Creating Backups

dd - convert and copy a file

```
$ dd if=/dev/sdb1 of=/root/sdb1.img
   if: input file
   of: output file
```

Make an ISO copy of the CDROM

```
$ dd if=/dev/sr0 of=/root/vmtools.iso
```

Creating Local User Groups

```
$ groupadd group-one
# Same as
$ addgroup group-two

$ cat /etc/group | grep 'group-one\|group-two'
   group-one:x:1002:
   group-two:x:1002:
```

To add user **linuxuser** to `group-one` and `group-two`

```
$ vim /etc/group
   group-one:x:1002:linuxuser
   group-two:x:1002:linuxuser
```

If we want to be able to change into a group so all subsequent files/directories created are under said group, we need to set a password for said group

```
$ gpasswd group-one
   # enter password
$ gpasswd group-two
   # enter password

# change into the group
$ newgrp group-one
   # type in group-one's password
```

All files/directories created now will be created as * user: linuxuser * group: group-one

To change the group ownership of a file/directory to `group-two` without logging into `group-two`

```
$ chgrp group-two file_name.txt
```

Managing File Permissions

```
$ chmod a+rwx file_name.txt

   a: all
   u: user
   g: group
   o: other

   r: read: 4
   w: write: 2
   x: execute: 1
```

Managing FSTAB Entries

```
//192.168.1.60/news   /mnt/tmp    cifs    credentials=/mnt/.smbcredentials,defaults   0   0
```

Managing Local User Accounts

```
$ useradd -d /home/testuser testuser
   -d: designate home directory$
$ passwd testuser
   <type password>

$ adduser testuser
   # spits out more information than useradd and contains more functionality

$ userdel testuser
   # does not automatically delete the user's home directory
$ userdel -r testuser
   # -r: remove the user's home directory
```

Managing the Startup Process and Related Services

```
# Ubuntu 14.04
$ cd /etc/init.d
$ status ssh
$ start/stop/restart ssh
$ echo "manual" > /etc/init.d/sshd.override
   # this will disbale sshd on startup; to re-enable, just remove the override file

# CentOS 7.0
$ cd /etc/rc.d
$ systemctl status sshd
$ systemctl start/stop/restart sshd
$ systemctl enable/disable sshd
```

Managing User Accounts

Creating a linux user manually without using the `useradd` or `adduser` utilities

```
$ vipw
   # VI the password file to add a new user
testuser:x:1002:1002:,,,:/home/testuser:/bin/bash

$ vigr
   # VI the group file to add a new gruop
testuser:x:1002:

$ mkdir /home/testuser
$ cp -rf /etc/skel/* /home/testuser
$ chown -R testuser:testuser /home/testuser

$ passwd testuser
   # type password
```

Managing User Account Attributes

```
$ chfn
   # Changing the full name of the user
$ chsh /bin/dash
```

Prevent a user from logging in and provide a message to them

```
$ chsh /bin/false
   # can still 'su - testuser'
   # cannot SSH into testuser
$ chsh /sbin/nologin
   # cannot 'su - testuser
   # cannot SSH into testuser
```

Managing User Processes

```
$ top
   # NICE: 20 = lowest priority
   # NICE: -20 = highest priority
$ htop
   # much nicer view of system processes
$ ps
   # show processes running for current user
$ ps aux
   # show all running processes for all users
```

Find a process based on the name

```
$ pgrep bash
   # returns a PID of 1794
$ ps aux | grep 1794
   # verifies that bash has a PID of 1794
```

Kill the PID of cron

```
$ ps aux | grep cron
   # returns the PID of cron - 1292 that is owned by root
$ kill 1292
   # the kill command must be run by root
$ kill -KILL 1292
   # send a signal to the OS kernel to shutdown the process
$ kill -15 or -9
$ kill -HUP 1292
$ kill -l
   # shows all signals that can be sent to the application or the OS
```

To re-priotize the application that is running, we can `nice` or `renice` the PID of the program

```
$ renice 10 1292
   # lower the priority of PID 1292 to 10
$ nice -n 20 1310
   # make the default value of 1310 start at priority of 20
```

Restoring Backed Up Data

```
dd if=/root/sdb1.img of=/dev/sdc1
mount -o loop /root/vmtool.iso /mnt/tmp
```

Accessing the Root Account

```
$ su -
```

Using SUDO to Manage Access to the Root Account

```
$ visudo
   # change the sudoers file
$ sudo vim /etc/sudoers
   # do not use the above method because you can bork the file
```

Basic Bash Shell Scripting: Basic Setup and Execution

```
$ export PATH=$PATH:/root/scripts
   # add the PATH /root/scripts to my $PATH so we can execute the script from anywhere
```

Basic Bash Shell Scripting: Conditionals and Loops

```
#!/bin/bash

DIRECTORY="/root/scripts/test"

if [ -d "$DIRECTORY" ]; then
    echo "This directory exists"
else
    echo "This directory does not exist"
fi

for count in 1 2 3 4 5
do
    echo "This is line # $count"
done

while read HOST; do
    ping -c 3 $HOST
done < myhosts
# a file called 'myhosts' contains 2 lines:
#   8.8.8.8
#   8.8.4.4
```