

**Shri Govindram Seksaria Institute of Technology and
Science, Indore**

CO44401: SYSTEM OPERATIONS LAB



REPORT

JULY'18 - DEC'18

SUBMITTED TO:
Mr. Rajesh Dhakad
Miss. Shweta Dubey

SUBMITTED BY:
Rachee Kanodia (CS58)

Contents

- 1 Installation with creation of file system**
- 2 Root password change**
- 3 Directory creation in which all can write but only owner can delete files**
- 4 WAP to modify a file. Only owner can have permission to modify the file**
- 5 Apply quota in /home partition**
- 6 Modify/change password through script**
- 7 Configure DHCP for static IP assignment based on client MAC address**
- 8 Configure remote booting using pxeboot, DHCP and TFTP tool**
- 9 Write a cron job to remote power up a system in a LAN**
- 10 Write a cron job to shutdown a linux system**
- 11 Proxy DNS with split DNS and DNS cache**
- 12 Apache web server for virtual hosting for multiple domain names**

ASSIGNMENT

1. Installation with creation of file system using fdisk (manual) with at least one logical partition.

The fdisk command is a text-based utility for viewing and managing hard disk partitions on Linux

List Partitions

```
rachee@rachee-HP-Notebook: ~  
Disk /dev/loop4: 87.9 MiB, 92114944 bytes, 179912 sectors  
Units: sectors of 1 * 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
  
Disk /dev/loop5: 87.9 MiB, 92164096 bytes, 180008 sectors  
Units: sectors of 1 * 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 512 bytes  
I/O size (minimum/optimal): 512 bytes / 512 bytes  
  
Disk /dev/sda: 931.5 GiB, 1000204886016 bytes, 1953525168 sectors  
Units: sectors of 1 * 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 4096 bytes  
I/O size (minimum/optimal): 4096 bytes / 4096 bytes  
Disklabel type: dos  
Disk identifier: 0xfee6ed8e  
  
Device      Boot      Start          End      Sectors      Size Id Type  
/dev/sda1   *           2048 1936971775 1936969728 923.6G 83 Linux  
/dev/sda2           1936973822 1953523711 16549890    7.9G  5 Extended  
/dev/sda5           1936973824 1953523711 16549888    7.9G 82 Linux swap / Solaris
```

The **sudo fdisk -l** commands lists the partitions on system.

Entering Command Mode

```
rachee@rachee-HP-Notebook: ~  
Disk identifier: 0xfee6ed8e  
  
Device      Boot      Start          End      Sectors      Size Id Type  
/dev/sda1   *           2048 1936971775 1936969728 923.6G 83 Linux  
/dev/sda2           1936973822 1953523711 16549890    7.9G  5 Extended  
/dev/sda5           1936973824 1953523711 16549888    7.9G 82 Linux swap / Solaris  
  
Partition 2 does not start on physical sector boundary.  
  
rachee@rachee-HP-Notebook:~$ sudo fdisk /dev/sda1  
  
Welcome to fdisk (util-linux 2.27.1).  
Changes will remain in memory only, until you decide to write them.  
Be careful before using the write command.  
  
/dev/sda1: device contains a valid 'ext4' signature; it is strongly recommended  
to wipe the device with wipefs(8) if this is unexpected, in order to avoid possi  
ble collisions  
  
Device does not contain a recognized partition table.  
Created a new DOS disklabel with disk identifier 0x1b867f3f.  
  
Command (m for help):
```

To work on a disk's partitions, enter into command mode using command **sudo fdisk /dev/sda1**

Using Command Mode

```

Command (m for help): m
Help:

DOS (MBR)
a  toggle a bootable flag
b  edit nested BSD disklabel
c  toggle the dos compatibility flag

Generic
d  delete a partition
F  list free unpartitioned space
l  list known partition types
n  add a new partition
p  print the partition table
t  change a partition type
v  verify the partition table
i  print information about a partition

Misc
m  print this menu
u  change display/entry units
x  extra functionality (experts only)

```

Viewing the Partition Table

```

Command (m for help): p

Disk /dev/sda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0006c031

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *        2048     39845887    19921920   83   Linux
/dev/sda2                39847934    41940991     1046529    5   Extended
/dev/sda5                39847936    41940991     1046528   82   Linux swap
/ Solaris

Command (m for help): 

```

Deleting a Partition

```

Command (m for help): d
Partition number (1-5): 5

Command (m for help): p

Disk /dev/sda: 21.5 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0006c031

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *        2048     39845887    19921920   83   Linux
/dev/sda2             39847934    41940991     1046529    5   Extended

Command (m for help): █

```

Creating a Partition

```

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1  *        2048     39845887    19921920   83   Linux
/dev/sda2             39847934    41940991     1046529    5   Extended

Command (m for help): n
Command action
  l   logical (5 or over)
  p   primary partition (1-4)
l
First sector (39849982-41940991, default 39849982):
Using default value 39849982
Last sector, +sectors or +size{K,M,G} (39849982-41940991, default 41940991):
Using default value 41940991

Command (m for help): █

```

Use **w** to write the changes you've made to disk.

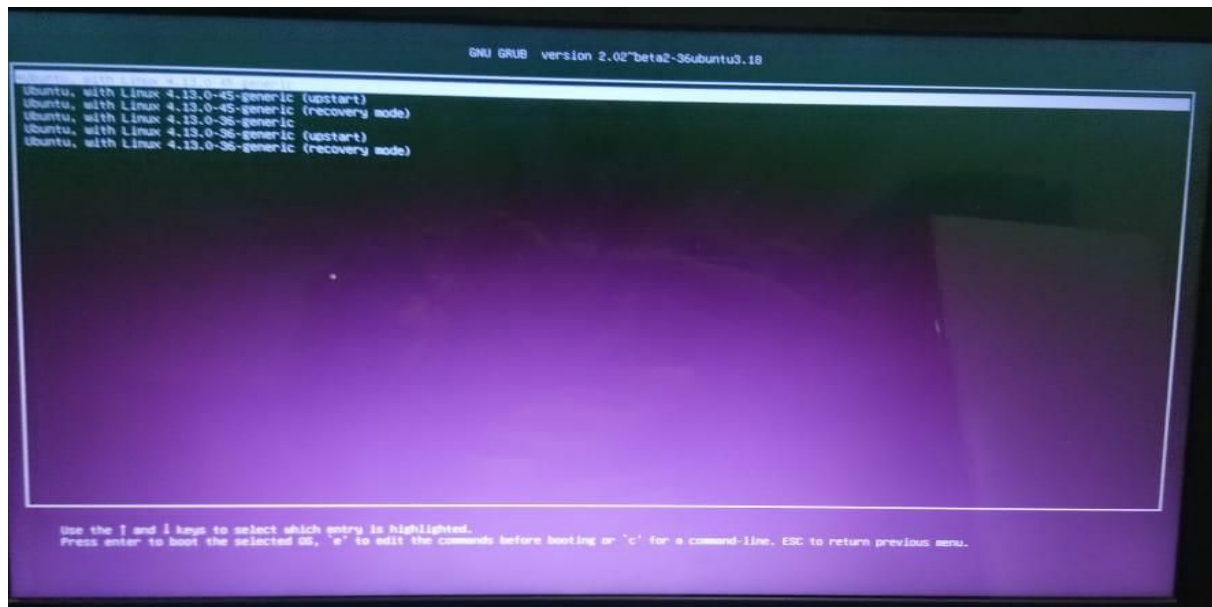
Use **q** if you want to quit without saving changes.

Formatting a Partition

Format the newly created partition using command **sudo mkfs.ext4 /dev/sda1**

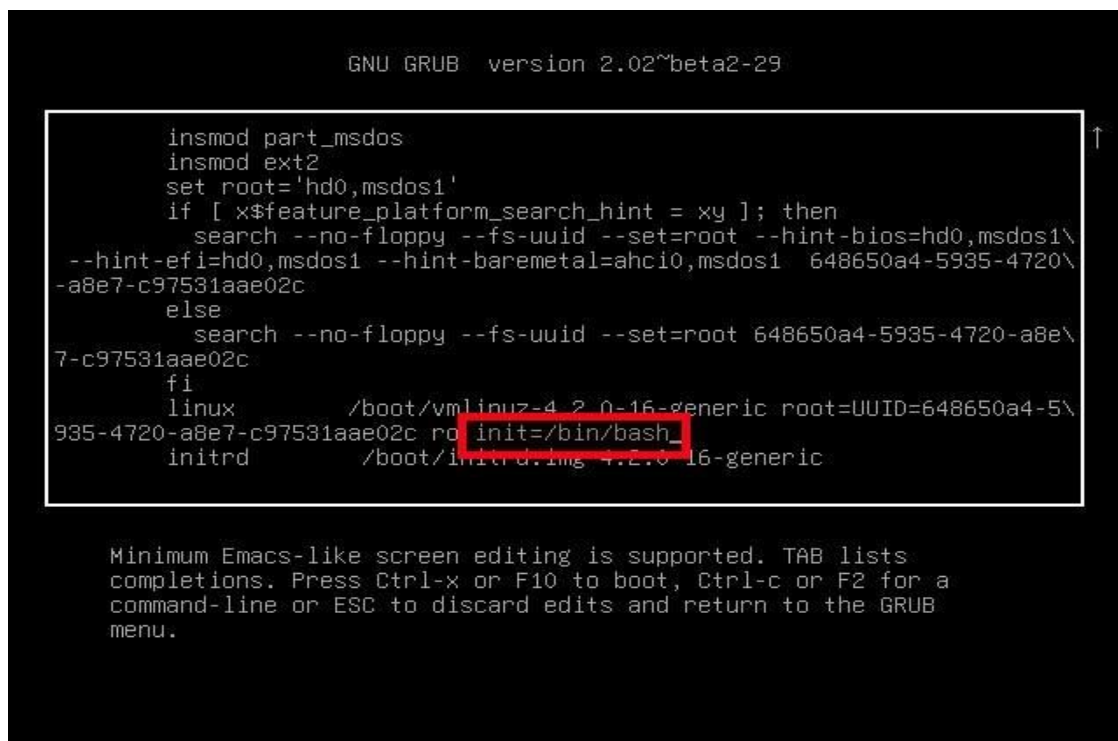
2. Root password change through boot loader options.

At the GRUB boot menu, select the boot entry and press e to edit the selected item.



Find the line that starts with word **linux**. Add the following line at the end.

`init=/bin/bash`



Then, press **CTRL-X** or **F10** to boot in to single user mode.

Type the following command to mount root (/) file system in to read/write mode.

mount -o remount,rw /

Then, change the password of your administrative account using command:

passwd sk

```
Begin: Loading essential drivers ... done.
Begin: Running /scripts/init-premount ... done.
Begin: Mounting root file system ... Begin: Running /scripts/local-top ... done.
Begin: Running /scripts/local-premount ... done.
Begin: Checking root file system ... fsck from util-linux 2.26.2
/dev/sda1: recovering journal
/dev/sda1: clean, 65655/1245184 files, 438442/4980480 blocks
done.
[ 1166.172583] EXT4-fs (sda1): mounted filesystem with ordered data mode. Opts:
(null)
done.
Begin: Running /scripts/local-bottom ... done.
Begin: Running /scripts/init-bottom ... done.
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
[ 1167.595787] random: nonblocking pool is initialized
root@(none):/#
root@(none):/#
root@(none):/# mount -o remount,rw /
[ 1185.378222] EXT4-fs (sda1): re-mounted. Opts: errors=remount-ro
root@(none):/# passwd sk
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@(none):/#
```

3. Directory creation in which all can write but only owner can delete files.

Create a directory using **mkdir** command

Assign it with the following mode and set its sticky bit

chmod 1777 Directory_name

Create a file inside directory using **touch** command. It will have the property where all can write but only owner can delete the file

4. WAP to modify a file. Only owner can have permission to modify the file. Change the permission of your compiled program in such a way so everyone can execute it and are able to modify your owned file.

Owner Script(src.sh file):

read file_name

read content

echo \$content>\$file_name

By other user:

Make a folder with its sticky bit set by setting its mode to 1777

Execute the code of src.sh

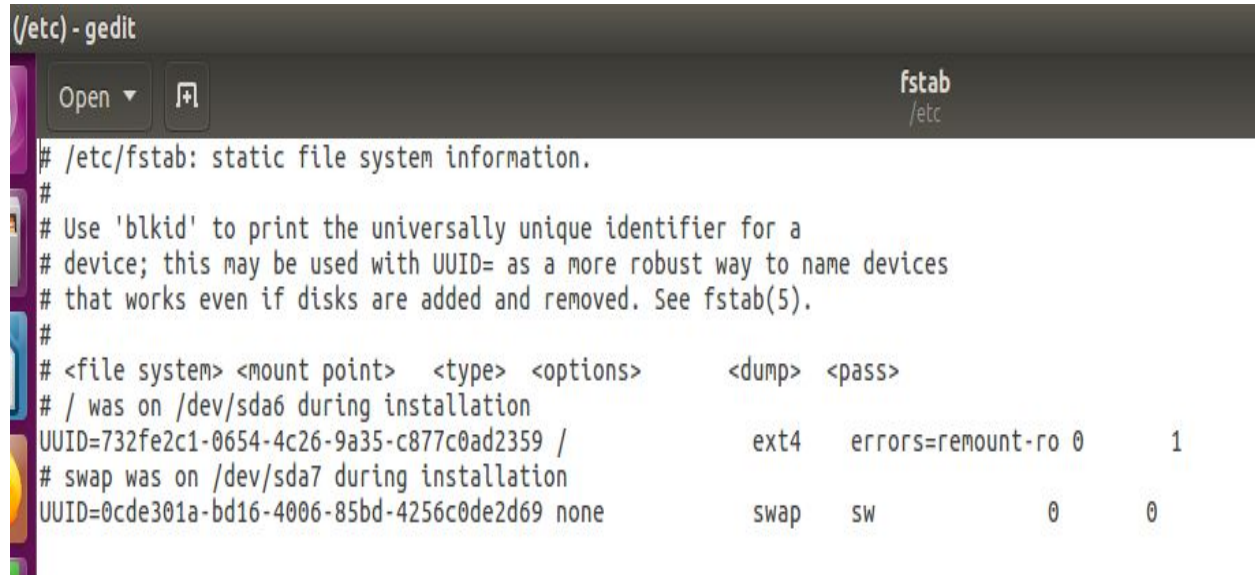
Try to create a file in the folder which can be only modified by the owner.

5. Apply quota in /home partition.

A **disk quota** is a limit set by a system administrator that restricts certain aspects of file system usage on modern operating systems. The function of setting quotas to disks is to allocate limited disk-space in a reasonable way.

Install quota-tools package.

Edit etc/fstab/ file



```
(etc) - gedit
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options>      <dump> <pass>
# / was on /dev/sda6 during installation
UUID=732fe2c1-0654-4c26-9a35-c877c0ad2359 /          ext4      errors=remount-ro 0      1
# swap was on /dev/sda7 during installation
UUID=0cde301a-bd16-4006-85bd-4256c0de2d69 none        swap      sw        0      0
```

Edit file to enable the quota mount option(s) on selected file systems

```
/dev/sda6 ext4 defaults,usrquota,grpquota 0 2
```

Remount the partition to apply the change:

```
# mount -vo remount /dev/sda6
```

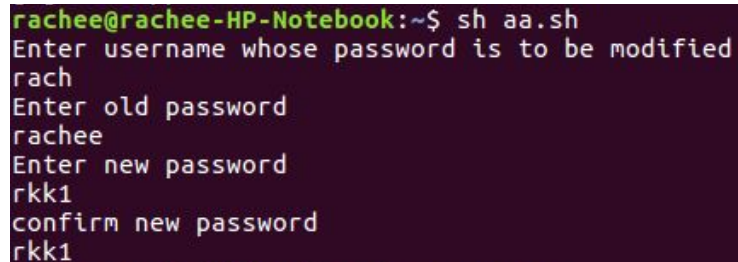
6. Modify/change password through script

Make a new script file as pass.sh

```
echo "Enter username whose password is to be modified"
read username
echo "Enter old password"
read oldpass
echo "Enter new password"
read newpass
echo "confirm new password"
read confirmpass
if [ newpass -eq confirmpass ]
then
passwd <<EOF
$username
$oldpass
$newpass
```



```
EOF
echo "password changed"
else
echo "password do not match"
fi
```

A terminal window with a dark purple background. The prompt is 'rachee@rachee-HP-Notebook:~\$'. The user runs 'sh aa.sh'. The script prompts for a username, 'rachee', an old password, 'rachee', a new password, 'rkk1', and a confirmation of the new password, 'rkk1'.

```
rachee@rachee-HP-Notebook:~$ sh aa.sh
Enter username whose password is to be modified
rachee
Enter old password
rachee
Enter new password
rkk1
confirm new password
rkk1
```

7. Configure DHCP for static IP assignment based on client MAC address.

Open the “/etc/network/interfaces” file.

```
sudo nano /etc/network/interfaces
Add/edit as described below
auto eth0
iface eth0 inet static
address 192.168.1.20
Network 192.168.1.0
netmask 255.255.255.0
Broadcast 192.168.255.255
gateway 192.168.1.1
```

Save the file and exit. This will set its IP address to “192.168.1.20”. Reboot

8. Configure remote booting using pxeboot, DHCP and TFTP tool.

PXE (Preboot Execution Environment) Server allows the user to boot a Linux distribution from a network and install it on hundreds of PCs at a time without any Linux iso images.

1. First setup PXE server to use a static IP. To set up a static IP address in your system, you need to edit the “/etc/network/interfaces” file.

Save the file and exit. This will set its IP address to “192.168.1.20”. Restart the network service.

2. DHCP, TFTP and NFS are essential components for configuring a PXE server. Install these components.

```
sudo apt-get update
```

```
sudo apt-get install isc-dhcp-Server inetutils-inetd tftpd-hpa syslinux nfs-kernel-Server
```

3. Configure DHCP server

1. Edit the “/etc/default/dhcp3-server” file.
`sudo nano /etc/default/dhcp3-server`

Add/edit as described below:
`INTERFACES="eth0"`
Save and exit the file.

2. Edit the “/etc/dhcp3/dhcpd.conf” file:
`sudo nano /etc/dhcp/dhcpd.conf`
Add/edit as described below:
`default-lease-time 600;
max-lease-time 7200;
subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.21 192.168.1.240;
option subnet-mask 255.255.255.0;
option routers 192.168.1.20;
option broadcast-address 192.168.1.255;
filename "pxelinux.0";
next-Server 192.168.1.20;
}`

Save the file and exit.

3. Start the DHCP service.

`sudo /etc/init.d/isc-dhcp-server start`

4. Configure TFTP server

The TFTP server is always listening for PXE clients on the network. When it detects any network PXE client asking for PXE services, then it provides a network package that contains the boot menu.

1. To configure TFTP, edit the “/etc/inetd.conf” file.

`sudo nano /etc/inetd.conf`

Save and exit the file.

3. Enable boot service for inetd to automatically start after every system reboot and start tftpd service.

`sudo update-inetd --enable BOOT`

`sudo service tftpd-hpa start`

4. Check status.

`sudo netstat -lu`

5. Configure PXE boot files

Now you need the PXE boot file “pxelinux.0” to be present in the TFTP root directory. Make a directory structure for TFTP, and copy all the bootloader files provided by syslinux from the “/usr/lib/syslinux/” to the “/var/lib/tftpboot/” path

Set up PXELINUX configuration file

The PXE configuration file defines the boot menu displayed to the PXE client when it boots up and contacts the TFTP server. By default, when a PXE client boots up, it will use its own MAC address to specify which configuration file to read, so we need to create that default file that contains the list of kernels which are available to boot.

Edit the PXE Server configuration file with valid installation options.

To edit “/var/lib/tftpboot/pxelinux.cfg/default,”

```
sudo nano /var/lib/tftpboot/pxelinux.cfg/default
```

Edit the “/var/lib/tftpboot/pxelinux.cfg/pxe.conf” file.

```
sudo nano /var/lib/tftpboot/pxelinux.cfg/pxe.conf
```

Add Ubuntu 14.04 Desktop Boot Images to PXE Server

For this, Ubuntu kernel and initrd files are required. To get those files, you need the Ubuntu 14.04 Desktop ISO Image. You can download the Ubuntu 14.04 ISO image in the /mnt folder by issuing the following command:

```
sudo cd /mnt
```

```
sudo wget http://releases.ubuntu.com/14.04/ubuntu-14.04.3-desktop-amd64.iso
```

Mount the ISO file, and copy all the files to the TFTP folder

6. Configure NFS Server to Export ISO Contents

To configure the NFS server, edit the “/etc/exports” file.

Run commands to reflect changes

```
sudo exportfs -a
```

```
sudo /etc/init.d/nfs-kernel-server start
```

7. Configure Network Boot PXE Client

A PXE client can be any computer system with a PXE network boot enable option. Now your clients can boot and install Ubuntu 14.04 Desktop by enabling “Boot From Network” options from their systems BIOS.

9. Write a cron job to remote power up a system in a LAN.

The software utility cron is a time-based job scheduler in Unix-like computer operating systems. People who set up and maintain software environments use cron to schedule jobs to run periodically at fixed times, dates, or intervals.

Each user has their own "crontabs" which control what jobs are executed and when. The general format of a crontab is:

```
* * * * * command to be executed
- - - - -
| | | | |
| | | | +----- day of week (0 - 6) (Sunday=0)
| | | +----- month (1 - 12)
| | +----- day of month (1 - 31)
| +----- hour (0 - 23)
+----- min (0 - 59)
```

So, for example, this will run ls every day at 14:04:

```
04 14 * * * ls
```

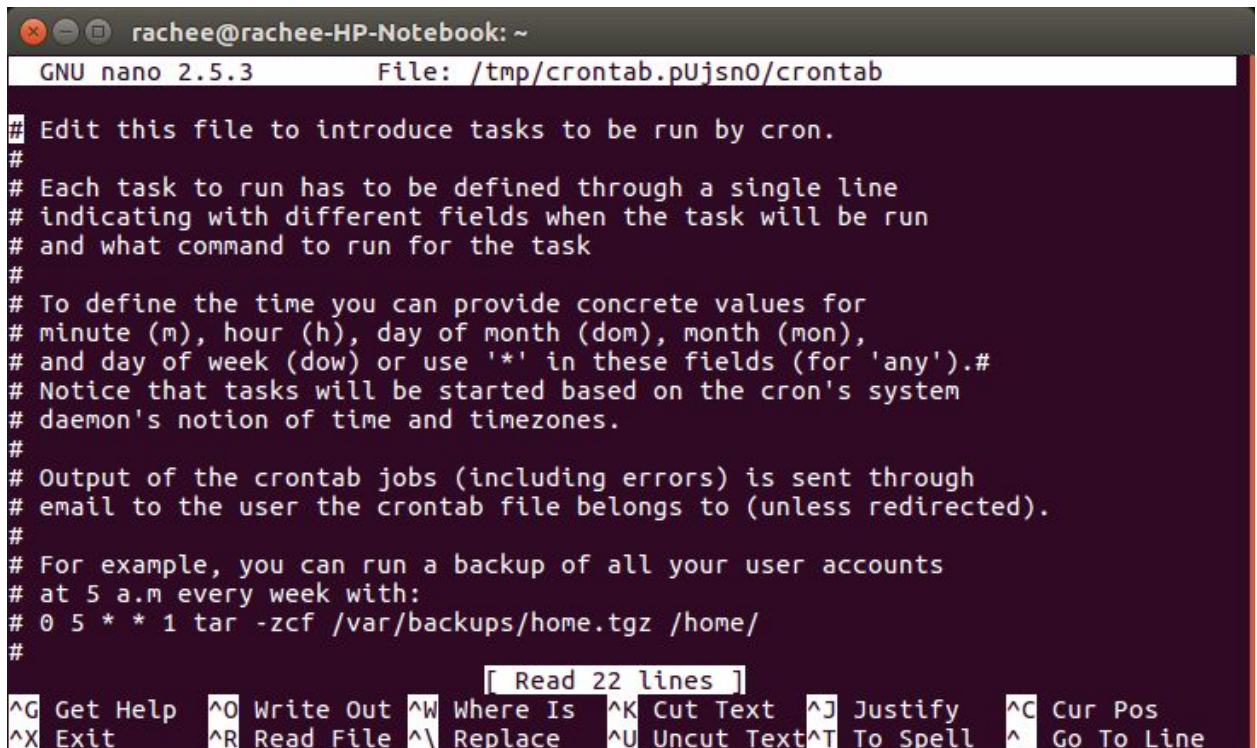
To set up a cronjob for remote shutdown:

1. Create a new crontab by running `crontab -e`. This will bring up a window of your favorite text editor.

```
rachee@rachee-HP-Notebook:~$ crontab -e
no crontab for rachee - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/ed
 2. /bin/nano <---- easiest
 3. /usr/bin/vim.tiny

Choose 1-3 [2]: 1
```



```
GNU nano 2.5.3      File: /tmp/crontab.pUjsn0/crontab

## Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#

[ Read 22 lines ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

2. Add this line to the file that just opened.
3. 34 14 15 5 wakeonlan 23:34:e3:ac:12:a3
4. Save the file and exit the editor.

10. Write a cron job to shutdown a linux system.

All the steps are same as above

Step 3 is modified as

```
34 14 15 5 shutdown
```

11. Proxy DNS with split DNS and DNS cache.

Split DNS is simply a configuration in which the IP address to which a DNS name resolves is dependent on the location of the client. It is most often used in a NAT environment to insure that local clients resolve the DNS names of local servers to their RFC 1918 addresses while external clients resolve the same server names to their public counterparts.

Setting up Split DNS

Setting up Split DNS is extremely simple:

1. Be sure that your firewall/router can resolve external DNS names.
2. Install the **dnsmasq** package
3. #

```
# hosts      This file describes a number of hostname-to-address
#            mappings for the TCP/IP subsystem.  It is mostly
#            used at boot time, when no name servers are running.
#            On small systems, this file can be used instead of a
#            "named" name server.
# Syntax:
#
# IP-Address Full-Qualified-Hostname Short-Hostname
#
127.0.0.1    localhost
172.20.0.1   openvpn.shorewall.net  openvpn
172.20.0.2   vpn02.shorewall.net    vpn02
2002:ce7c:92b4::1 gateway6.shorewall.net gateway6
2002:ce7c:92b4:1::2 mail6.shorewall.net  mail6
2002:ce7c:92b4:1::2 lists6.shorewall.net lists6
2002:ce7c:92b4:2::2 server6.shorewall.net server6
```

4. If your local hosts are configured using DHCP, that is a simple one-line change to the DHCP configuration.

local clients will resolve those names in the firewall/router's /etc/hosts file as defined in that file. All other names will be resolved using the firewall/router's Name Server as defined in /etc/resolv.conf.

Example:

From an Internet Host:

```
gateway:~ # host linksys.shorewall.net
```

linksys.shorewall.net has address 206.124.146.180

```
gateway:~ #
```

From Tipper (192.168.1.132):

```
teastep@tipper:~$ host linksys
linksys.shorewall.net has address 172.20.1.1
teastep@tipper:~$
```

Apache web server for virtual hosting for multiple domain names.

all of the configuration files for Apache are located in /etc/httpd/conf and /etc/httpd/conf.d. The data for the websites is located in /var/www by default. With multiple websites, you will need to provide multiple locations, one for each site you host.

12. Apache web server for virtual hosting for multiple domain names.

Name-based virtual hosting

With name-based virtual hosting, you can use a single IP address for multiple websites. Modern web servers, including Apache, use the hostname portion of the specified URL to determine which virtual web host responds to the page request. This requires only a little more configuration than for a single site.

Even if you are starting with only a single website, I recommend that you set it up as a virtual host, which will make it easier to add more sites later. In this article, I'll pick up where we left off in the previous article, so you'll need to set up the original website, a name-based virtual website.

Preparing the original website

add the following stanza to the bottom of /etc/httpd/conf/httpd.conf configuration file

```
<VirtualHost 127.0.0.1:80>
    DocumentRoot /var/www/html
    ServerName www.site1.org
</VirtualHost>
```

This will be the first virtual host stanza, and it should remain first, to make it the default definition. That means that HTTP access to the server by IP address, or by another name that resolves to this IP address but that does not have a specific named host configuration stanza, will be directed to this virtual host. All other virtual host configuration stanzas should follow this one.

Restart the HTTPD server to enable the changes to the httpd configuration. You can then look at the website using the Lynx text mode browser from the command line.

```
[root@testvm1 ~]# systemctl restart httpd
[root@testvm1 ~]# lynx www.site1.org
```

Configuring the second website

Create a new website directory structure with the following command:


```
[root@testvm1 html]# mkdir -p /var/www/html2
```

Notice that the second website is simply a second html directory in the same /var/www directory as the first site.

Create a new configuration stanza in httpd.conf for the second website and place it below the previous virtual host stanza (the two should look very similar). This stanza tells the web server where to find the HTML files for the second site.

```
<VirtualHost 127.0.0.1:80>
    DocumentRoot /var/www/html2
    ServerName www.site2.org
</VirtualHost>
```

Restart HTTPD again and use Lynx to view the results.

```
[root@testvm1 httpd]# systemctl restart httpd
```

```
[root@testvm1 httpd]# lynx www.site2.org
```