

# NFS and Automounting

---

*“Transparent access to files and directories on the network”*

---

CIS 68C2

UNIX Network Administration

# NFS

## □ NFS – Network File System

- ✗ Shares directory trees across the network

- ✗ Developed by Sun Microsystems

- ✗ Supported on

- ✗ all major UNIX's

- ✗ PCs via PC NFS

- ✗ Macs via Mac NFS

- ✗ File sharing means: *transparent access*

- ✗ Remote files & directories exist somewhere under /

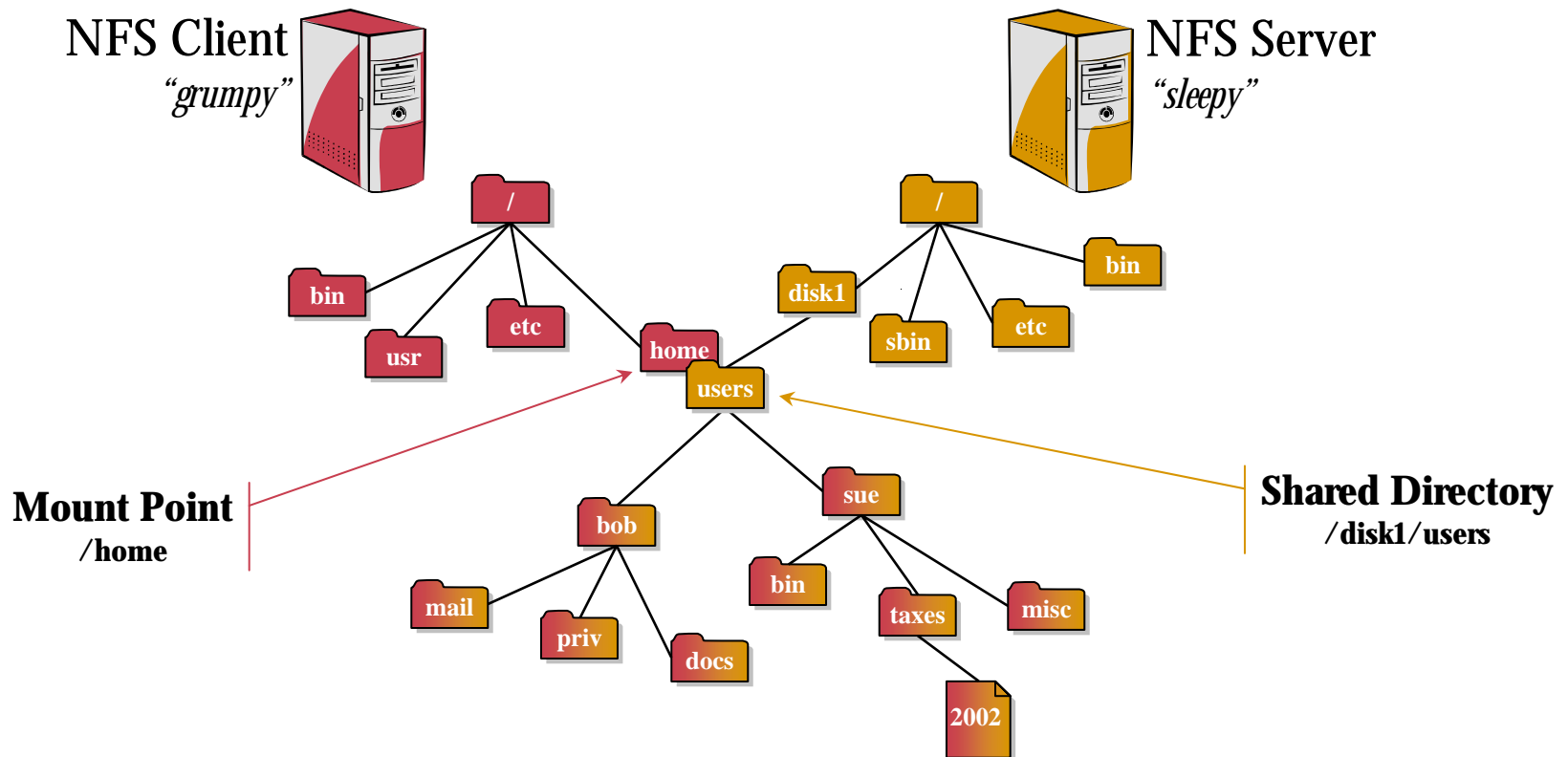
- ✗ UNIX commands have transparent access to remote files

- ✗ No need to transfer files via **ftp**, **rnp**, etc.

*“The poster child for everything that is or ever has been wrong with UNIX security”*

# NFS

- *Sleepy's* shared files appear within *grumpy's* filesystem



# NFS

## □ NFS

- ✗ Two versions of the NFS protocol exist
  - ✗ Version 2
  - ✗ Version 3
    - ✗ Much better performance; supports large files
    - ✗ Falls back to V2 protocol to interoperate w/V2 clients
- ✗ Runs over RPC, using TCP or UDP
- ✗ NFS has both client-side and server-side components
- ✗ Started from boot script `/etc/init.d/nfs`
  - ✗ Runlevel 3 or 5: `/etc/rc[35.]d/[SK]*nfs`

# NFS

## □ NFS Daemons

### ✗ nfsd

- ✗ Server-side component
- ✗ Services NFS requests from clients
- ✗ Several copies of the **nfsd** process is typical
  - ✗ **ps -ef** will show multiple processes or **nfsd N**

### ✗ mountd (or rpc.mountd)

- ✗ Server-side component
- ✗ Services filesystem **mount** requests from clients

### ✗ quotad (or rpc.quotad)

- ✗ Supports client-side viewing of disk quotas
- ✗ Generally considered obsolete, since large disks are cheap

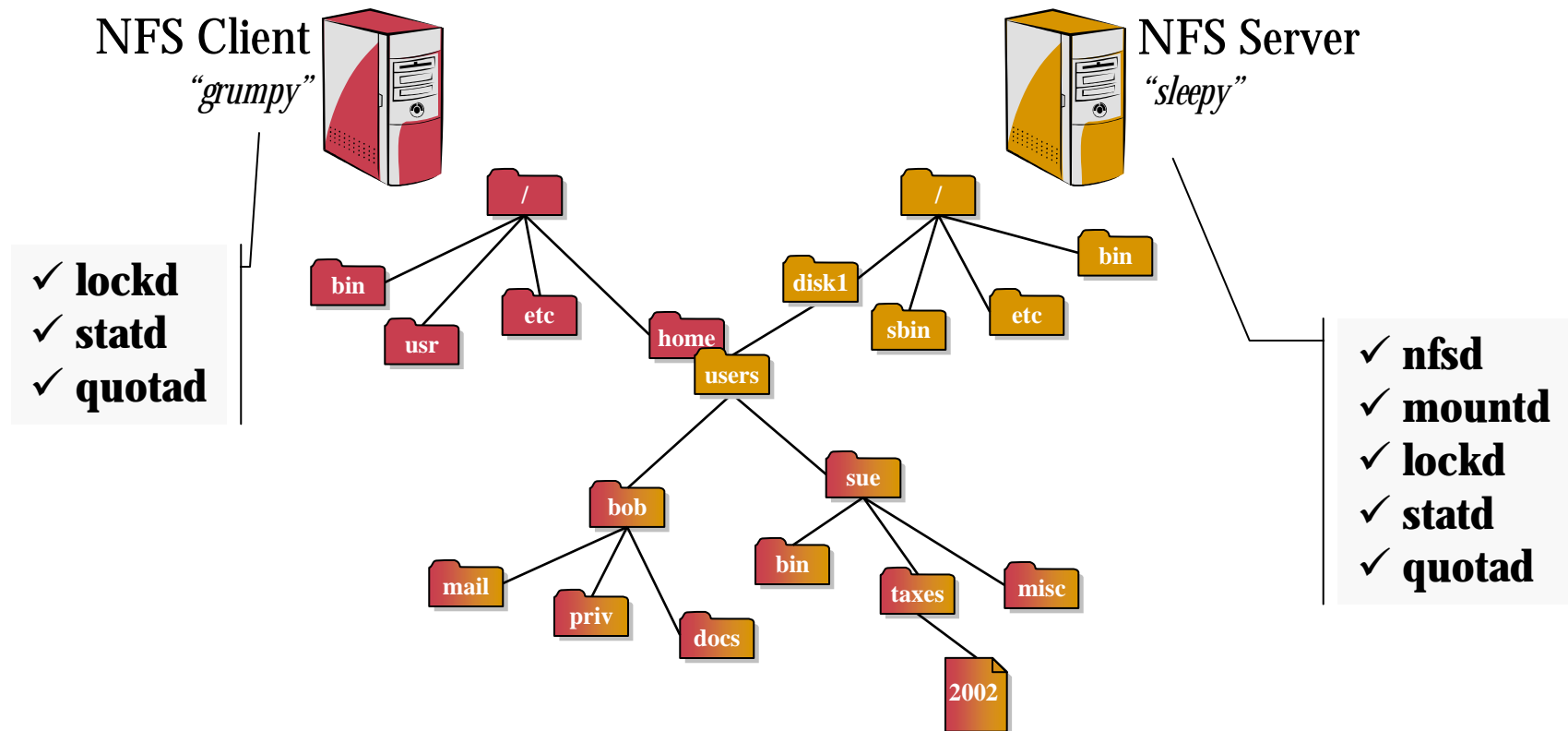
# NFS

## □ NFS Daemons

- ✗ lockd (or rpc.lockd)
  - ✗ Client-side and server-side components
  - ✗ Supports remote *file locking* requests
- ✗ statd (or rpc.statd)
  - ✗ Client-side and server-side components
  - ✗ Monitors NFS server status
  - ✗ Required for **lockd**
    - ✗ Removes unused locks not removed properly
- ✗ Remote file locking is problematic at best

# NFS

## □ NFS Daemons



# NFS

## □ Sharing / Exporting

- ✗ Only entire directories trees can be shared
  - ✗ Cannot share individual files
  - ✗ All files and directories under exported directory are shared
- ✗ Directories are exported using **exportfs** command
  - ✗ Maintains the binary file **/var/lib/nfs/xtab**
    - ✗ Read by **mountd** when remote host requests access
    - ✗ Created from ASCII file **/etc/exports**
  - ✗ **-a** option exports all entries in **/etc/exports**
  - ✗ **-r** option synchronizes **xtab** and the kernel export table
    - ✗ Use after adding or deleting entries from **/etc/exports**

# NFS

## □ /etc/exports

✗ Configuration table listing exported directories & options

✗ Syntax: *directory* [*host(option)*] ...

*directory*      The directory to be shared

*host*            The host(s) allowed to mount *directory*; can be a hostname,  
\* and ? wildcards, and IP network address(es)

*option*          Filesystem options (see: *man exports*)

```
$ cat /etc/exports
/usr/local            *(ro)
/usr/share/man        *(ro) bigcheese(rw,no_root_squash)
/                     bigcheese(rw,no_root_squash)
/project/src          *.eng.localdomain(rw)
/usr/src/redhat        @developers(ro)
/home/export          192.168.10.0/24(rw,root_squash)
```

# NFS

## □ Mounting/Un-mounting NFS Filesystems

✗ Use **mount** command to mount an exported filesystem

✗ Syntax: **mount *host.rmdir localdir***

*host*            The hostname that is sharing *rmdir*

*rmdir*           Remote directory to be mounted locally  
May mount any sub-tree of *rmdir*

*localdir*        Where *rmdir* will be accessed locally

✗ **-a** option will mount all *auto* filesystems in /etc/fstab

✗ Use **umount** command to un-mount a filesystem

✗ Syntax: **umount *host.rmdir***

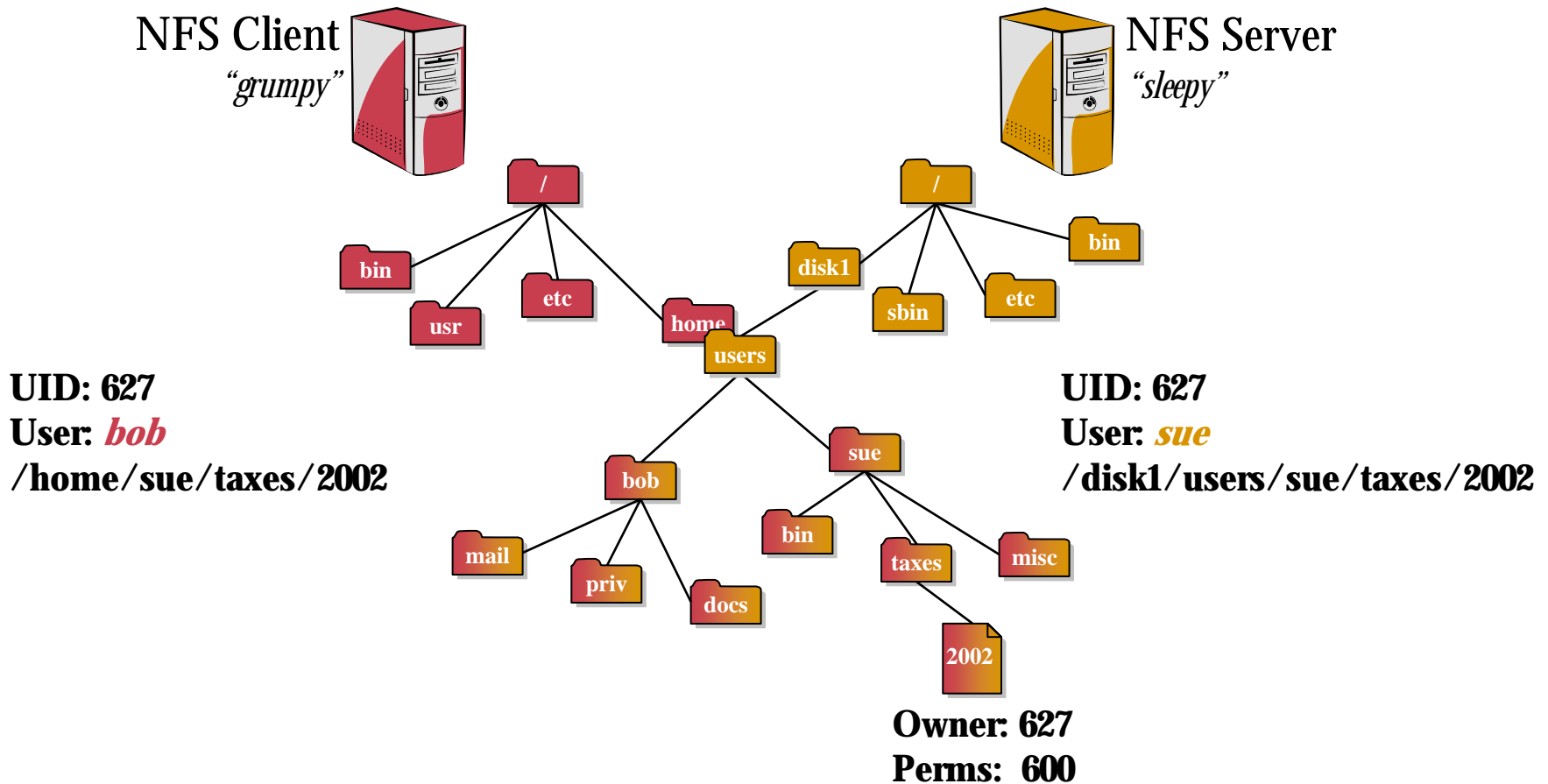
*or*        **umount *localdir***

# NFS

- NFS: UIDs & GIDs
  - ✗ File/directory ownership is relative to NFS client
    - ✗ NFS requires globally consistent user and group IDs
- NFS: UID 0 (root)
  - ✗ UID 0 on NFS client would allow access to NFS files
  - ✗ NFS provides mechanism to remap UID 0 to a safer UID
    - ✗ **root\_squash** is default option in **/etc/exports**
      - ✗ Maps UID 0 to UID -2 for NFS activity
    - ✗ Often there is an entry in **/etc/passwd** for this entry
      - ✗ Account name **nobody**, **nfsnobody**, or **anonymous**

# NFS

## □ NFS: UIDs & GIDs



# NFS

- /etc/fstab
  - ✗ Filesystem mounting configuration table
  - ✗ Assists in boot-time mounting
  - ✗ Used for both local and network filesystems
  - ✗ Used by **mount** command
    - ✗ Unspecified arguments to **mount** are taken from /etc/fstab
  - ✗ Also used by **umount**, **fsck**, **dump**, and **swapon**
  - ✗ Order of /etc/fstab records matters
    - ✗ Read by commands top-to-bottom

# NFS

## □ /etc/fstab

✘ Syntax: *filesystem mountpoint fstype options dump-level fsck-pass*

<i>filesystem</i>	Filesystem (or device) to be mounted
<i>mountpoint</i>	Where filesystem (or device) is to be accessed
<i>fstype</i>	Type of filesystem (i.e. nfs, ext2, iso9660, msdos)
<i>options</i>	Options supplied to mount command (see: man mount) auto – will be mounted with mount –a command
<i>dump-level</i>	Level for dump command (set to 0 for NFS mounts)
<i>fsck-pass</i>	Determines relative order <b>fsck</b> will run on filesystem (set to 0 for NFS mounts)

# NFS

## □ /etc/fstab

### ✗ Example

```
$ cat /etc/fstab
LABEL=/ / ext2 defaults 1 1
LABEL=/home /home ext2 defaults 1 2
/dev/cdrom /mnt/cdrom iso9660 noauto,owner,ro 0 0
/dev/cdrom1 /mnt/cdrom1 iso9660 noauto,owner,ro 0 0
/dev/fd0 /mnt/floppy auto noauto,owner 0 0
/dev/fd0 /mnt/floppyextfs ext2 noauto,owner 0 0
none /proc proc defaults 0 0
none /dev/pts devpts gid=5,mode=620 0 0
/dev/hde5 swap swap defaults 0 0
bambi:/usr/local /usr/local nfs auto,hard,intr,bg 0 0
doc:/usr/local/man /usr/local/man nfs auto,hard,intr,bg 0 0
goofy:/source /home/src nfs noauto,hard,intr,bg 0 0
```

# NFS

## □ NFS Diagnostics

- ✗ **showmount** – information about exported filesystems
  - ✗ Syntax: **showmount *options* [ *host* ]**
  - ✗ **-e** option
    - ✗ Lists exported filesystems
  - ✗ **-a** option
    - ✗ Lists exported filesystems that have been mounted by some client
  - ✗ Optional *host* argument returns information relative to that host
- ✗ **nfsstat** – NFS client/server statistics
  - ✗ Used for performance tuning and troubleshooting
- ✗ Also **netstat -s** and **rpcinfo**

# Automounter

- Automounter
  - ✗ Mounting of filesystems happens auto-magically
    - ✗ Mounted when file/directories within filesystem are referenced
    - ✗ Un-mounted when unused
  - ✗ Two varieties of automounter for Linux:
    - ✗ **autofs**
      - ✗ Virtual filesystem implemented within the kernel
    - ✗ **amd**
      - ✗ User level program, runs on any UNIX platform
    - ✗ We will not be covering the **amd** automounter

# Automounter - autofs

## □ Automounter Components

### × **autofs**

- × Kernel-level virtual filesystem driver

### × **automount**

- × Daemon that assists kernel's **autofs** in mounting / un-mounting
- × One **automount** process per mount point

### × Configuration Files (Maps)

- × A master configuration file, and one or more second-level files

### × **/etc/init.d/autofs**

- × Script that processes the master configuration file
- × Takes single argument: **start, stop, restart, reload, status**
- × Don't confuse **autofs** kernel driver with **/etc/init.d/autofs**

# Automounter - autofs

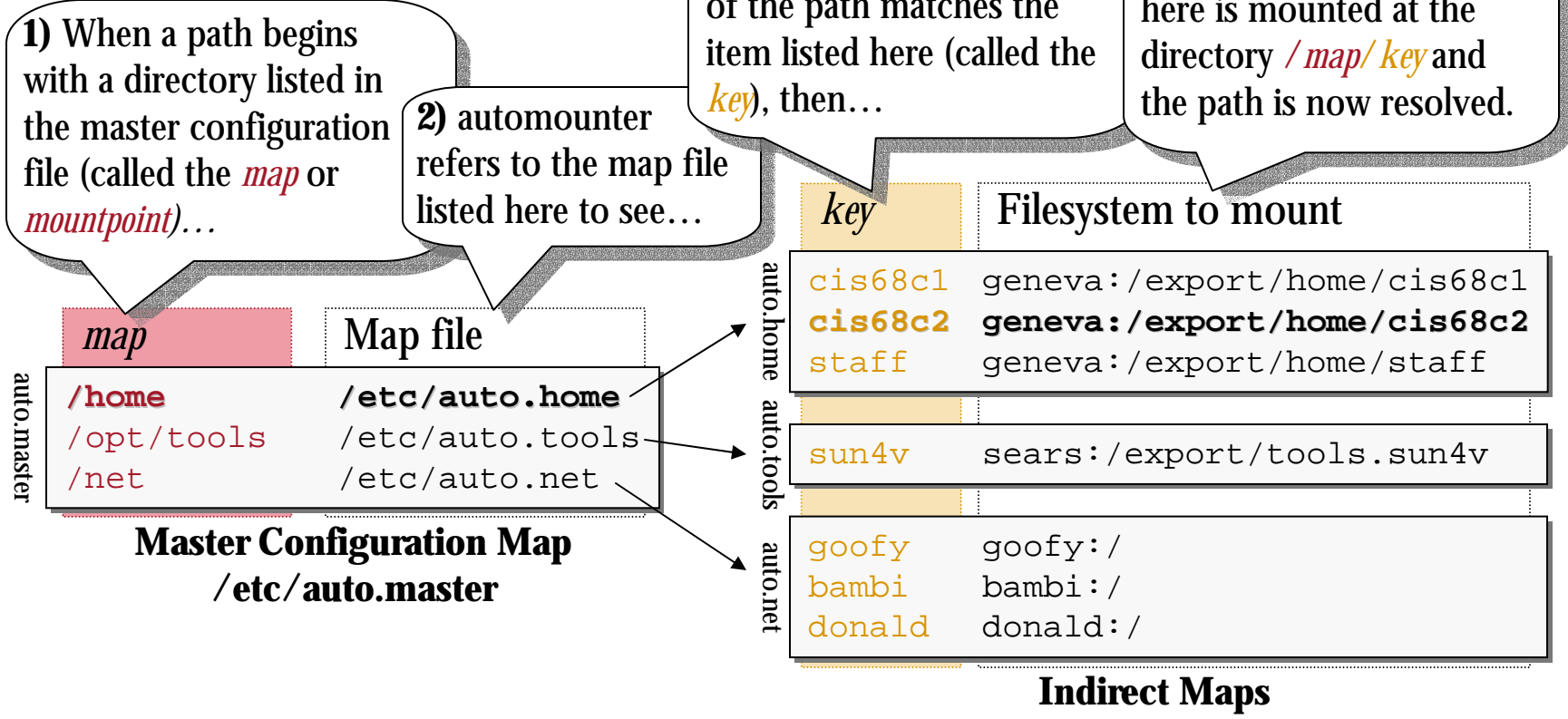
- Configuration Files (also called *maps*)
  - ✗ /etc/auto.master - Master configuration file
    - ✗ The map of maps
    - ✗ Each entry refers to a second level configuration file
      - ✗ Except for Direct maps, which are unsupported in Linux
  - ✗ Second level configuration files (maps):
    - ✗ Indirect map
      - ✗ Used when there is a common leading prefix in automounted paths
    - ✗ Executable map
      - ✗ Used to run a program to run to generate automounter instructions

# Automounter - autofs

## □ Example

### ✗ Indirect Maps

**/home** / **cis68c2** / **gumby** / **letter.txt**  
*map*      *key*      *relative path on mounted fs*



# Automounter - autofs

## □ /etc/auto.master

✘ Modifications to file are not noticed by automounter

✘ Notify automounter with command: **/etc/init.d/autofs reload**

✘ Syntax: *map mapfile options*

*map*        Top-level mount directory (AKA: *mountpoint*)

*mapfile*    Name of map file to use for sub-directory resolution

*options*    Mounting options; cumulative with those in *mapfile*

```
$ cat /etc/auto.master
# Format of this file:
# map mapfile options
# For details of the format look at autofs(8).
/net            /etc/auto.net            --timeout=60
/home          /etc/auto.home         --timeout=300
/opt/tools     /etc/auto.tools        --timeout=300
```

# Automounter - autofs

## □ Indirect Maps

- ✗ Series of records, each indicating a directory beneath *map*, mounting options, and the filesystem to be mounted
- ✗ Syntax: *key options location*

*key* Subdirectory under *map* – mount happens at */map/key*

*options* Mounting options, cumulative w/auto.master

*location* Filesystem to be mounted; typically NFS mount but can be any mountable filesystem (local, samba, etc.)

```
$ cat /etc/auto.net
# Eg. When /net/goofy/XYZ is referenced, goofy:/ is mounted
# at /net/goofy and XYZ is resolved
goofy      -rw,hard,intr      goofy:/
bambi     -rw,hard,intr      bambi:/
donald    -rw,hard,intr      donald:/
```

# Automounter - autofs

## □ Indirect Maps

✗ Wildcards can be used to simplify mount mappings

\* As the *key* matches everything

& Refers to the *key* that was matched

✗ Variables generalize system dependent mappings

✗ See autofs(5) for list of built-in variables

```
$ cat /etc/auto.home
# Automounts goofy:/export/home/joe onto /home/joe
joe          -rw,hard,nointr      goofy:/export/home/&
# Automounts xxx:/home/xxx onto /home/xxx
*           -rw,hard,nointr      &:/home/&
$ cat /etc/auto.tools
# Automounts sears:/export/tools.$ARCH onto /opt/tools/$ARCH
$ARCH      -rw,hard,nointr      sears:/export/tools.$ARCH
```

# Automounter - autofs

- Executable maps
  - ✗ Also called **program** maps
  - ✗ Allows scripting to generate more complex scenarios
  - ✗ If mapfile is an executable program, it is run when needed
    - ✗ The program is given one argument, the *key*
    - ✗ The program must return either:
      - ✗ One line of a map for use by the automounter
      - ✗ Nothing

# NFS and Automounting

## □ See Also

- ✗ O'Reilly's *Managing NFS and NIS*
- ✗ CIS8C1 Lecture 9
  - ✗ <http://cis68c1.mikecappella.com>
- ✗ *NFS HowTo*
  - ✗ <http://www.tldp.org/HOWTO/NFS-HOWTO/index.html>
- ✗ Man pages
  - ✗ nfsd(8) – NFS server process
  - ✗ mountd(8) – NFS mount daemon
  - ✗ exportfs(8) – management of NFS export list
  - ✗ nfsstat(8) – NFS / RPC statistics utility

# NFS and Automounting

## □ See Also

### ✘ Man pages (*continued*)

- ✘ rpcinfo(8) – RPC information utility
- ✘ fstab(5) – file system information table
- ✘ autofs(5) – autofs configuration file format
- ✘ autofs(8) – control script /etc/init.d/autofs
- ✘ auto.master(5) – master configuration file format
- ✘ automount(8) - configures mount points for autofs
- ✘ mount(8) – the mount command (for mount options)