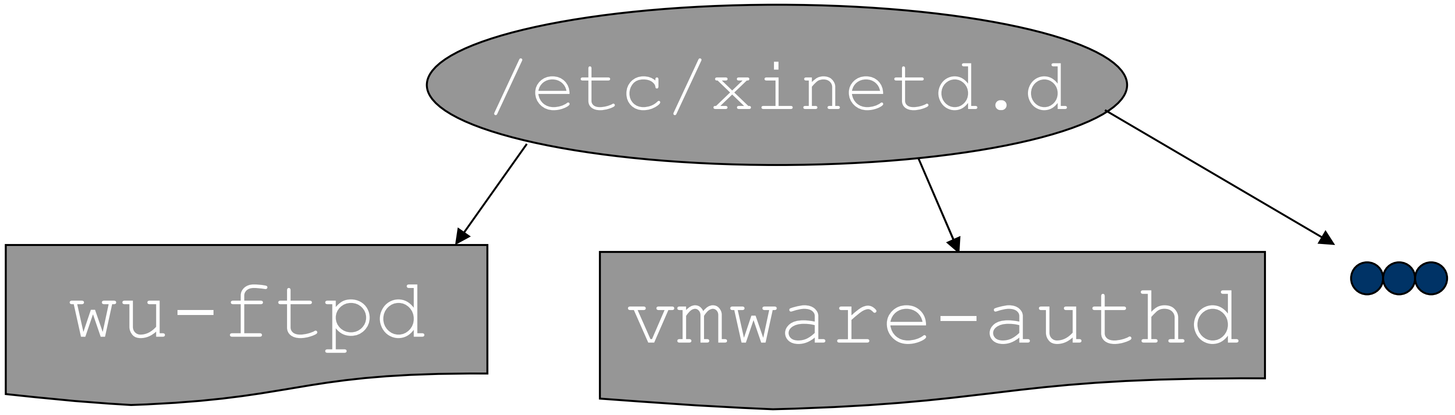
```
*.info;mail.none;authpriv.none;cron.none;➘
     local6.none /var/log/messages
local6.info            /var/log/vmksummary
```

- `sshd` is governed by `/etc/ssh/sshd_config`

```
#Port 22
#Protocol 2,1
…
Subsystem sftp /usr/libexec/openssh/sftp-server
```

xinetd and Remote Console (S56xinetd)

- xinetd listens for incoming requests for service



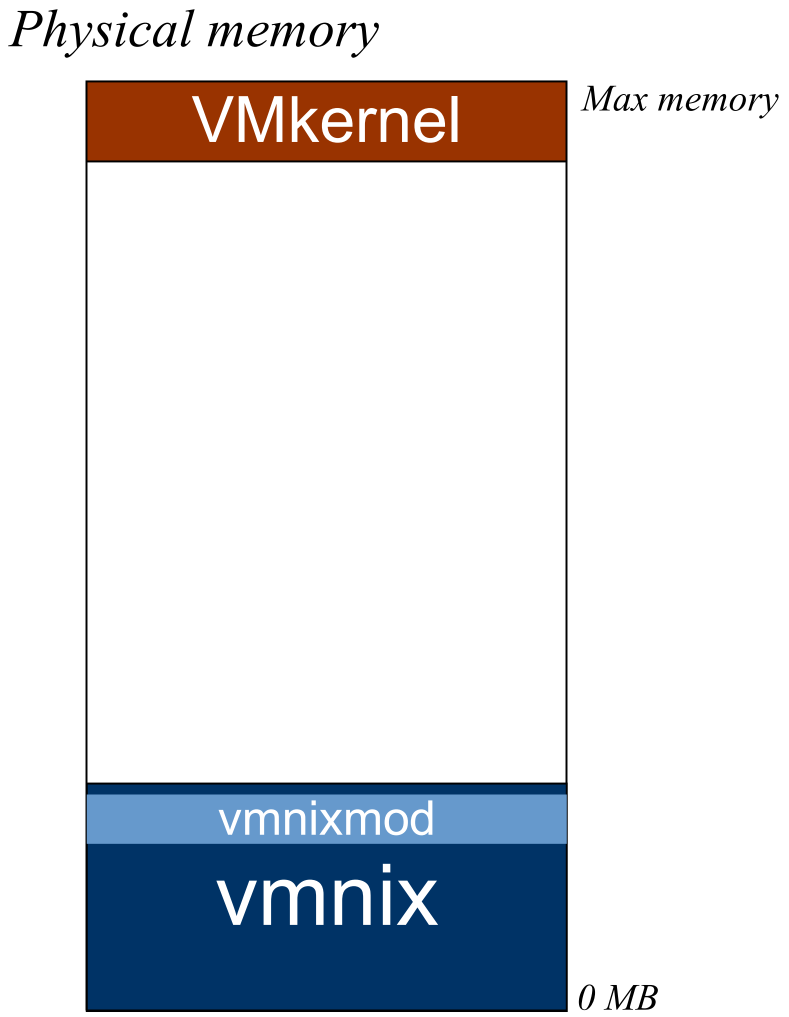
```
service vmware-authd {
   disable = no
   port = 902
   server = /usr/sbin/vmware-authd
}
```

- vmware-authd is the front end to Remote Console

Steps 9-12: The VMkernel (`S90vmware`)

`vmware` script

```
modprobe ➔
   vmnixmod.o
vmkloader
```



VMkernel device drivers (`S90vmware`)

- Each PCI device has a vendor and sub-vendor ID
  - To scan physical system for PCI devices, showing vendor IDs

```
# lspci -H1 -M -n
[…]
00:0f.0 Class 0200: 8086:1229 (rev 05)
[…]
```

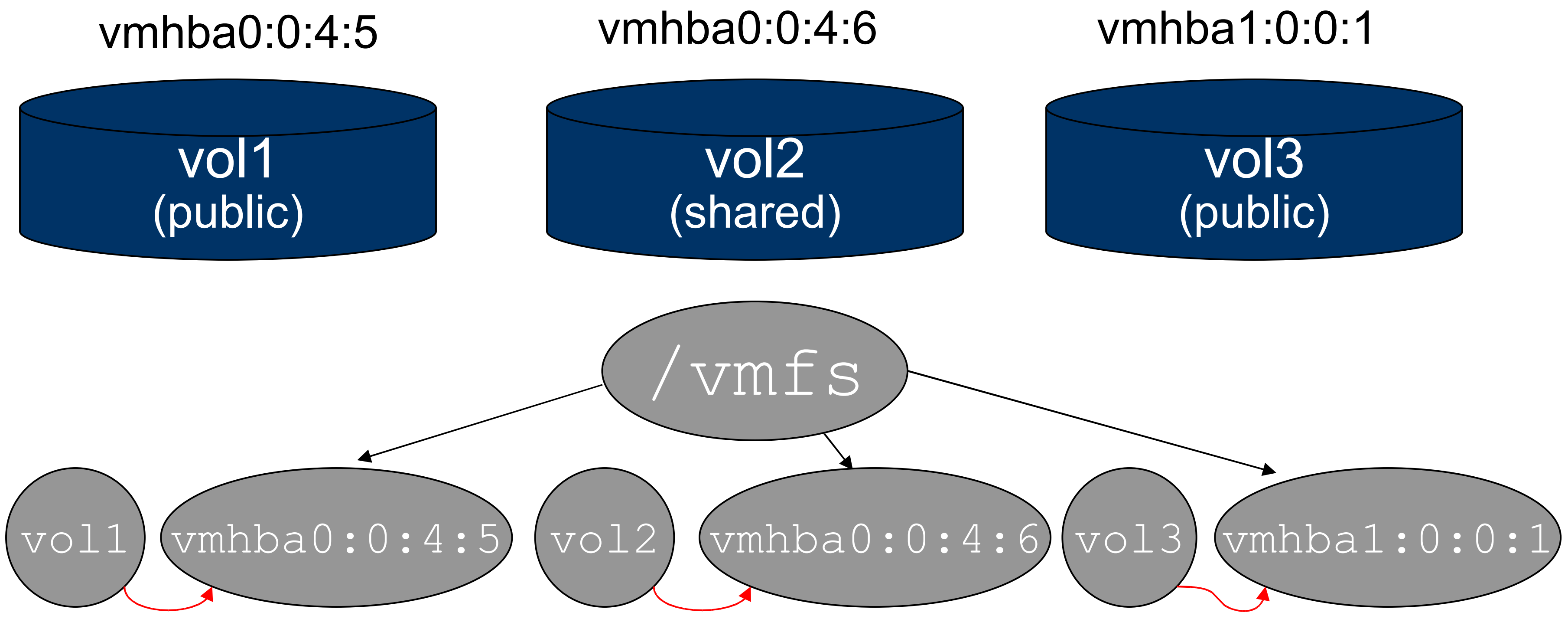
Intel vendor ID

- Mapping from PCI vendor IDs to VMkernel drivers:

```
# grep 1229 /etc/vmware/vmware-devices.map
device,0x8086,0x1229,vmnic,Intel PRO/100,e100.o
```

  - To ensure that upgrades do not clobber your changes use `/etc/vmware/vmware-devices.map.local`
- VMkernel drivers reside in `/usr/lib/vmware/vmkmod`

vmkloader mounts VMFS volumes (S90vmware)



- VMFS volumes are always visible in Service Console

Steps 13-14: VMkernel services (`S90vmware`)

- `vmware` boot script starts `vmklogger`
  - Relays VMkernel messages to `syslogd`, the system logger daemon
- `vmware` boot script runs `vmkdump`
  - Saves any prior VMkernel dump from the VMkernel dump partition

Step 15: Virtual machine autostarts (`S90vmware`)

- `vmware` boot script launches `vmstartstop.pl`
- `vmstartstop.pl` reads all registered VMs' configuration files from `/etc/vmware/vm-list`
- If a VM's configuration file contains `autostart = "true"`, it will be powered on
  - And boot, if it has a boot device with a valid boot block
  - `autostart = "poweron"` and `autostart = "resume"` will also be powered on

Step 16: VMware Management Interface (`S91httpd.vmware`)

- Script `S91httpd.vmware` starts the Web server for the VMware Management Interface ("MUI")
  - Special-purpose Apache Web server
- Web server uses a proprietary form of XML RPC to communicate with `vmware-serverd`
  - `vmware-serverd` does back-end processing for MUI, Remote Console, scripting interfaces
  - Started on demand

`vmware` as seen by `ps`

- `ps -ef` reveals:

```
susie 1360 1 [vmware-vmx]
susie 1362 1360 [vmware-mks]
```

  - Subprocesses perform Service Console I/O on behalf of the VM
- These `vmware` binaries…
  - loads the monitor
  - Supplies the virtual mouse, keyboard, screen
  - Saves a monitor log to the Service Console’s filesystem
  - Runs as long as either VM is powered on or a Remote Console session exists

ESX Server's concept of users

- A Service Console user is an ESX Server user
- Each file on disk has an owner
- Each virtual machine has an owner
  - As defined by the owner of its configuration file
  - Configuration files reside under users' home directories

    ```
    /home/susie/vmware/webserver/webserver.vmx
    ```
- Service Console stores known users in `/etc/passwd`

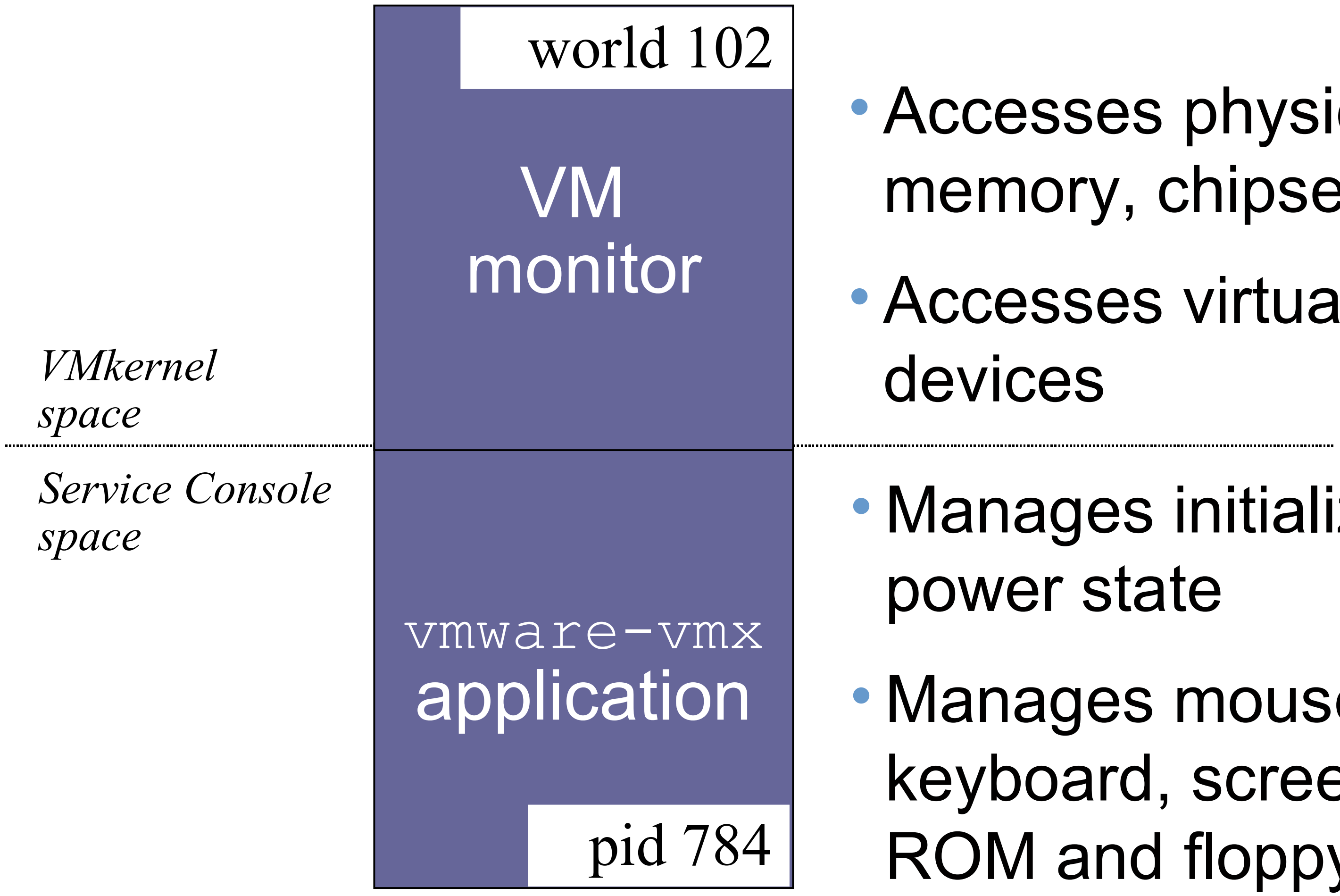
  ```
  susie:x:501:501:Susie Park:/home/susie:/bin/bash
  ```
  - One-way-encrypted passwords are stored in `/etc/shadow`

    ```
    susie:$1$YyÞsWÉFï$mtPIp6NF32Fu5LG1MzBDV0: ⮯
    120180:99999:7:::
    ```

Launching a virtual machine

- VMkernel's concept of a process is a *world*
  - Each VM occupies its own world
    - There are also non-VM helper worlds
  - Each world has an integer *world id*
- VMkernel to load an instance of the monitor for this VM into a new world
  - Monitor guarantees correct access to shared resources
  - Runs as owning user

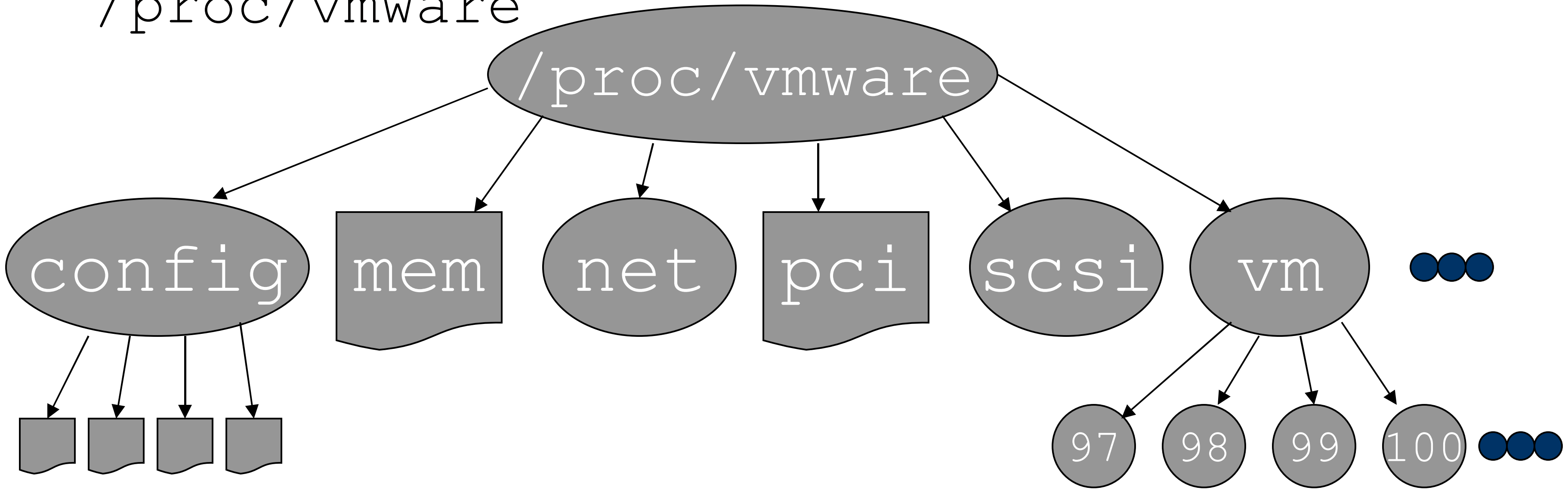
A virtual machine under ESX Server



- Accesses physical memory, chipset
- Accesses virtual devices
- Manages initialization, power state
- Manages mouse, keyboard, screen, CD-ROM and floppy

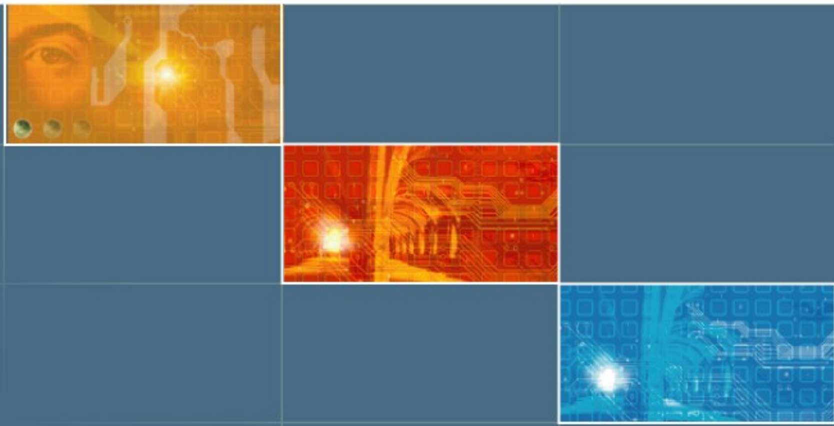
The `/proc/vmware` hierarchy

- Unix/Linux kernels publish status under `/proc`
  - Files and directories occupy no disk space
  - Offer an easy-to-use window into the kernel's state
- VMkernel publishes information under `/proc/vmware`



vmware™

Questions?



ESX Server System Management II
Module 2