Freescale Semiconductor
Application Note

AN2913
Rev. 1, 07/2005

Example of a Linux Cluster

by *Maurie Ommerman*
Computing Platform Division
Freescale Semiconductor, Inc.
Austin, TX
risc10@freescale.com

This application note describes the steps to create and build a cluster of Genesi Pegasos (or any Linux combination) machines so that one file system can be shared and the same users can login to each machine. In this particular case, two machines are common and one machine has a separate user structure. This example is the cluster we use for development, however, all the names and IP addressees have been changed.

NOTE

For the purposes of this application note, the arrangement of Genesi Pegasos machines is referred to as a cluster. The term cluster is used to describe a group of computers that can share a home directory, therefore, any user, no matter which machine in the cluster they are logged into, can access their home directories.

Contents

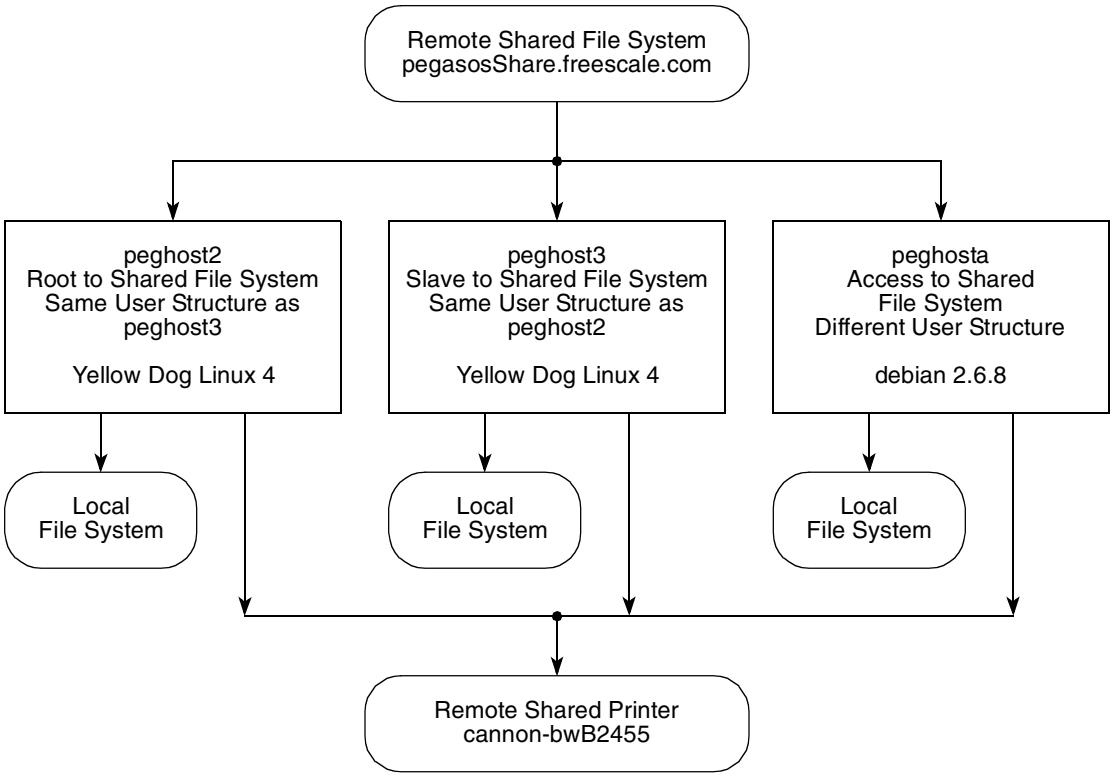
freescale™ semiconductor

1 Introduction

Disclaimer: There are many ways to accomplish the same task on Linux; this is an example of one way to build a specific type of cluster.

It is useful to understand how to administer a series of Linux machines so that a single file system can be shared and backed up.

The following diagram shows how our cluster is designed:



2 Terminology

The following terms used in this document are described below:

Term	Meaning
Linux OS	Linux Operating System
Yellow Dog	One of the distributions of Linux.
Debian	One of the distributions of Linux.
aka	also know as
DNS	Domain name service – a computer whose function is to relate computer names to their fixed IP addresses
Daemon	Small program that is always running in the background, usually waiting for activity. For example, a network daemon is waiting for activity on the network.
cluster	For the purposes of this paper, a cluster is a group of computers that is organized together for some purpose—in this case, as a remote development system.

nfs	network file system, i.e. remote file system – a protocol for making a file system available to remote hosts via a network.
IT	For the purposes of this paper, IT stands for Information Technology and is loosely applied to the staff of people who manage the site computers, servers, internet and other services.
ssh	secure shell – a secure method of connecting to a linux machine from a remote computer, which may or may not be running Linux.
scp	secure copy – a secure method of copying files from a linux computer to a remote computer.
sftp	secure file transfer protocol – a method of interactively sending/receiving files from one Linux computer to another computer, which may or may not be running Linux.
OF firmware	Open Firmware, OF, is the firmware used on the Genesi Pegasos computers. The firmware is responsible for bringing the hardware up to a sane state and starting the Linux Operating System.

3 Hardware

The hardware consists of three Genesi Pegasos computers—two running Yellow Dog Linux 4 and one running Debian. There is one remote file system that is mounted on all three machines. All three computers must have static IP addresses and must be accessible via DNS.

Each computer has its own local file system. Thus, changing to the / directory will allow one to access the local hard drive. The root user's home directory /root is on the local hard drive, as well as all the control files in /etc. In fact, these directories on all the machines (peghost2, peghost3, and peghosta) running Yellow Dog Linux 4 or Debian reside on the local hard drive.

All these directories on peghost3 running Yellow Dog Linux reside on the hard drive:

```
[user1@peghost3 /]$ ls -F
bin/   etc/            lib/         mnt/   root/ tmp/
boot/  home_original/  lost+found/ sys/      usr/
dev/   initrd/         misc/        proc/  sbin/  test/     var/
```

All these directories on peghosta running Debian reside on the local hard drive:

```
user1@debian:/$ ls -F
dev/     lib/         pegasos sys/       usr/
bin/             etc/     lost+found/  proc/    tftpboot/  var/
boot/            floppy/  media/       root/    tmp/
cdrom/           home/    mnt/         sbin/    usb/
cdrom0/          initrd/ srv/     usb4/
```

Notice that the original /home directory on peghost2 and peghost3 has been renamed to /home_original. This is further discussed in the individual sections on mounting the shared file system.

The original /home directory on peghosta has not been moved to /home_original.

Only the user home directories reside on the remote shared file system, which is mounted at /mnt/nfs/home.

For the purposes of this example, all the computers will share one remote printer.

NOTE

It is not necessary for all the computers to share one printer. They could share several printers or have local printers.

4 Setting Up the Shared File System

In our environment, a request was made to the IT department, who allocated a file system from one of their secure and backed up servers, then gave us the access location. The access location is in the format <server>:/<directory>. For this example, call the server pegasosShare.freescale.com and call the directory entry /proj02/pegasos so it can be mounted as pegasosShare.freescale.com:/proj02/pegasos. The complete mount command is:

```
mount -t nfs pegasosShare.freescale.com:/proj02/pegasos /mnt/nfs/home.
```

Root must create the mount point, /mnt/nfs/home. Any mount point can be used, but, for this example, use /mnt/nfs/home.

5 Setting Up the First Computer

In this example, we will configure peghost2 as the master computer. This computer will be running Yellow Dog Linux 4.

NOTE

peghost2 is the root master for the shared file system, therefore, peghost2 is the only computer in the cluster that can create new directories on the shared file system.

5.1 Setting Up the Fixed IP Address for Yellow Dog Linux

Yellow Dog Linux and Debian use a different set of files to set up Ethernet IP addresses. Since peghost2 is running Yellow Dog Linux, this description is for Yellow Dog Linux systems, which is the same structure as Red Hat and Mandrake. peghosta is running Debian, which is described in Section 7, “Setting Up the Third Computer.”

For Yellow Dog Linux and its similar distributions, the two network files /etc/sysconfig/network and /etc/network-scripts/ifcfg-eth0 are used.

The /etc/sysconfig/network file describes the hostname and whether networking is turned on. In this case, the hostname is peghost2.am.freescale.net:

```
[user1@peghost2 sysconfig]$ cat /etc/sysconfig/network

NETWORKING=yes

HOSTNAME=peghost2.am.freescale.net
```

The /etc/sysconfig/network-scripts/ifcfg-eth0 describes the name, protocol, and type of ethernet:

```
[user1@peghost2 user1]$ cat /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0

BOOTPROTO=static

IPADDR=10.82.119.188
```

```
NETMASK=255.255.252.0
GATEWAY=10.82.119.254
ONBOOT=yes
TYPE=Ethernet
```

The IP address, netmask, and gateway are specific to a site, thus, these will most likely not be the same on another site. The protocol is static and the network is started on boot (ONBOOT=yes).

Once the network files are properly configured, then the network can be started with these two commands:

```
ifdown eth0
ifup eth0
```

Also, whenever the computer is booted, Ethernet will be started automatically using these IP addresses. Remember that it is necessary for the computer to be known to the DNS servers and to have static IP addresses.

The `ifconfig` command will display the network parameters:

```
[root@peghost2 root]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0B:2F:42:8B:0C
          inet addr:10.82.119.188  Bcast:10.82.119.255  Mask:255.255.252.0
          inet6 addr: fe80::20b:2fff:fe42:8b0c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5688 errors:0 dropped:0 overruns:0 frame:0
          TX packets:723 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:536647 (524.0 Kb)  TX bytes:112588 (109.9 Kb)
          Interrupt:9 Base address:0x800

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2252 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2252 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0

          RX bytes:2617808 (2.4 Mb)  TX bytes:2617808 (2.4 Mb)
```

5.2 Mounting the Shared File System

On these two identical systems, peghost2 and peghost3, the shared file system will become the home directory for all users. In our case, these machines have been used previously and users had their home directories mounted on /home. In order to allow these users to get access to their original /home directory, /home was renamed as /home_original. All the directories and files were given full permissions because when these users were recreated using the shared file system as /mnt/nfs/home, the user IDs may have changed and the users would not have access to their original home directories. See Section 5.4, “Creating Users,” for more information.

By convention, nfs mounts (i.e. remote mounts) are mounted to a mount point in /mnt/nfs. Since this shared file system will be used as the home directory for all users, it makes sense to mount them to the mount point /mnt/nfs/home.

This mount point needs to be created by the root user on peghost2:

```
cd /mnt

mkdir nfs (only if it does not currently exist)

cd nfs

mkdir home (only if it does not currently exist)

cd
```

Temporarily mount the shared file system onto this mount point:

```
mount -t nfs pegasosShare.freescale.com:/proj02/pegasos /mnt/nfs/home
```

At this point, the nfs is mounted at /mnt/nfs/home.

5.3 Creating Groups

The group must be created on peghost2. Users and groups must be created here first because only peghost2 has root permission to create directories and files on the shared file system /mnt/nfs/home.

For the purposes of this application note, each user must be a member of the same group so that they can share files. The linux file system controls access to file and directories based on a permission system. The permissions are displayed by the `ls -l` command as a set of octal triples. The triplet ‘rwx’ means read, write, execute permissions.

```
[user1@peghost2 user1]$ ls -l

total 24

drwxr-xr-x   8 user1 taiga 1024 Mar  9 18:02 dink

drwxr-xr-x  19 user1 taiga 1024 Mar  9 18:40 linuxppc_2_5_devel

-rw-r--r--   1 user1 root  2623 Mar 10 14:33 samba.list
```

The first character indicates the type of file, where ‘d’ indicates a directory, and ‘-’ indicates a file. The first triplet following the first character represents owner permission, the second triplet is group permissions, and the last triplet is for anybody permission. For the example above, ‘dink’ is a directory and the owner has all permissions, the group has only read and execute permission (the execute in this case indicates that the group members can descend into the directory), and any user also has read and execute permission. samba.list is a file, owner has read and write permission, it is not an executable file, and group and any user have only read permission.

This triplet can also be represented as an octal number, 0–7, where each bit represents the r, w, or x, as in the following example: o1 = 001= --x, o4 = 100 = r--, and o2= 010 = -w-. Combinations such as o6 = 110 = rw- are possible.

NOTE

An ‘o’ preceding a number means an octal number. For example, o6 means an octal 6, which is binary 110.

The default permission set when a user creates a new file is determined by the umask, which can be changed by the user using the `umask` command. The `umask` command indicates which bits should be zero. Thus, the default umask is 0022—for the 022, the 0 for user indicates rwx for executable files and rw- for non executable files, which is determined by the application that creates the file, the 2 for group indicates r-x for executable files and r-- only for non executable files, and the 2 for others indicates r-x for executable files and r-- only for non executable files.

A file can have its permission changed by using the `chmod` command. Either a numeric argument or a text argument can be used.

`chmod 744`, or its equivalent `chmod u+r u+w u+x g+r o+r`, changes the permissions to rwxr--r--.

The previous discussion explains the reason that all users on this system are in the same group is so that files and directories can control access for group members.

In this cluster, create the group “taiga”.

This command will add the group named taiga to the system file /etc/group and assign it the group ID 400. The command `info groupadd` describes this command in detail:

```
groupadd <group name>

groupadd taiga -g 400
```

This command will list the group members:

```
[user1@peghost2 user1]$ cat /etc/group

root:x:0:root

bin:x:1:root,bin,daemon

.... many others ....

taiga:x:400:
```

5.4 Creating Users

The users must be created on peghost2 since only peghost2 has root permission to create directories and files on the shared file system /mnt/nfs/home. Thus, only peghost2 can create the users and create their home directories.

The `useradd` command is used to create the users on this system. The command `info useradd` describes this command. For the purposes of this application note, users must be created using the home directory /mnt/nfs/home and the group membership of taiga with an initial password. To make things easy, just set up each user’s password as their user name and let them change the password the first time they log in. The command to use is:

```
useradd -d /mnt/nfs/home/username -g taiga -p <password which is username>
username.
```

This example will create the first user, assign the home directory in /mnt/nfs/home, and set the password equal to the user name:

```
useradd -d /mnt/nfs/home/user1 -g taiga -p user1 user1
```

NOTE

The original /home directory has been moved to /home_original and each user will now have his/her home directory on the shared file system /mnt/nfs/home as determined by the -d (home directory) parameter.

This is not necessary since each user will have their home directory in /mnt/nfs/home, /home could be preserved and still used for local-only users. In this case, there are no local-only users and it is more clear to rename /home.

A script can be built with the following format which will create all the users automatically:

```
edit/create the file, makeuser

insert the useradd command for each user desired.
```

The `edit` command creates text files, thus the permission set for this will be rw-r--r--. In order to use this script, change the permission set to rwxr--r-- with the command `chmod 744 makeuser` or, alternately, `chmod u+x makeuser`.

Execute the script with the ./ preface (./makeuser). The ./ preface indicates to execute the script from the current directory.

The makeuser script looks like this:

```
[root@pegasus2 root]# cat makeuser

useradd -d /mnt/nfs/home/user1 -g taiga -p user1 user1

useradd -d /mnt/nfs/home/user2 -g taiga -p user2 user2

useradd -d /mnt/nfs/home/user3 -g taiga -p user3 user3

useradd -d /mnt/nfs/home/user4 -g taiga -p user4 user4

useradd -d /mnt/nfs/home/user5 -g taiga -p user5 user5

useradd -d /mnt/nfs/home/user6 -g taiga -p user6 user6

useradd -d /mnt/nfs/home/user7 -g taiga -p user7 user7

useradd -d /mnt/nfs/home/user8 -g taiga -p user8 user8

useradd -d /mnt/nfs/home/user9 -g taiga -p user9 user9

useradd -d /mnt/nfs/home/user10 -g taiga -p user10 user10

useradd -d /mnt/nfs/home/user11 -g taiga -p user11 user11

useradd -d /mnt/nfs/home/user12 -g taiga -p user12 user12

useradd -d /mnt/nfs/home/user13 -g taiga -p user13 user13
```

It is a good idea to have another script to delete all the users, in case you messed up and need to try again. The entries in this script are of the form:

```
userdel -r user1
```

It can be called deluser and invoked as ./deluser.

The `-r` command removes the home directory as well.

The deluser script looks like this:

```
[root@pegasus2 root]# cat deluser
userdel -r user1
userdel -r user2
userdel -r user3
userdel -r user4
userdel -r user5
userdel -r user6
userdel -r user7
userdel -r user8
userdel -r user9
userdel -r user10
userdel -r user11
userdel -r user12
userdel -r user13
```

The `useradd` command and/or script will create all the users in your system, which updates the /etc/passwd and /etc/shadow files, and creates all the user home directories on the /mnt/nfs/home mount point. Thus, the /mnt/nfs/home mount point will look like this after these users are created:

```
[root@peghost2 root]# ls -l /mnt/nfs/home
total 112
drwxr-x---   3 user1 taiga 1024 Mar  9 16:38 user1
-rw-r--r--   1 root    bin    751 Mar  8 14:47 Contents
drwxr-x---   5 user2  taiga 1024 Mar 10 16:26 user2
drwxr-x---   3 user3    taiga 1024 Mar  9 16:49 user3
drwxr-x---   3 user4   taiga 1024 Mar  9 16:49 user4
drwxr-x---   3 user5   taiga 1024 Mar  9 16:49 user5
drwxr-x---   3 user6  taiga 1024 Mar  9 16:49 user6
drwxr-x---   3 user7    taiga 1024 Mar  9 16:50 user7
```

```
drwxr-x---   3 user8     taiga 1024 Mar  9 16:49 user8
drwxr-x---  12 user9  taiga 1024 Mar 15 17:12 user9
drwxr-x---   4 user10 taiga 1024 Mar  9 16:51 user10
drwxr-xr-x   2 root    root    80 Mar 15 13:05 pegfilea
drwxr-x---   4 user11     taiga 1024 Mar  9 17:03 user11
drwxr-x---   7 user12     taiga 1024 Mar 15 11:37 user12
drwxr-x---   4 user13     taiga 1024 Mar  9 17:38 user13
```

Notice that each user has an entry, each user is a member of the taiga group, and directory permissions are set to read, write, execute for the home directory, and read, execute for the group taiga. Thus, each user can see into other users' home directories, but cannot create any files or directories into other user home directories. The file 'Contents' is not a home directory, it is owned by root and group bin, and all users can read the file. In this case, this file describes the contents of this shared directory. This file was supplied by the IT staff. The directory entry, pegfilea, was created on peghost2 because it has root permission on the shared file system. This will become clear in the section on peghosta, Section 7, "Setting Up the Third Computer." Since peghosta does not have root permission on the shared file system, it cannot read or write to any entry, so the directory entry pegfilea is owned by root, and root is always User ID (UI) 0, peghosta can read and write to this lone directory. When the group taiga is added on peghosta, root will be added to this group, which will then give root on peghosta read access to all the directories.

NOTE

NFS maps the user 'root' to the user 'nobody', unless told to do otherwise, i.e. root equivalency.

5.5 Making the Mount Permanent

Mounting is controlled with the /etc/fstab file. This file is read during the boot process and the mount points are mounted during boot, so that they are available when Linux starts, allowing users access to their home directory when they log in. The format of the file is explained in the `info fstab` command.

The listing below is a fairly typical fstab file:

```
[root@peghost2 root]# cat /etc/fstab
/dev/hda6                /                       ext3    defaults        1 1
none                     /dev/pts                devpts  gid=5,mode=620  0 0
none                     /dev/shm                tmpfs   defaults        0 0
none                     /proc                   proc    defaults        0 0
none                     /sys                    sysfs   defaults        0 0
/dev/cdrom           /mnt/cdrom           udf,iso9660 noauto,owner,kudzu,ro 0 0
```

The entry for the /mnt/nfs/home directory is added as the second line of the file and now looks like this.

```
[root@peghost2 root]# cat /etc/fstab
/dev/hda6                /                       ext3    defaults        1 1
```

```
pegasosShare.freescale.com:/proj02/pegasus /mnt/nfs/home nfs defaults 1 1
none                    /dev/pts                devpts  gid=5,mode=620  0 0
none                    /dev/shm                tmpfs   defaults        0 0
none                    /proc                   proc    defaults        0 0
none                    /sys                    sysfs   defaults        0 0
/dev/cdrom          /mnt/cdrom          udf,iso9660 noauto,owner,kudzu,ro 0 0
```

The line added corresponds directly to the mount command used earlier:

```
pegasosShare.freescale.com:/proj02/pegasus /mnt/nfs/home nfs defaults 1 1
```

The first field, pegasosShare.freescale.com:/proj02/pegasus, is the server name pegasosShare.freescale.com and the directory on that server, /proj02/pegasus.

The second field, /mnt/nfs/home is the mount point.

The third field nfs is the file system type, corresponding to the -t in the mount command.

The fourth field indicates to use the default mount options.

The fifth and sixth fields are used by dump and fsck file dump and file check utilities at boot time.

Compare the fstab entry just created to the mount command used previously:

```
mount -t nfs pegasosShare.freescale.com:/proj02/pegasos /mnt/nfs/home.
```

The next time the system is booted, this mount point will be automatically mounted, assuming there are no errors (otherwise, it will not be mounted).

5.6 Setting Up a Remote Printer

For this application note, set up the same remote printer for each member of the cluster. This procedure is done for each machine.

The printer daemon and driver is cups, cupsd, or cupsys, all of which control printers, and any one of these drivers may be on your system. The configuration for printers is controlled by the driver via a listening device on the local system, aka localhost. This localhost is specified in the /etc/host files and on Yellow Dog Linux in the /etc/sysconfig/network file.

```
[root@peghost2 cups]# cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1               localhost.localdomain localhost
10.82.119.188           peghost2 peghost2.am.freescale.net
[root@peghost2 cups]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=peghost2.am.freescale.net
```

The method used to interface to the printer drivers is via the localhost port 631. Thus, to configure a printer, start a web browser, such as mozilla, then navigate to the http://localhost:631.

From a remote login (i.e. from a ssh connection instead of on the system console) export the display IP address and then start mozilla.

In this example, the remote computer has the IP address shown below:

```
export DISPLAY=10.82.124.155:0.0
```

Start mozilla:

```
mozilla&  (& says run in background)
```

NOTE

If the user forgot to type in the '&', then he/she can hit 'cnt-z' followed by 'bg' to background it. 'cnt-z' alone will only stop the job.

Start the printer configuration:

```
http://localhost:631
```

Choose add a printer and specify the information requested. As an example, this printer was configured:

```
cannon-bwB2455

AppsSocket/HP jet Direct

LaserJet Series cups v1.1

URI is socket://10.82.119.224:9100

Driver is HP 4SI/4SI (only if asked)
```

The directory /etc/cups contains many useful files describing the printer configuration:

```
[root@peghost2 cups]# ls -F

certs/        cupsd.conf       mime.convs  ppds.dat

classes.conf mime.types  printers.conf    pstoraster.convs

client.conf   interfaces/      ppd/         printers.conf.O
```

The interesting files are cupsd.conf, used to configure the cups driver, and printers.conf, the file created by the printer driver after a printer configuration is completed as above.

```
[root@peghost2 cups]# cat /etc/cups/printers.conf

# Printer configuration file for CUPS v1.1.20

# Written by cupsd on Thu 10 Mar 2005 05:39:46 PM CST

<DefaultPrinter cannon-bwB245>

Info black and write printer

Location PR B2455

DeviceURI socket://10.82.119.224:9100

State Idle
```

```
Accepting Yes

JobSheets none none

QuotaPeriod 0

PageLimit 0

KLimit 0

</Printer>
```

Removing printers should be done via the printer configuration shown above as http://localhost:631, but if the configuration is bad or too complicated to remove one printer at a time, the /etc/cups/printers.conf file can be removed and the root user can then start configuring printers again.

The cupsd.conf file defines access type and users. It is a rather large file. It is described by the `info cupsd.conf` and `info cupsd` commands.

The cups driver itself can be stopped, started, or restarted with the command `/etc/init.d/cups stop/start/restart` (or `/etc/init.d/cupsd` or `/etc/init.d/cupsys`, depending on which driver this Linux system is using).

The following commands are available to use the printer (all are described in their corresponding `info` command):

```
lp  <file>,  lpstat,  lpq, enscript
```

These are the files in /etc/cupsd for Yellow Dog Linux and in /etc/cupsys for Debian:

- cupsd.conf for setting up who can talk to it on port 631
- printers.conf for printer info (delete if you want to start over)
- /etc/init.d/cups restart will restart the deamon.

5.7 Rebooting and Testing

Use the command `shutdown -r now` to reboot the first computer, peghost2. The Genesi Pegasos computers boot by default item 6, which is Yellow Dog Linux.

When peghost2 is back up and running, verify that all the users created can log in and that the shared file system is accessible by all users. A simple method would be the following:

1. Log in as user1 (i.e. the first user on the list):

   ```
   pwd (assure that it is the /mnt/nfs/home/user1 directory)

   ls -a (assure that all the expected files are shown)

   su - user2 (the next user on the list)
   ```

2. Enter password:

   ```
   pwd

   ls
   ```

3. Repeat this sequence for all other users.
4. If there is any problem logging in, use the `moduser` command to fix it:

   ```
   usermod -d < home directory> -G <group> -p <password>,
   ```

   changing only the values that are wrong.

- `ls -l mnt/nfs/home` will display all the users and home directories.
- `cat /etc/passwd` will display all the users, verify that all the users exist in this file.
- `cat /etc/group` will display all the groups, verify that the group, taiga, is in this list.

5. Finally, change the root password for peghost2 and in the next section change the root password for peghost3 to the same password.

6 Setting Up the Second Computer

In this example we will configure peghost3 as the slave computer. This computer will be running Yellow Dog Linux 4.

NOTE

peghost3 is one of the slave computers for the shared file system. peghost3 can only access existing directories, however, it can create subdirectories in existing directories on the shared file system if the user ids and/or group ids match.

6.1 Setting Up the Fixed IP Address for Yellow Dog Linux

This procedure is identical to Section 5.1, “Setting Up the Fixed IP Address for Yellow Dog Linux,” except that the name and IP address are different. The section is repeated here with the new name and IP address. See Section 7.1, “Setting Up the Fixed IP Address for Debian Linux,” for the instructions to set up the IP address for Debian Linux.

Yellow Dog Linux and Debian use a different set of files to set up Ethernet IP addresses. Since peghost3 is running Yellow Dog Linux, this description is for Yellow Dog Linux systems, which is also the same structure as Red Hat and Mandrake. peghosta is running Debian, so the set for Debian will be described in Section 7, “Setting Up the Third Computer”.

For Yellow Dog Linux and its similar distributions, the two network files, /etc/sysconfig/network and /etc/network-scripts/ifcfg-eth0 are used.

The /etc/sysconfig/network file describes the hostname and whether networking is turned on. In this case, the hostname is peghost3.com.am.freescale.net.

```
[user1@peghost3 sysconfig]$ cat /etc/sysconfig/network

NETWORKING=yes

HOSTNAME=peghost3.am.freescale.net

The /etc/sysconfig/network-scripts/ifcfg-eth0 describes the name, protocol and
type of ethernet.

[user1@peghost3 user1]$ cat /etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE=eth0

BOOTPROTO=static

IPADDR=10.82.119.189

NETMASK=255.255.252.0

GATEWAY=10.82.119.254
```

```
ONBOOT=yes

TYPE=Ethernet
```

The IP address, netmast, and gateway are specific to a site, thus, these will most likely not be the same on another site. The protocol is static and the network is started on boot (ONBOOT=yes).

Once the network files are properly configured, then the network can be started with these two commands:

```
ifdown eth0

ifup eth0
```

Also, whenever the computer is booted, Ethernet will be started automatically using these IP addresses. Remember, it is necessary to have static IP addresses for a computer to be known to the DNS servers.

The `ifconfig` command will display the network parameters:

```
[root@peghost2 root]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0B:2F:42:8B:0C
          inet addr:10.82.119.189 Bcast:10.82.119.255  Mask:255.255.252.0
          inet6 addr: fe80::20b:2fff:fe42:8b0c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5688 errors:0 dropped:0 overruns:0 frame:0
          TX packets:723 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:536647 (524.0 Kb)  TX bytes:112588 (109.9 Kb)
          Interrupt:9 Base address:0x800


lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2252 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2252 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0

      RX bytes:2617808 (2.4 Mb)  TX bytes:2617808 (2.4 Mb)
```

6.2 Mounting the Shared File System

This section is identical to Section 5.2, “Mounting the Shared File System”.

6.3 Copying the Control Files

Since this is a computer that will be used by the same set of users, it needs identical group and user invocations. The easiest way to accomplish this is to copy the three control files, /etc/passwd, /etc/group, and /etc/shadow from the first computer to the second.

An example of transferring files for a non root user is shown below:

```
[user1@peghost3 user1]$ scp user1@peghost2:samba.list.

The authenticity of host 'peghost2 (10.82.118.188)' can't be established.

RSA key fingerprint is 3a:7d:73:1a:f5:b4:21:2f:e7:67:55:dd:c8:25:c0:20.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added 'peghost2,10.82.118.188' (RSA) to the list of known
hosts.

user1@peghost2's password:

test                                          100%   23    52.4KB/s   00:00
```

For this case, transfer the three control files as root user.

First save the existing files:

```
mv /etc/passwd /etc/passwd_orig

mv /etc/shadow /etc/passwd_orig

mv /etc/group /etc/group_orig
```

Then do the remote copies either with scp or sftp:

```
scp root@peghost2:/etc/passwd /etc/passwd

scp root@peghost2:/etc/shadow /etc/shadow

scp root@peghost2:/etc/group /etc/group
```

6.4 Making the Mount Permanent

One could copy the /etc/fstab file from peghost2, however, peghost3 may have some other devices mounted different from peghost2, so it is better to just modify the /etc/fstab file. The procedure is identical to Section 5.5, “Making the Mount Permanent”.

6.5 Setting Up a Remote Printer

This is identical to Section 5.6, “Setting Up a Remote Printer”.

6.6 Rebooting and Testing

Use the command `shutdown -r now` to reboot the first computer, peghost3. The Genesi Pegasos computers boot by default item 6, which is Yellow Dog Linux.

When peghost3 is back up and running, verify that all the users created can log in and that the shared file system is accessible by all users.

ls -l mnt/nfs/home will display all the users and home directories.

A simple method would be the following:

1. Log in as user1 (i.e. the first user on the list):

   ```
   pwd (assure that it is the /mnt/nfs/home/user1 directory)

   ls -a (assure that all the expected files are shown)

   su - user2 (the next user on the list)
   ```

2. Enter password:

   ```
   pwd

   ls
   ```

3. Repeat this sequence for all the other users.
4. If there is any problem logging in, use the `moduser` command to fix it:

   ```
   usermod -d < home directory> -G <group> -p <password>, changing only the values
   that are wrong.
   ```

   — `cat /etc/passwd` will display all the users, verify that all the users exist in this file.
   — `cat /etc/group` will display all the groups, verify that the group, taiga, is in this list.
5. Finally, change the root password for peghost3 to the same password as peghost2.

7 Setting Up the Third Computer

In this example, we will configure peghosta as a computer with a different login structure (i.e. different set of users) and mount the shared file system for read only access. This computer will be running Debian Linux.

7.1 Setting Up the Fixed IP Address for Debian Linux

Debian Linux is very different in setting up IP addresses. Whereas Section 6.1, “Setting Up the Fixed IP Address for Yellow Dog Linux,” described the method for Yellow Dog Linux and Red Hat, this section will describe the method of Debian.

The Debian network setup is maintained by two files, /etc/hostname and /etc/network/interfaces. The command `man interfaces` describes this file:

```
Look at /etc/hostname
```

```
user1@debian:~$ cat /etc/hostname

debian
```

In Debian the hostname file defines the computer hostname, whereas in Yellow Dog Linux, the hostname was defined in the /etc/sysconfig/network file.

Look at, cat, or Edit this file:

```
root@debian:/etc/network# cat /etc/network/interfaces

auto lo

iface lo inet loopback


# This entry was created during the Debian installation

auto eth0

#iface eth0 inet dhcp

#auto eth0

iface eth0 inet static

        address 10.82.127.201

        netmask 255.255.252.0

        gateway 10.82.127.254
```

The lines preceded by the pound sign '#' are commented out, thus, the dynamic dhcp protocol is commented out. The next line starting with `iface` indicates a static IP address, which is what is needed for peghosta to be in the DNS tables.

Note that peghost2 has an IP address of 10.82.119.188, peghost3 has an IP address of 10.82.119.189, and peghosta has an IP address of 10.82.127.201. It is not necessary for the IP address to be sequential, or even in the same subnet, as peghosta is in the 127 subnet, while peghost2 and peghost3 are in the 119 subnet.

7.2 Mounting the Shared File System

This section is identical to Section 5.2, "Mounting the Shared File System".

7.3 Adjusting the Group ID

In order for root and other users to access the shared system, the group ID must be adjusted. Once /mnt/nfs/home is mounted, it can be accessed. However, since this system has a whole set of different users and groups than peghost2 and peghost3, the system does not know the owners, thus, it can only print out the user IDs and the group IDs because no users in peghosta's passwd file match the user IDs. If, coincidentally, a user matched the user ID of any of these directories, then that user name would be displayed.

The list display of /mnt/nfs/home looks like this:

```
root@debian:~# ls -l

total 112
```

```
-rw-r--r--    1 root     daemon        751 Mar  8 14:47 Contents
drwxr-x---    3 500      400          1024 Mar  9 16:38 user1
drwxr-x---    5 508      400          1024 Mar 10 16:26 user2
drwxr-x---    3 501      400          1024 Mar  9 16:49 user3
drwxr-x---    3 505      400          1024 Mar  9 16:49 user4
drwxr-x---    3 509      400          1024 Mar  9 16:49 user5
drwxr-x---    3 504      400          1024 Mar  9 16:49 user6
drwxr-x---    3 502      400          1024 Mar  9 16:50 user7
drwxr-x---    3 503      400          1024 Mar  9 16:49 user8
drwxr-x---   14 511      400          2048 Mar 16 14:46 user9
drwxr-x---    4 512      400          1024 Mar  9 16:51 user10
drwxr-xr-x    2 root     root           80 Mar 15 13:05 pegfilea
drwxr-x---    4 507      400          1024 Mar  9 17:03 user11
drwxr-x---    7 510      400          1024 Mar 16 13:52 user12
drwxr-x---    4 506      400          1024 Mar  9 17:38 user13
```

The /etc/group file does not contain an entry for the group taiga. Thus, it is necessary to create such a group with the group ID of 400.

```
addgroup -gid 400 taiga
```

This will change the listing of /mnt/nfs/home to look like this:

```
root@debian:~# ls -l
total 112
-rw-r--r--    1 root     daemon        751 Mar  8 14:47 Contents
drwxr-x---    3 500      taiga        1024 Mar  9 16:38 user1
drwxr-x---    5 508      taiga        1024 Mar 10 16:26 user2
drwxr-x---    3 501      taiga        1024 Mar  9 16:49 user3
drwxr-x---    3 505      taiga        1024 Mar  9 16:49 user4
drwxr-x---    3 509      taiga        1024 Mar  9 16:49 user5
drwxr-x---    3 504      taiga        1024 Mar  9 16:49 user6
drwxr-x---    3 502      taiga        1024 Mar  9 16:50 user7
drwxr-x---    3 503      taiga        1024 Mar  9 16:49 user8
drwxr-x---   14 511      taiga        2048 Mar 16 14:46 user9
drwxr-x---    4 512      taiga        1024 Mar  9 16:51 user10
```

```
drwxr-xr-x    2 root     root           80 Mar 15 13:05 pegfilea
drwxr-x---    4 507      taiga        1024 Mar  9 17:03 user11
drwxr-x---    7 510      taiga        1024 Mar 16 13:52 user12
drwxr-x---    4 506      taiga        1024 Mar  9 17:38 user13
```

However, no user or even root can access these directories because no user or root on peghosta is a member of the group taiga.

To understand why root can't access these files, recall the following note from Section 5, "Setting Up the First Computer":

NOTE

peghost2 is the root master for the shared file system, therefore, peghost2 is the only computer in the cluster that can create new directories on the shared file system.

Hence, root on peghosta can not read or create any files on this shared file system unless peghosta's root is the owner. The file /mnt/nfs/home/pegfilea is owned by root and, even though it was created on peghost2, root user ID is always 0 on all Linux systems. Therefore, root has access to the two files, Contents and pegfilea.

NOTE

NFS maps the user 'root' to the user 'nobody' unless told to do otherwise, i.e. root equivalency

In order to give peghosta root read access to all the directories, it is necessary to add root to the taiga group list, and the same for any user, such as user1.

```
usermod -G taiga root
usermod -G taiga user1
```

Displaying the /etc/group file shows that root and user1 are added to the group, taiga:

```
root@debian:~# cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3
........ intervening lines are removed.....:
taiga:x:400:user1,root
```

Also, the `groups` command will show what group members are for a particular user:

```
root@debian:~# groups
root taiga
user1@debian:~$ groups
user1 taiga
```

After these changes, root and user1 can access and read these directories, but they do not have permissions to write files or create directories in these home directories. That could be granted by changing the permissions on peghost2 by the owner of the home directory or root, or on peghost3 by the owner of the directory, but not by root and, of course, no one can change the permissions on peghosta. Thus, peghosta can see all the user directories.

7.4 Making the Mount Permanent

One could copy the /etc/fstab file from peghost2, however, peghosta may have some other devices mounted different from peghost2, so it is better to just modify the /etc/fstab file. The procedure is identical to Section 5.5, “Making the Mount Permanent”.

7.5 Setting Up a Remote Printer

This is identical to Section 5.6, “Setting Up a Remote Printer,” however, under this release of debian, the cups system driver is cupsd, so the command to start, stop, and restart cups is:

```
/etc/init.d/cupsd
```

7.6 Changing the Default Boot in the OF Firmware

Since all these cluster machines will be used remotely and this entire exercise has been done remotely, it is now necessary to change the default boot for this Genesi Pegasos machine. The command `shutdown -r now` will shutdown the current Linux kernel and reboot into the OF firmware of the machine. The firmware, which must be run at the local console and not remotely, will run the menu file, which will determine the default boot. As delivered, the default boot is to item 6, Yellow Dog Linux, which is fine for the other two machines in this cluster. However, for this machine, it is necessary to change the default to boot item 4, Debian. For this example, peghost2 and peghost3 will run Yellow Dog Linux, and this machine peghosta will run Debian Linux.

The menu file resides on the first partition of the hard drive, which is /dev/hda1. This partition is an ext3 partition, so it can be mounted on peghosta, which is running Debian. Mount this partition on /mnt/hd. If /mnt/hd does not exist, create it first:

```
cd /mnt

mkdir hd

mount /dev/hda1 /mnt/hd

cd /mnt/hd
```

Edit the menu file (for this paper, use vi):

```
vi menu


\ FORTH is identified by a forth comment at first line

\

\ terminal control stuff

\
```

```
: TTY.CSI d# 27 EMIT ASCII [ EMIT ;

: TTY.HOME    TTY.CSI ASCII H EMIT ;

: TTY.CLR_EOS TTY.CSI ASCII J EMIT ;

: TTY.HOME_CLR TTY.HOME TTY.CLR_EOS ;

\

\ boot menu stuff

\

: my-max-boot-num 7 ;

: my-boot-default 6;

: my-boot-delay d# 300 ; \ unit = 100 ms

: my-print-menu ( -- )

.............. the rest of the file is not shown here .........
```

Line 13, my-boot-default 6, needs to be changed to item 4, so change line 13 to look like this:

```
my-boot-default 4.
```

Save the file and unmount the mount point:

```
umount /mnt/hd.
```

7.7 Rebooting and Testing

Use the command `shutdown -r now` to reboot the third computer, peghosta. This Genesi Pegasos computer will boot by default item 4, which is Debian Linux, because of the change we made in Section 7.6, “Changing the Default Boot in the OF Firmware”.

When peghosta is back up and running, verify only that the shared file system is accessible by all users, since they all have a group of taiga.

NOTE

Since we did not copy the /etc/passwd and /etc/shadow file, peghosta still retains it’s original users and the shared file system is not the home directory of any of these users. The shared file system is only accessible as read only.

`ls -l mnt/nfs/home` will display all the home directories used by the other two computers.

`cat /etc/group` will display all the groups. Verify that the group, taiga, is in this list.

Finally, change the root password for peghosta to the same password as peghost2 and peghost3, or to another password, if that is desired.

8 Accessing Any Member of the Cluster

peghost2 and peghost3 can be logged into by a remote computer using the ssh protocol. For the purposes of this application note, a windows computer is used to remotely log into any of these cluster machines.

The command ssh can be used.

ssh peghost2 or ssh peghost3 will connect to the appropriate computer and any of the users can log in, including root, and have access to the same shared home directories. Any changes made to peghost2 by any user, other than root, will be reflected on the shared home directory. Thus, at any future time, the same user can log into the other cluster machine and the home directory will be the same. However, any user on peghosta, which has a different user structure, will be able to see these shared home directories but will not be able to write to them.

scp will allow a remote user to copy files from any of the cluster machines to any other Linux/Unix computer either in the cluster or not.

sftp will allow a remote user to interactively copy files from any of the cluster machines to any other Linux/Unix computer either in the cluster or not.

The equivalent facilities are available on window machines with the putty facility, which is freely downloaded from http://www.chiark.greenend.org.uk/~sgtatham/putty/. The equivalent commands in putty are `pssh`, `pscp`, and `psftp` from a console window. Putty also supplies a very nice remote login facility.

9 Ongoing activites for the existing cluster.

9.1 New members of the cluster.

After creating a new cluster member, you need to ensure that certain system users have the corresponding user id, ui for the particular machine. In particular, the user, gdm, which runs the x window display, must have ownership of its file system, which is /var/lib/gdm. Ensure that gdm is the owner. An easy way to do this is to use the command ls -ld /var/lib/gdm pegasus1:~# ls -ld /var/lib/gdm

drwxr-x--- 2 107 107 4096 May 16 22:14 /var/lib/gdm

If the owner and group is some number instead of gdm, then change it with these commands. chown gdm /var/lib/gdm chgrp gdm /var/lib/gdm

Then the command

ls -ld /var/lib/gdm will look like this.

pegasus1:~# ls -ld /var/lib/gdm

drwxr-x--- 2 gdm gdm 4096 May 16 22:14 /var/lib/gdm

The symptom of this problem is that after a reboot, the graphics terminal will not start and you will get an error indicating that gdm is not the owner of /var/lib/gdm.

9.2 Adding new users to an existing machine or a new machine.

This cluster concept does not implement a central method for controlling users. Each machine has it's own /etc/passwd, /etc/shadow, and /etc/group file. Thus it is necessary to control and coordinate the user id, ui, for all the users on each machine of the cluster. When the cluster was first built, this was controlled by copying these three files, /etc/passwd, /etc/shadow, and /etc/group from one machine to all the other machines. However, as more users are added, these files can get out of sync for new users.

When creating new users on an existing machine, it is good to control the user id, ui, of the new user. In the case of this paper, all users have ui's in the range of 500 to 599. Therefore, when adding a new user, force the ui to be in the range desired.

Look at the current user id's with the command cat /etc/passwd. Force the next user to be the next available ui, and ensure that this user has the same id on every machine in the cluster. Use the useradd or usermod command with the -u parameter to force the desired user id. useradd -d/mnt/nfs/home/username -g groupname -p password -u userid username where userid is the next user id number desired.

If we have this /etc/passwd file

guest:x:515:400::/mnt/nfs/home/guest:

pegasos:x:516:400::/mnt/nfs/home/pegasos:

user1:x:517:400::/mnt/nfs/home/user1/:

Looking at the /etc/passwd file above, the entries are as follows: username:x:userid:groupid::home directory

And we want to add a new user, user2, which follows user1, i.e has a user id of 518, then use this command.

useradd -d /mnt/nfs/home/user2 -g taiga -p user2 -u 518 user1

10 Summary

By following these steps, one can set up a cluster of any number of Linux computers sharing a file system and having identical user and groups on all machines and, alternatively, having some Linux computers sharing identical user and groups, and some Linux computers sharing only the file system. Any of the computers can have any Linux distribution, Yellow Dog Linux, Debian, or others and still share the file system and the user and group IDs. Additional Linux computers with the same user and share file structure can be easily added by following the steps outlined for peghost3 in Section 6, “Setting Up the Second Computer”. This paper arbitrarily chose Yellow Dog Linux for peghost2 and peghost3 and Debain for peghosta, however, all the steps are identical for both of these Linux distributions, except for the network files and the driver for cups. Thus, a Debian distribution could be added to the cluster with identical user structure and all steps in Section 6, “Setting Up the Second Computer,” would be valid except Section 6.1, “Setting Up the Fixed IP Address for Yellow Dog Linux”. In the Debian case, use Section 7.1, “Setting Up the Fixed IP Address for Debian Linux”.

This application note has shown how to create a cluster of Linux machines with two different access abilities.

10.1 Cluster Type 1

This paper has arbitrarily defined the cluster type 1 to be shared user, group, and home, i.e. identical user and group structure with a shared /home directory.

- peghost2
  - root user has read/write on /root and read/write on /mnt/nsf/home
  - all other users have read/write on their home directory and read on all the home directories.

- peghost3
  - root user only has read/write on /root
  - all other users have read/write on their home directory
  - and all users, including root, have read on all the home directories.
- peghost2 and peghost3
  - have all the system files and directories, including the tools chains on its respective local hard drive file system
  - and all users have their home directories on the /mnt/nfs/home shared file system.

10.2 Cluster Type 2

This paper has arbitrarily defined the cluster type 2 to be different user and group structure with the remote home directory accessible to all users.

- peghosta
  - root user only has read/write on /root
  - all users, including root, have only read on all the home directories.
- peghosta
  - has all the system files, directories, and its /home directory on the local hard drive file system
  - and read only access to all the shared home directories on /mnt/nfs/home.

10.3 Result

As mentioned above, any number of other Linux machines, Genesi Pegasos, or other manufactures' computers, even other architectures, can be added either to the first type or the second type in this cluster.

11 References

The following documents are referenced in this document:

1. AN2739: Genesi Pegasos II Debian Linux
2. AN2802: Genesi Pegasos II Yellow Dog 4 Linux
3. AN2738: Genesi Pegasos II Firmware

For assistance or answers to any questions on the information that is presented in this document, send an e-mail to risc10@freescale.com.

12 Document Revision History

Table 1 provides a revision history for this application note.

Table 1. Document Revision History

Rev. No.	Date	Substantive Change(s)
1	07/14/05	Inserted as Section 9, "Ongoing Activities for the Existing Cluster."
0	04/12/05	Initial release.

THIS PAGE INTENTIONALLY LEFT BLANK

THIS PAGE INTENTIONALLY LEFT BLANK

How to Reach Us:

USA/Europe/Locations Not Listed:
Freescale Semiconductor
Literature Distribution Center
P.O. Box 5405,
Denver, Colorado 80217
1-480-768-2130
(800) 521-6274

Japan:
Freescale Semiconductor Japan Ltd.
Technical Information Center
3-20-1, Minami-Azabu, Minato-ku
Tokyo 106-8573, Japan
81-3-3440-3569

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T. Hong Kong
852-26668334

Home Page:
www.freescale.com

Learn More: For more information about Freescale Semiconductor products, please visit www.freescale.com

AN2913
Rev. 1
07/2005

freescale™
semiconductor