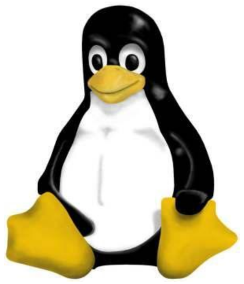
NowSecure™

# Linux Tools for Forensic Analysis

*Background, Technique and Analytic Tools*



## What You’ll Learn

As a forensic investigator, Linux knowledge is essential. Linux contains several native tools that are extremely powerful for data extraction and analysis.

This course will give you the foundation needed to be more effective as a forensic investigator.

## What You’ll Learn

This course is split into a number of areas:

- Installation and setup of a Linux OS
- Linux overview and filesystems
- Common commands

Throughout, you will complete hands - on exercises testing what you’ve learned.

Enjoy Linux 101 - Linux for Mobile Forensics.

## Installing and setting up the Santoku VM

Prior to starting this course, visit the following link to download Santoku Linux (base OS for the App Testing Community Edition):

https://www.nowsecure.com/apptesting/community/#download

To run Santoku, you must install virtual machine (VM) software. Download the appropriate version of VirtualBox for your host machine (Windows, OSX, or Linux) here:

https://www.virtualbox.org/wiki/Downloads

Note: If you have viaExtract 2.1.1 or later, you can use your viaExtract OS as well.

NowSecure™

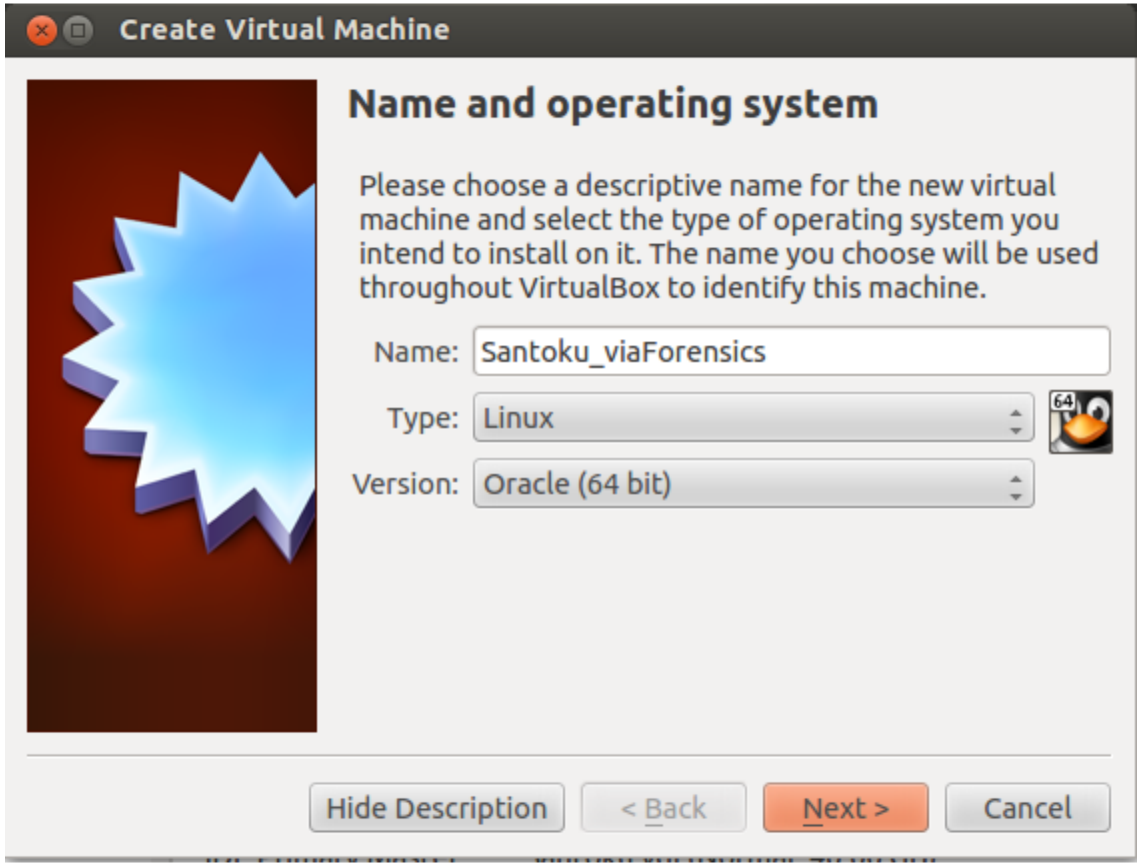
## Installing and setting up the Santoku VM

After VirtualBox has downloaded, install the virtual machine software, and follow the next steps to start the VM:

- Locate your VirtualBox installation.
- Select New to create a new VM.
- Go through the VM wizard.

# Installing and setting up the Santoku VM

Go through the wizard and create a name for your VM. Select the Linux/Debian Operating System and Version.



Name and operating system

Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

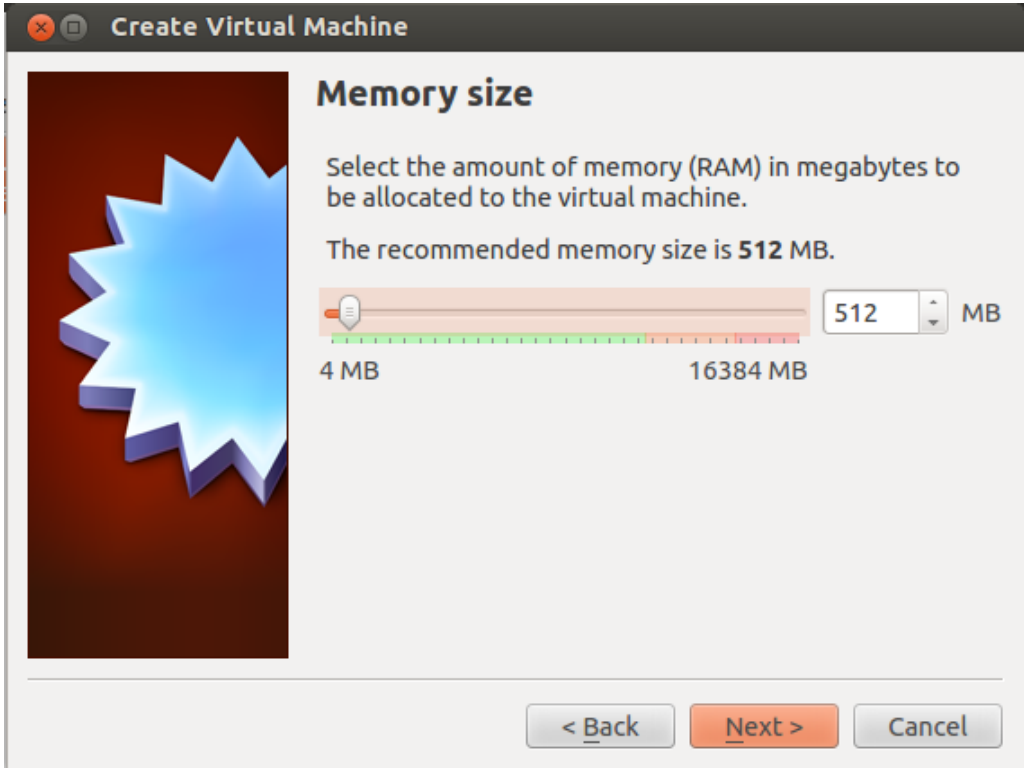
Name: Santoku_viaForensics

Type: Linux

Version: Oracle (64 bit)

## Installing and setting up the Santoku VM

Select an appropriate amount of memory for the VM. The base memory size is defaulted to 512MB.  Feel free to add more if your system is powerful enough to spare some extra processing power, and then click Next.


Create Virtual Machine

Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

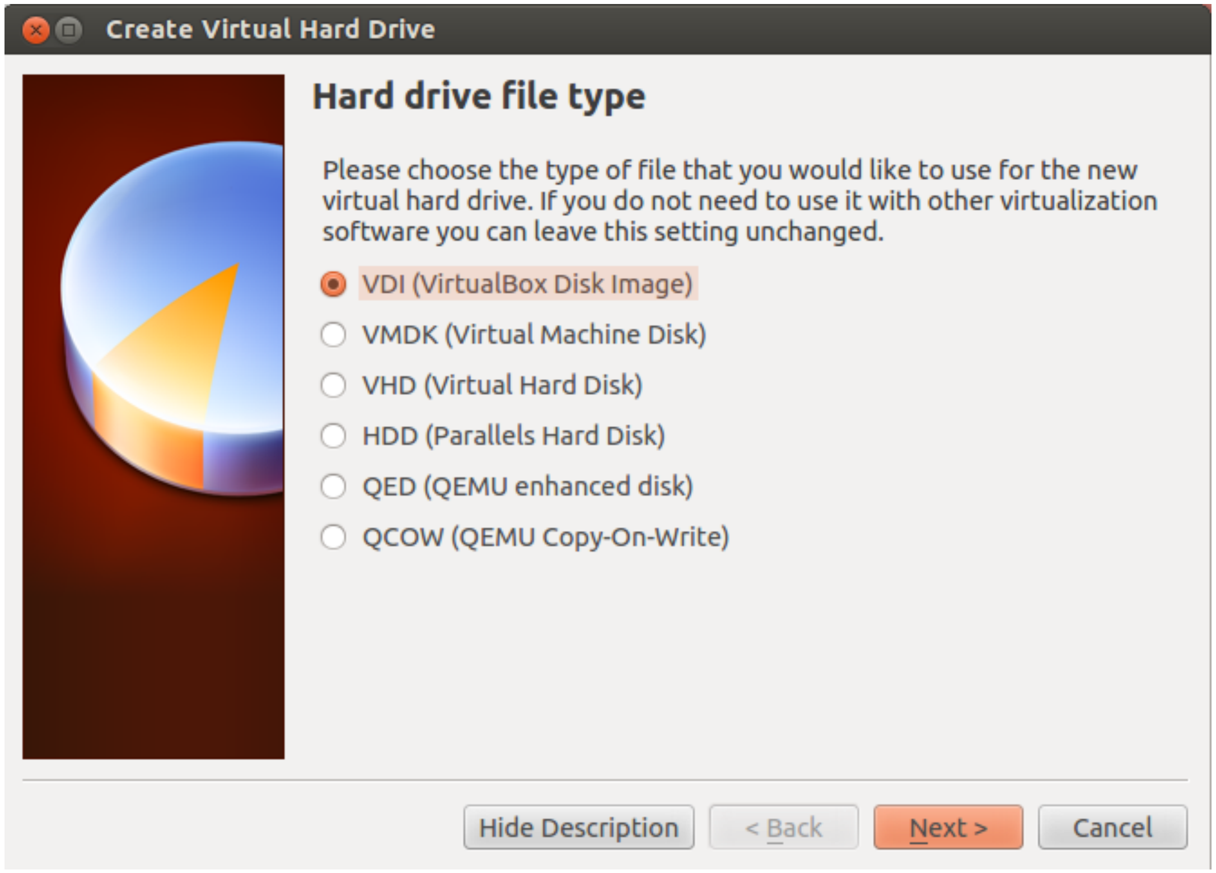
The recommended memory size is **512** MB.

512 MB

NowSecure™

# Installing and setting up the Santoku VM

On the "Hard Disk" screen, select Create new hard disk, and then click Create. Then, to create a new hard disk, select VDI (VirtualBox Disk Image) option. Click Next.



Create Virtual Hard Drive

Hard drive file type

Please choose the type of file that you would like to use for the new virtual hard drive. If you do not need to use it with other virtualization software you can leave this setting unchanged.

- [✓ VDI (VirtualBox Disk Image)]
- VMDK (Virtual Machine Disk)
- VHD (Virtual Hard Disk)
- HDD (Parallels Hard Disk)
- QED (QEMU enhanced disk)
- QCOW (QEMU Copy-On-Write)

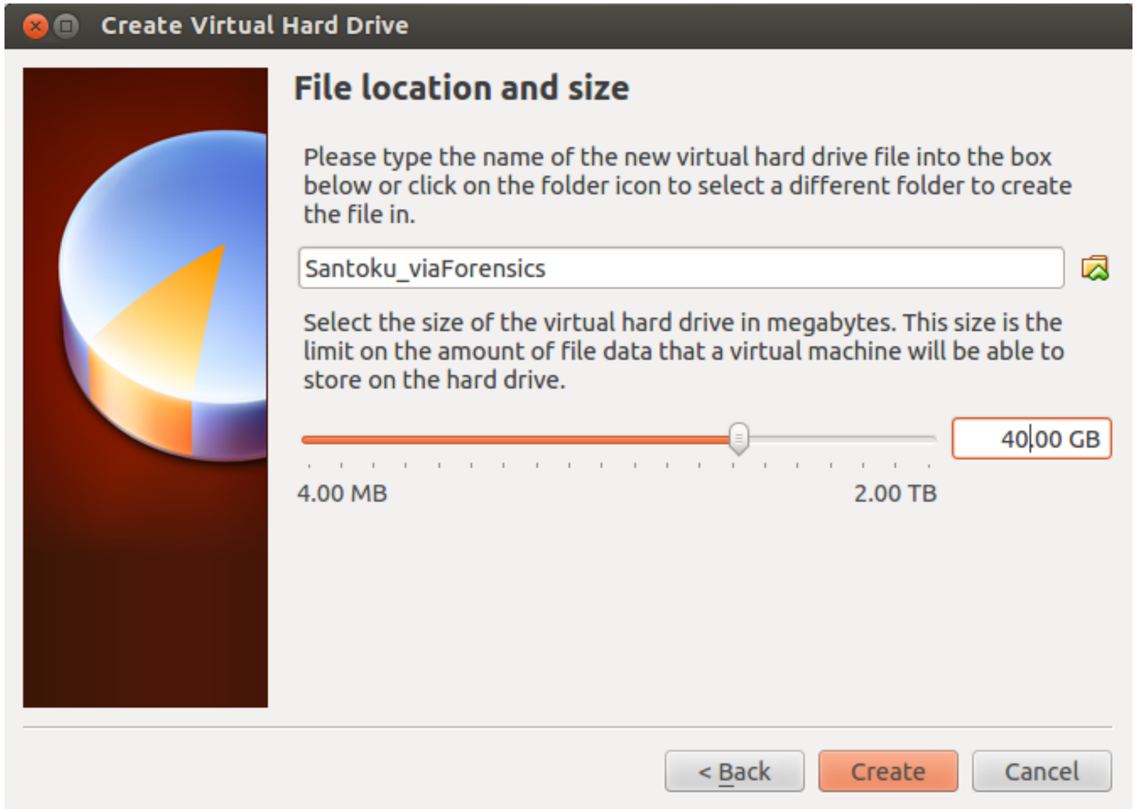
NowSecure™

## Installing and setting up the Santoku VM

On the next screen, select Dynamically Allocated. Selecting this option allows for the the virtual hard drive file to grow as it is used, up to a maximum fixed size.

# Installing and setting up the Santoku VM

Next, name your virtual hard drive, and adjust the size slider to allocate the amount of space of your choice to your Santoku hard drive. The default space in VirtualBox is 8GB. We recommend increasing this to 40GB. Click Create.



Create Virtual Hard Drive

File location and size

Please type the name of the new virtual hard drive file into the box below or click on the folder icon to select a different folder to create the file in.

Santoku_viaForensics

Select the size of the virtual hard drive in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard drive.

40,00 GB

4.00 MB — 2.00 TB

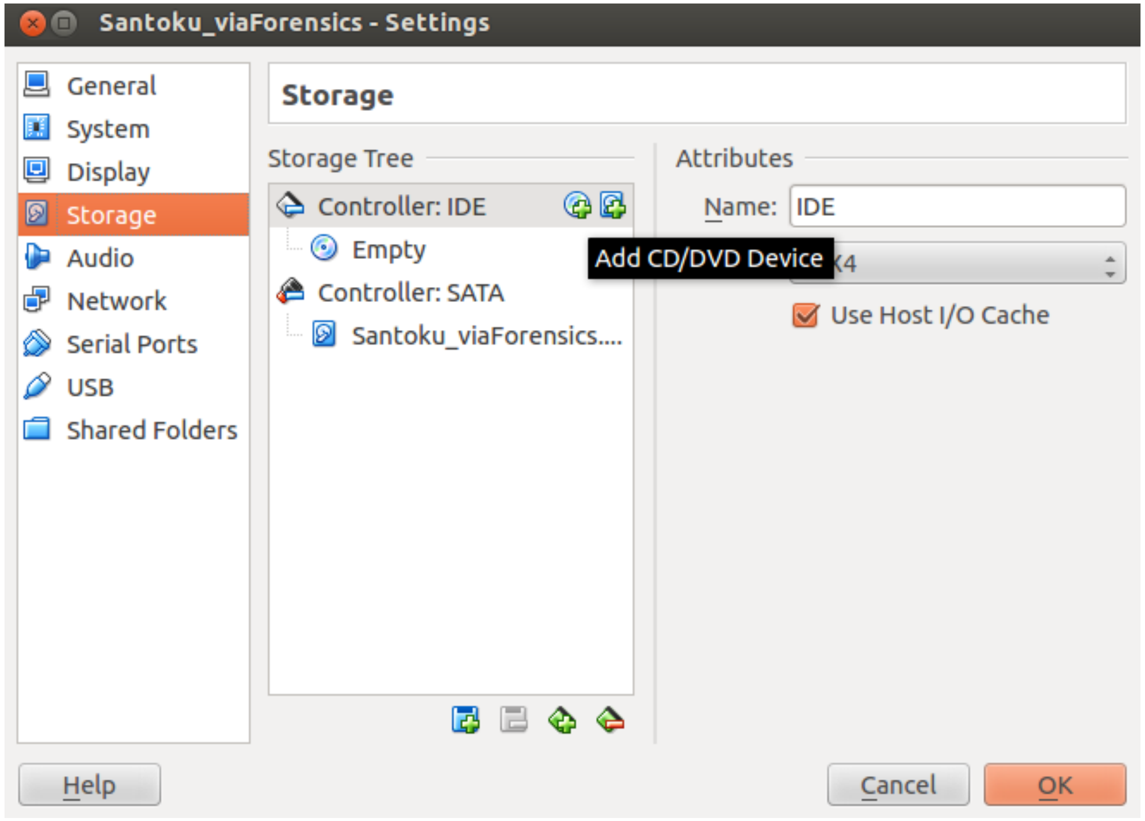
## Installing and setting up the Santoku VM

You’ll now be directed back to the main VirtualBox menu. Congratulations! You’ve created your Virtual Machine!

To get Santoku Linux to run on the VirtualBox, you need to have it attached to your newly created Virtual Machine. This is equivalent to inserting a Windows Install CD (for example) into a CD-ROM drive to boot for the first time you are installing the new OS.

# Installing and setting up the Santoku VM

To do this, select the Santoku Linux VM that you just created and click Settings at the top of the screen. Select Storage on the left of the Settings screen, and then click the cd icon next to the "IDE Controller."



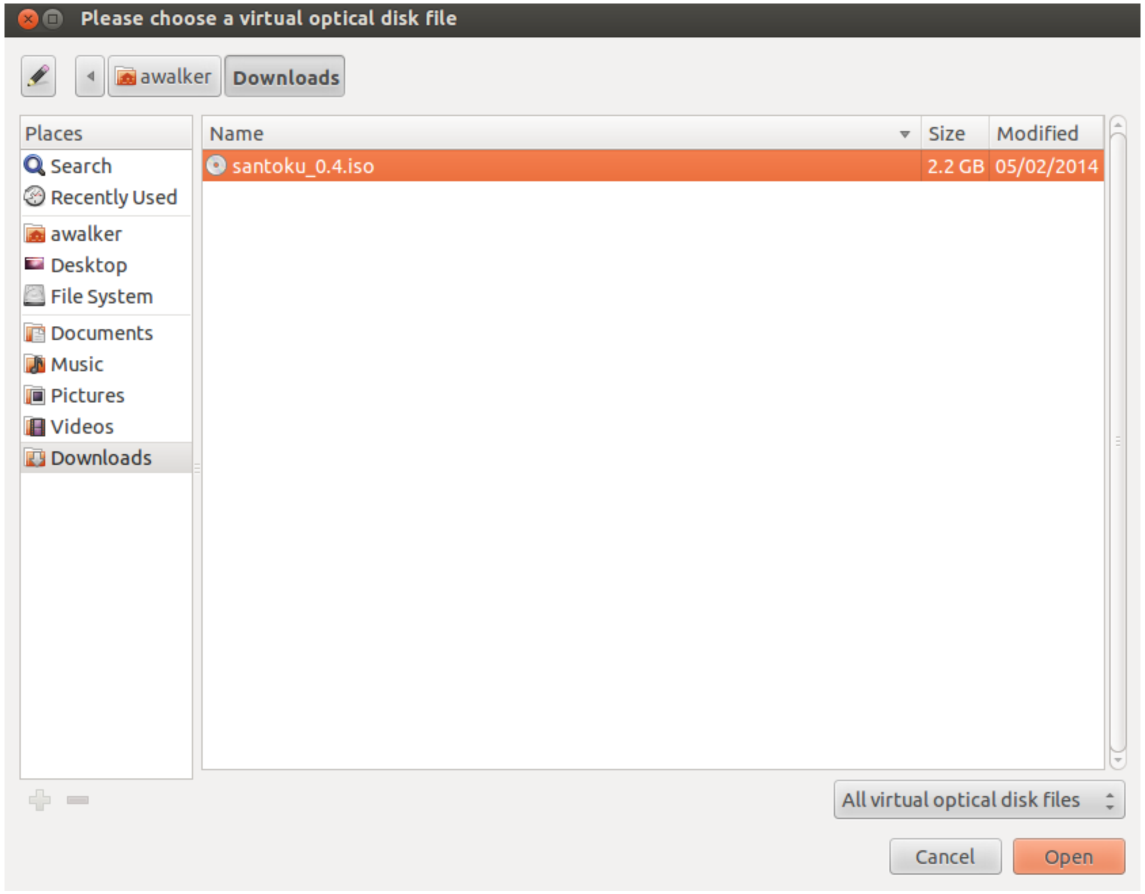
Name: IDE

Type: [partially hidden]X4

✓ Use Host I/O Cache

## Installing and setting up the Santoku VM

A warning will pop up asking you to choose a virtual DVD. Select Choose Disk and navigate to your recently downloaded Santoku.iso file (in this case, it's in the home user's /Downloads/Santoku file). Click Open and then OK.



| Name | Size | Modified |
|---|---|---|
| santoku_0.4.iso | 2.2 GB | 05/02/2014 |

NowSecure™

# Installing and setting up the Santoku VM

Installing Santoku:

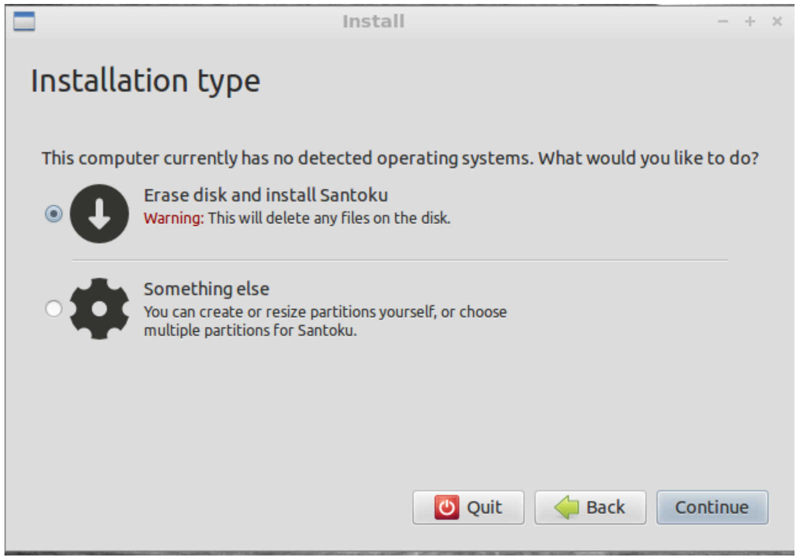
You can now click Start on the main VirtualBox screen to load the VM. Select to either boot from the live DVD or install. If you have created this in a VM, choose install-start the installer directly.

# Installing and setting up the Santoku VM

Choose your language, time zone and clock settings, and then select Erase disk and install Santoku on the "Installation type" screen.

Note: If you are choosing this option and you are not installing Santoku in a VM (you are installing it directly to your hard drive), choosing this will ERASE YOUR HARD DRIVE.

Click Continue.



Installation type

This computer currently has no detected operating systems. What would you like to do?

[✓ Erase disk and install Santoku]
Warning: This will delete any files on the disk.

Something else
You can create or resize partitions yourself, or choose multiple partitions for Santoku.

## Installing and setting up the Santoku VM

Add your username and password, and click Install.

After the Installation is complete, reboot when prompted, and then log in using the username and password you created during the installation process.

## Installing and setting up the Santoku VM

VMware uses a similar wizard interface for configuring a new virtual machine. If you are using VMWare, simply follow the prompts for setting up an existing virtual machine.

## Installing and setting up the Santoku VM

Congratulations! You've successfully set up your Santoku virtual machine!

NowSecure™

## Installing Training Tools

Let’s install some additions to your Santoku Linux that you’ll use later on during your training:

- Linux Training SDCard (this will be used later in the training): https://downloads.nowsecure.com/direct/linux-training-sdcard-image.dd

- Tools Folder: https://downloads.nowsecure.com/direct/opt-via-folder.zip

- Download the opt-via-folder.zip to your Santoku desktop. Unpack the zip to the desktop then open a terminal and type:

```
sudo mkdir /opt/
sudo mv -r ~/Desktop/via /opt/via
```

## Enable File Sharing

You need to enable file sharing between the guest and the host because the VM hard drive is set at 40GB, and it will fill up quickly if you use it to image drives and other devices.

In addition, you will need various files for this training that are more easily-accessed from your host machine.

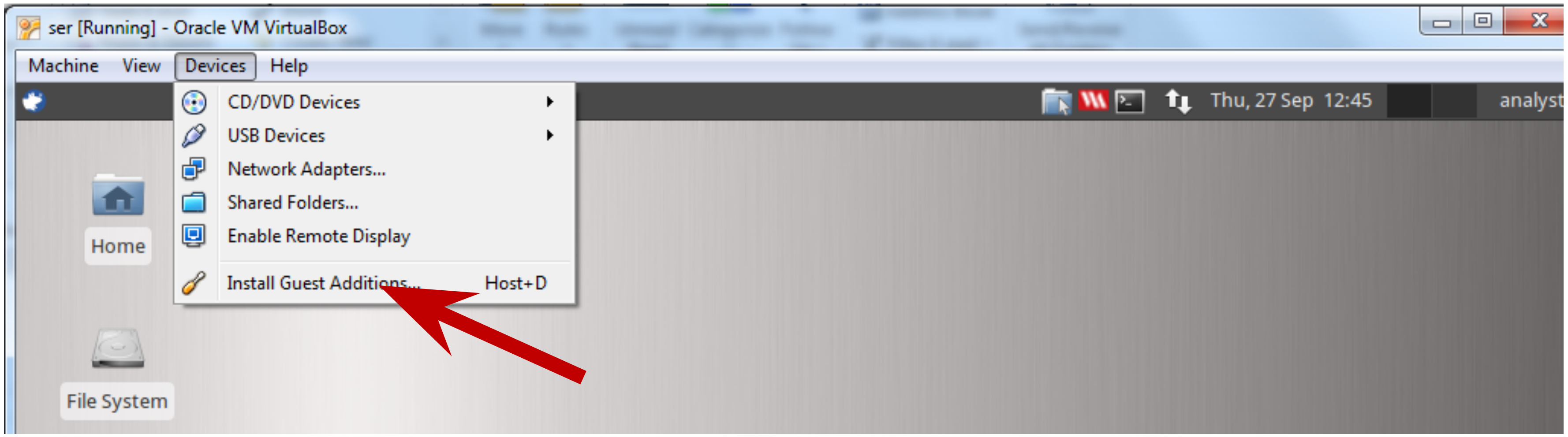
## Enable File Sharing

If you are using VirtualBox follow the instructions in this section.

If you are using VMware follow the instructions in the next section.

## Enable File Sharing - VirtualBox

From the VM Devices menu, click Install Guest Additions:



# Enable File Sharing - VirtualBox

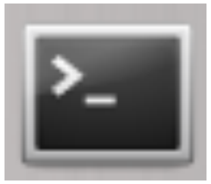
You should now see a VBOXADDITIONS icon on your Ubuntu desktop.

1. Right click on the icon and select mount volume
2. Open a terminal window and navigate to the /media directory by typing the following, then hit “tab.”

   ```
   cd /media/VBOXADDITIONS
   ```

TAB will be your friend throughout this training so use the tab key to autocomplete the command!

NOTE: To open a terminal window, hover your mouse over the lower portion of the screen and wait for the shortcuts to appear. Click on the black terminal shortcut:



NowSecure™

## Enable File Sharing - VirtualBox

3. Next type:

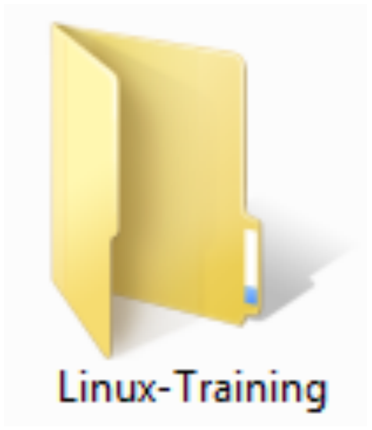
```
sudo sh ./VBoxLinuxAdditions.run
```

4. If prompted for a password, type your vm login password
5. When finished, right click on the VBOXADDITIONS icon on your desktop and eject volume.

NowSecure™

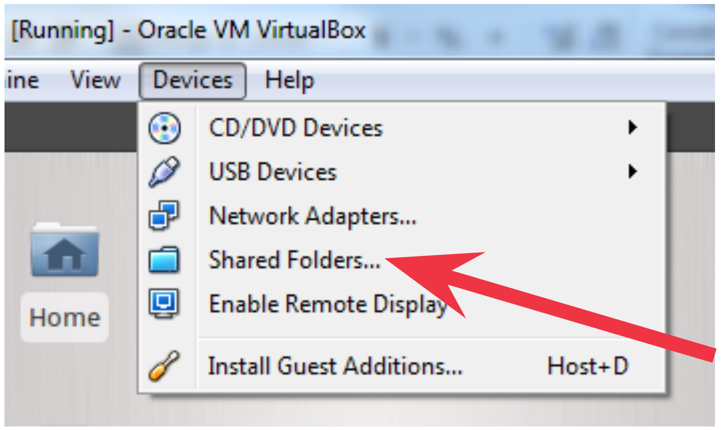
## Enable File Sharing - VirtualBox

If you haven't done so already, you should create a folder on your host workstation for all files needed in this course.  If you are using Windows on your host machine just create a directory under "Documents" called Linux-Training.



## Enable File Sharing - VirtualBox

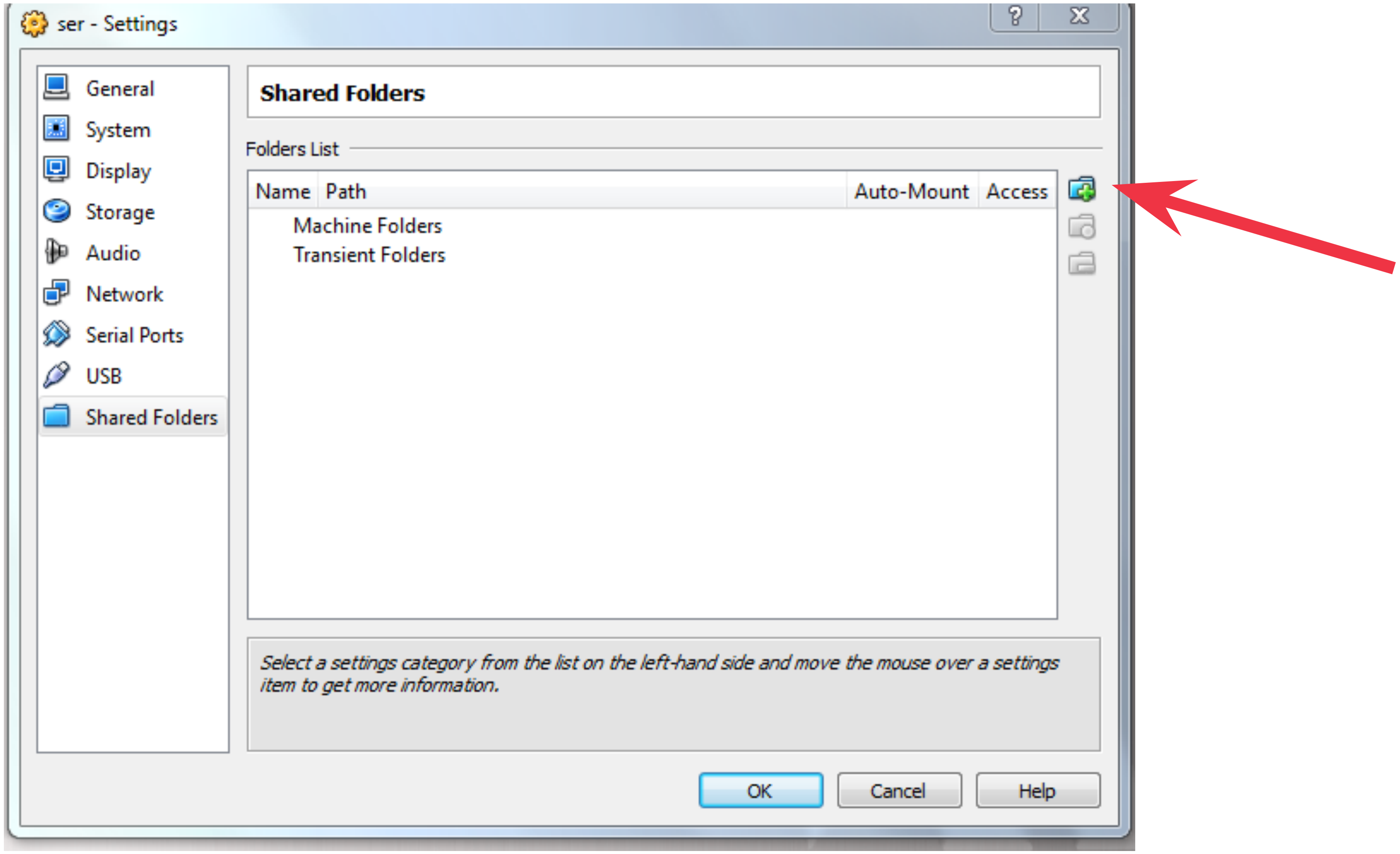
Inside your VM, click on Devices in the main menu bar and select Shared Folders ...



NowSecure™

## Enable File Sharing - VirtualBox

Click the ADD icon to add a new shared location:



**Shared Folders**

Folders List

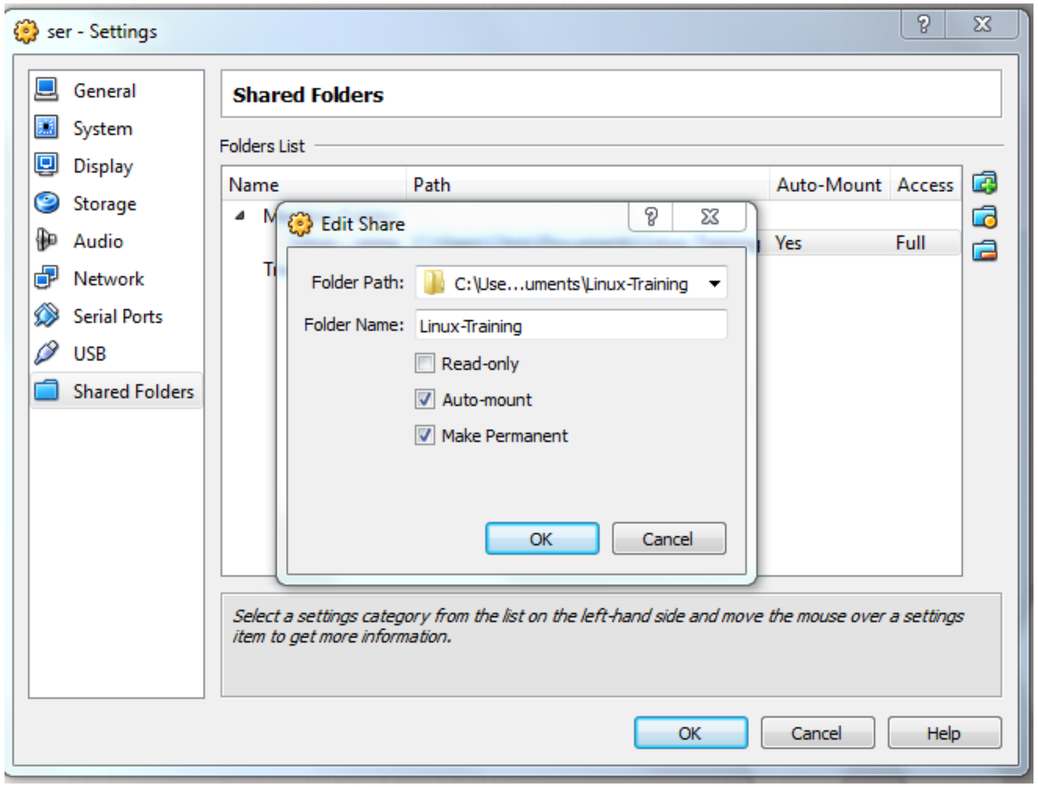
| Name | Path | Auto-Mount | Access |
|---|---|---|---|
| Machine Folders | | | |
| Transient Folders | | | |

*Select a settings category from the list on the left-hand side and move the mouse over a settings item to get more information.*

NowSecure™

## Enable File Sharing - VirtualBox

Under the folder path select OTHER and navigate to the Linux-Training folder you created on your host machine.  Select Auto-mount and Make Permanent option and click OK on both dialog windows:



Folder Path: C:\Use...uments\Linux-Training

Folder Name: Linux-Training

☐ Read-only

☑ Auto-mount

☑ Make Permanent

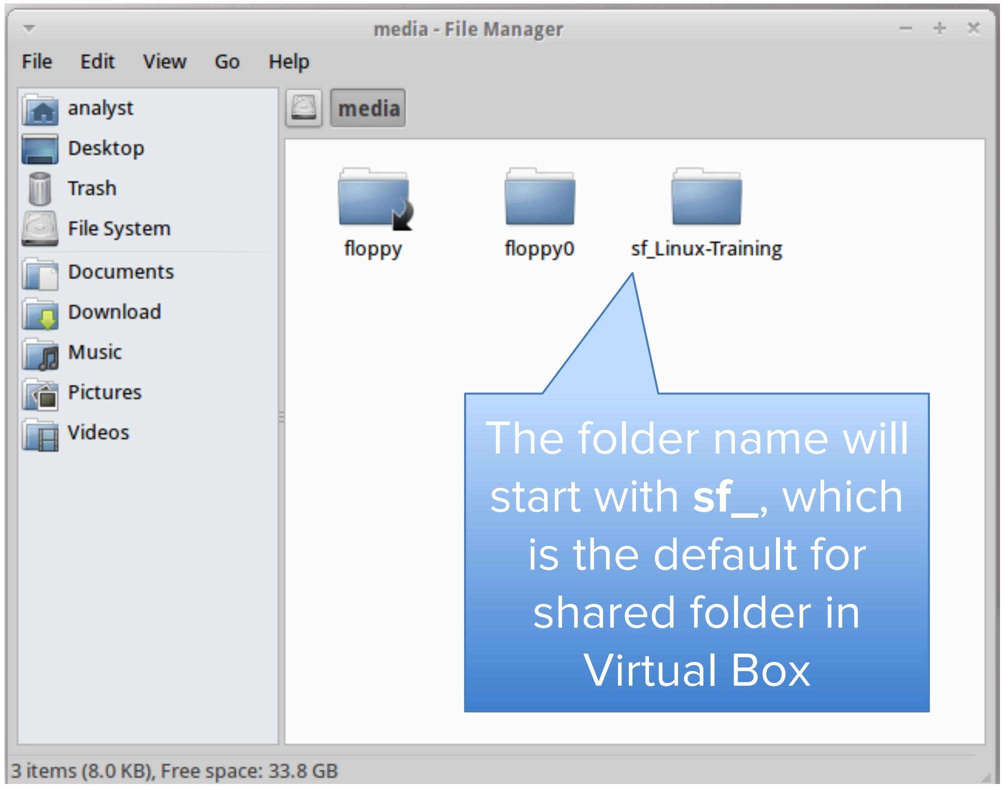
*Select a settings category from the list on the left-hand side and move the mouse over a settings item to get more information.*

NowSecure™

## Enable File Sharing - VirtualBox

Reboot the virtual machine and log back in. You should now have access to the shared folder you just created, which can be accessed by navigating to the media folder located in the root of the file system:

Tip: to navigate to this new folder, click on the blue folder icon in your upper tool tray, then click file system, then media.



## Enable File Sharing - VirtualBox

You can now copy and paste files from your shared location on your host machine to your Ubuntu virtual machine.

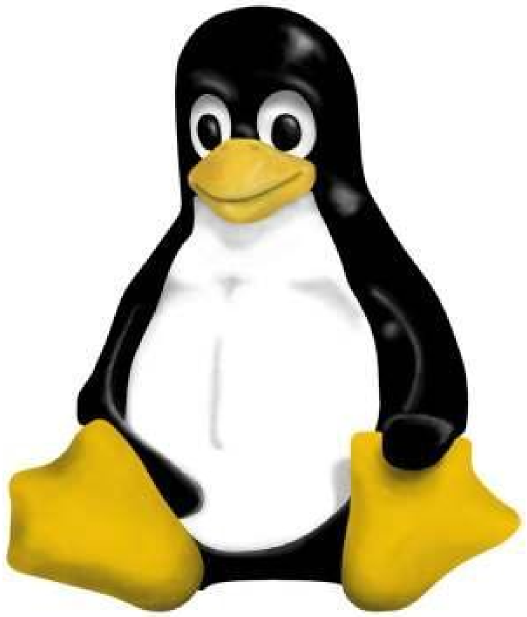
Take the time now to download the remaining files and save them to this directory.

Once you have finished downloading all the required course files your setup is complete!

## Training Outline

Here is a brief outline of the material that will be presented in this Training. Each module will have a short quiz at the end to assist in the application of the data you have just learned. Be sure to download the Linux Command Cheat Sheet to help you as you go through the training.

1) **Linux OS Overview**
    a. Overview
    b. File system
    c. Basic directory structure
    d. File Permissions
2) Linux Commands
    a. Basics
    b. Quiz: Linux Commands
3) Forensic Analysis Tools
    a. Lab: Analysis Part 1
    b. Lab: Analysis Part 2
    c. Lab: Analysis Part 3



## Why Linux?

Provided an examiner is familiar with the commands, Linux tools can be very powerful and flexible.

Much of what we do in mobile forensics, and especially in Android, revolves around a Linux environment. A strong foundation is essential to truly understand how your forensic utilities are affecting your device and data.

Linux
Just because sometimes its fun to feel smarter than people who can't use a command line

## Why Linux?

Some other advantages of using Linux as your forensics platform include:

- Open source development community with a plethora of free tools available including the Sleuth Kit (TSK)
- These tools will allow you to quickly search through a significant amount of data in a small amount of time
- Sometimes automated forensic tools don't work so you have to resort to manual methods

## Why Linux?

Parsing the data manually in Linux will give you an entirely new perspective on the power of the automated forensic suites, as well as a better understanding on how they function behind the scenes.

As a forensic investigator, understanding how your tools function is as crucial as recovering the evidence.  You can be sure that if you can't explain how the data was recovered the opposing counsel will be there to discredit your reputation as an expert.

## Linux OS Overview

Here are some interesting facts to help familiarize you with the Linux basics…

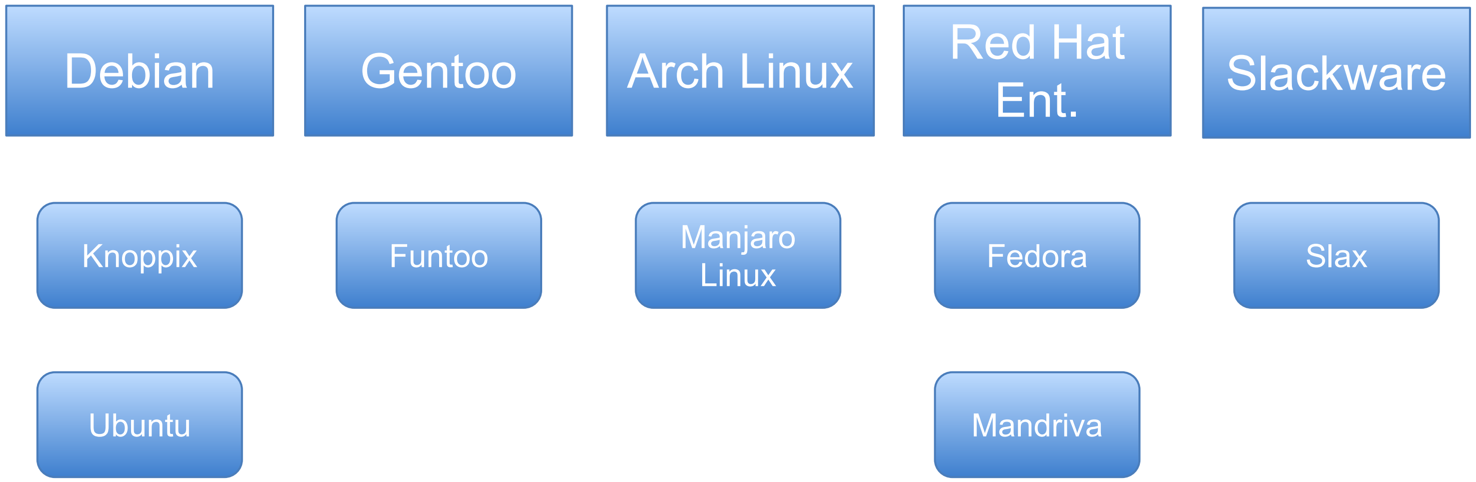
- Linux was originally created by Linus Torvalds, a young student from Finland
- Version 1.0 of the Linux Kernel was first released in 1994 (the latest stable version is v3.17.1)
- The kernel was developed under GNU General Public License (this means it's free!)
- Many apps now run on Linux, though there were not many available when Linux first started out

NowSecure™

# Linux OS Overview

There are Several hundred distributions of Linux in existence and they are all based on the same Linux kernel.

Developers can customize their distribution by adding their own tools and programs to each individual build.



## Linux OS Overview

Here are some common distributions that you may have heard of:

- Fedora (Red Hat)
- openSuSE (Novell)
- Ubuntu (Canonical Ltd.)
- Mandriva Linux (Mandriva)
- Knoppix (Live CD distribution)
- Debian (dev community)
- Gentoo (dev community)

So you can see that there are several iterations of Linux with each one offering something slightly different in the user interface.

Next let’s dive a little deeper into the construct of Linux!

## The Linux Kernel Layers

Linux is broken down into two layers: the user space and the kernel space.

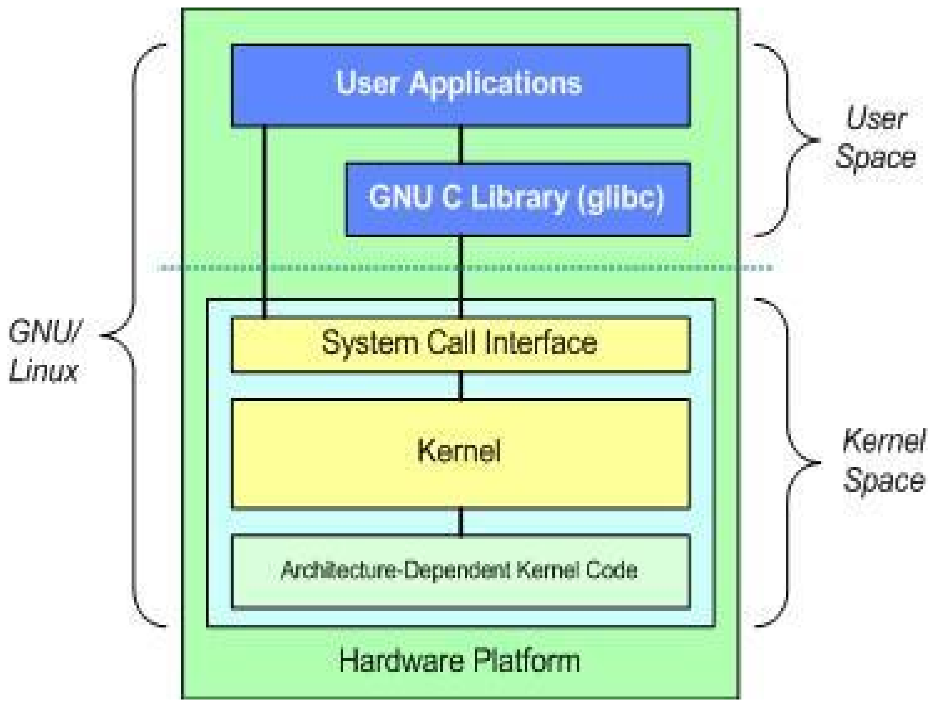
The user space is where apps are executed and the kernel space is where the Linux kernel exists.

In the next few slides we'll discuss these spaces a little bit more in depth.

# The Linux Kernel Layers

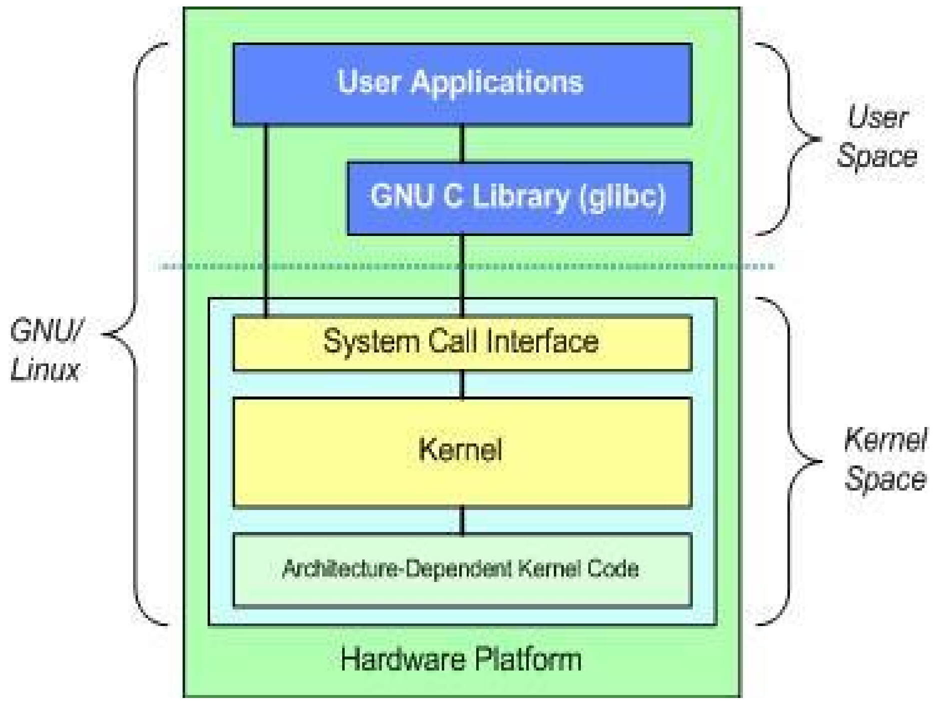
The two layers occupy different protected address spaces, and the user space is completely customizable - hence the different distributions of Linux.

For example, you will find different base applications installed on Red Hat than you would on Ubuntu.



# The Linux Kernel Layers

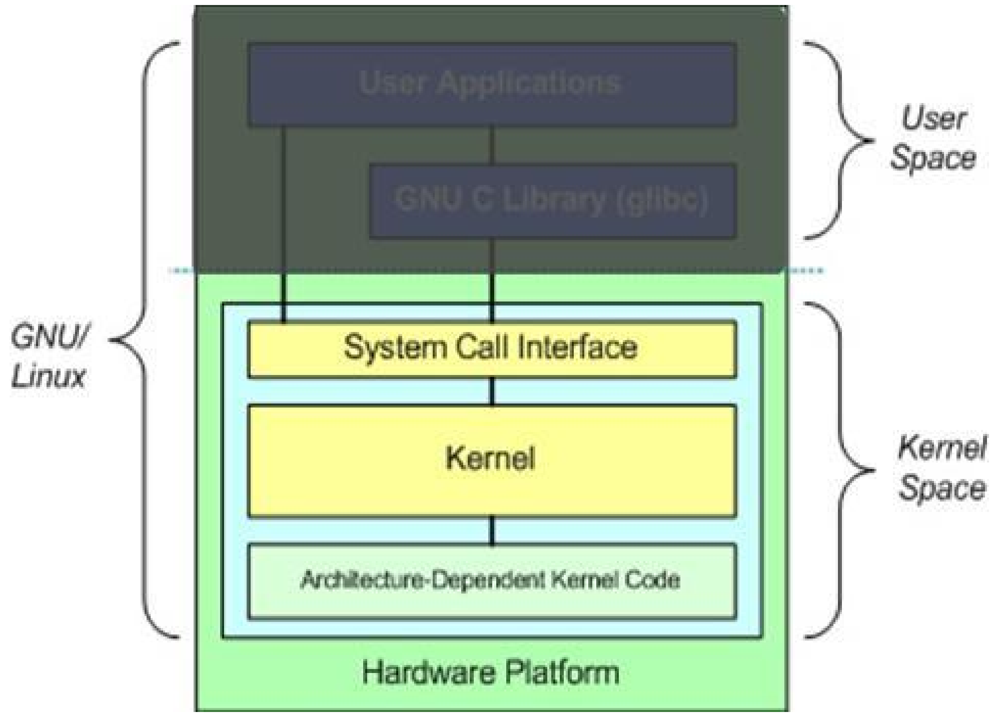
The GNU C Library (glibc) is contained in the user space and provides the interface to translate between the user level and kernel space.



# The Linux Kernel Layers

The Base Kernel is common across distributions and is layered into several subsystems. It's not important to go into those subsystems in great detail but be aware that it takes care of certain functions that include:

- Process management
- Virtual File System
- Network Stack
- Device Drivers
- Architecture definition (explained on the next slide)

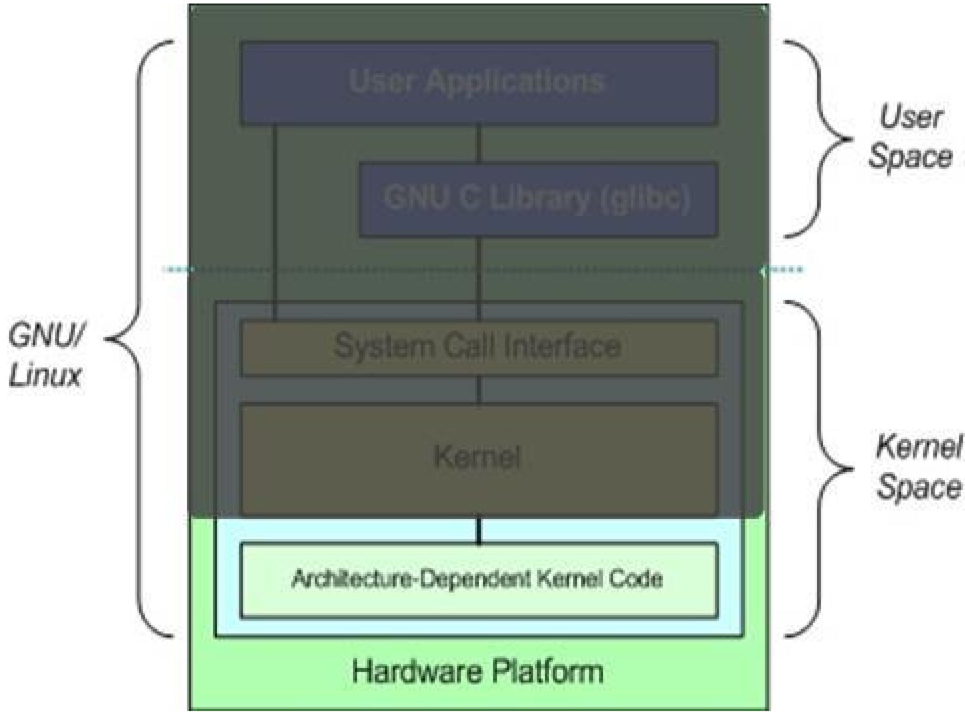


# The Linux Kernel

The Architecture-Dependent Kernel Code contains the kernel code for the given architecture.

This code is located in ./linux/arch and contains subdirectories that focus on different aspects such as boot, memory management, and others.

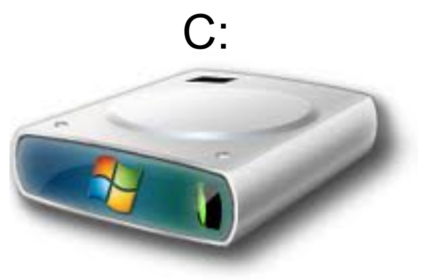
Again, remember that the Architecture-Dependent Kernel Code is not customizable and is released as a Linux kernel version. Only the User Space can be customized for each distribution.



## The Linux File System

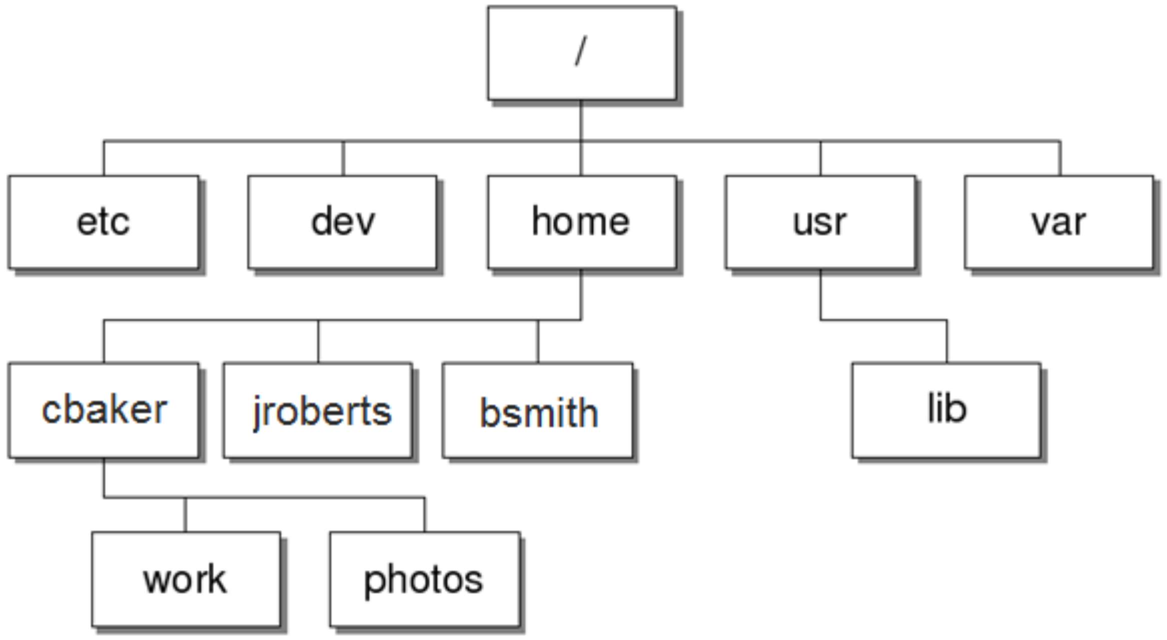
Now that you're an expert in Linux history and basic background let's start digging into the file system a little deeper.

In Linux, all files are part of the same file structure, as compared to Windows which has separate drives (C: - hard disk; A: - floppy, etc.).



## The Linux File System

The entire directory structure in Linux falls under root, which is classified as a "/". In the graphic below, you can see that the file system representation is similar to an organizational chart, with root at the very top followed by some core directories and branching out to the individual subdirectories where additional data is stored.



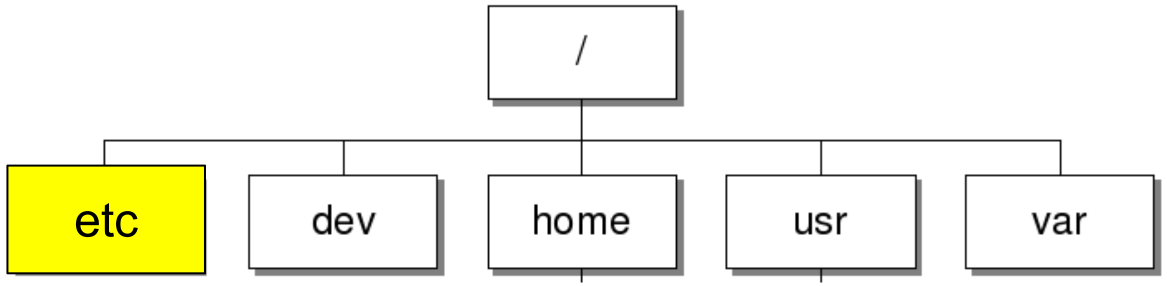
# The Linux File System

Let's explore a few of the main core directories that fall under root.

/etc – contains the configuration files for the system.  These configuration files are stored in text documents and can be edited if necessary.

An example of a configuration file is /etc/inittab, which defines the processes that run at startup and what certain key combinations can do.

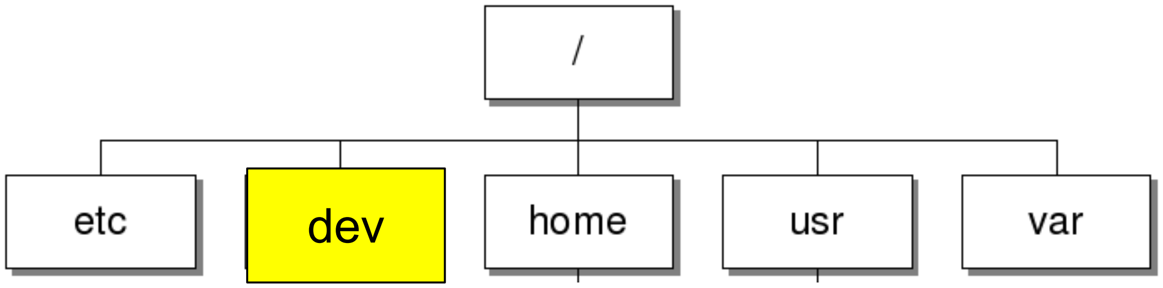
Another example is /etc/fstab, where the individual mount points such as external hard drives are defined.



# The Linux File System

/dev – contains devices that are available to the Linux system, such as USB drives.  Remember that Linux treats additional devices like files so you can read/write the contents like looking at a file.

Some examples of Linux devices would include the CD Rom drive (/dev/cdrom), USB drive (/dev/sdb), and the hard drive (/dev/hda).

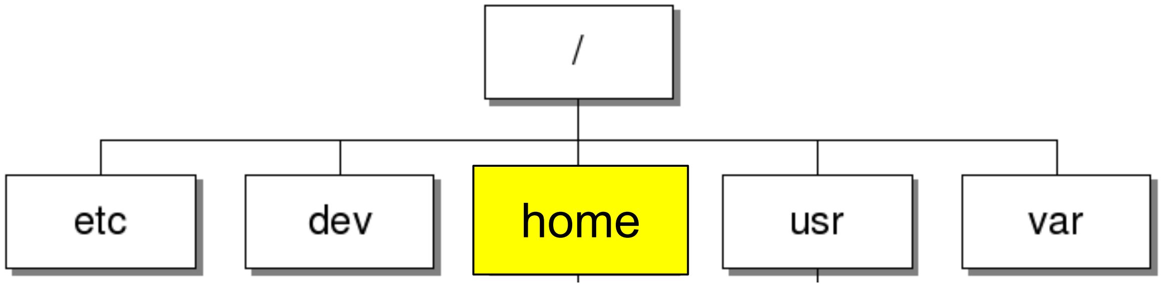


## The Linux File System

/home - contains all the users that have been created on the system.  You can compare this to the users directory in Windows where you will find a folder belonging to each user profile.

Any files relating to a specific user are secure in his/her profile as long as the files or folders are stored in the /home directory.  This is also the default path when you open a new terminal window… /home/<username>

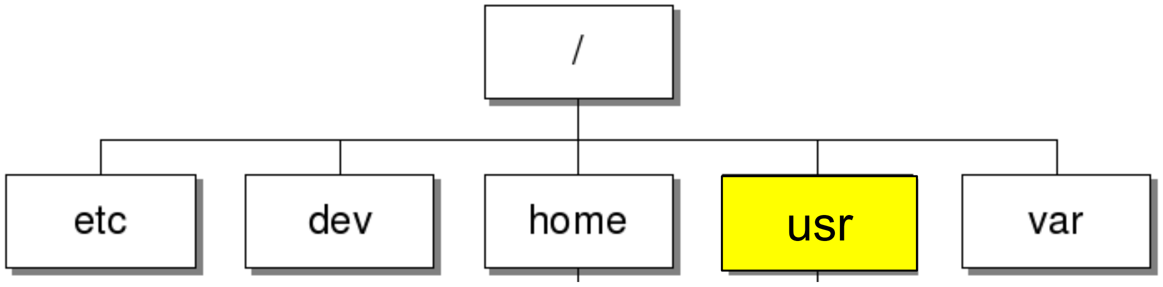
/home/<username> is synonymous with ~, which will be explained during the command line practical exercises later in this training.



# The Linux File System

/usr – contains installed applications and information about those applications.  This is usually the largest directory on the Linux system.

Some examples of what you might find in the /usr directory include application documentation (/usr/doc), source code for installed applications (/usr/src), and any files that might be shared among several applications (/usr/share).

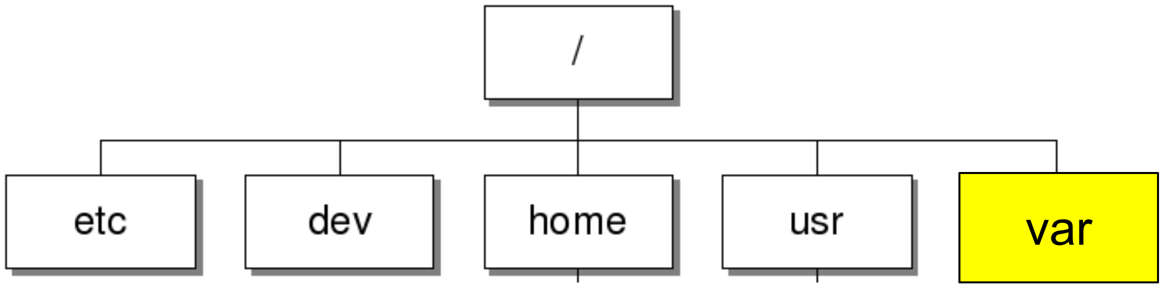


NowSecure™

# The Linux File System

/var – contains variable data, that is, files that are constantly changing while the system is running.

Some examples include system log files (/var/log), documents that are spooled for printing (/var/spool), and even incoming and outgoing mail (/var/mail).



## Linux - Common Disks

We discussed earlier how Linux organizes attached devices into subdirectories instead of volumes. Let’s take a closer look at what some of these devices might look like and what it all means.

On your Linux workstation, open a terminal window by hovering over the bottom of your screen and clicking the terminal shortcut.

Type the following command at the terminal prompt:

```
cd /dev
```

*(`cd` means change directories. Since the terminal by default starts in the home directory we had to change over to the /dev directory before we could run the next step)*

# Linux - Common Disks

Next perform a directory listing of /dev by typing:

```
ls
```

*(`ls` means list.  This lists out the contents of a directory in a very basic output. There are some options that will present you with more information such as `ls -l`, which uses a long listing format showing file permissions and MAC information.  In this case a simple `ls` did the trick).  It is the same as doing a dir in a Windows command prompt.*

You will see that there are several devices attached to the system, which all play an important role in how the system functions.  In the next slide we will examine some of the more important ones.

## Linux - Common Disks

Every system is a little different since not all devices are common among them.  Some of the most common devices include:

- hda/hdb, which is a IDE hard drive; “a” for master drive, “b” for slave
- sda/sdb, which represents SATA/SCSI devices that are connected over USB and/or firewire
- loop, which is used to “emulate” a block device (hard drive, CD-ROM, etc).  You would use a loop device if you want to mount an image file in the Linux file system.

You will also notice that there may be a number following the device name, such as sda1, sda2, etc.  This number represents partitions within that specific drive, whereas the entire drive would just be sda.

*Note: Since our Ubuntu workstation is running within a virtual machine you will not see an entry for hda because it is using a virtual disk on the host computer.*

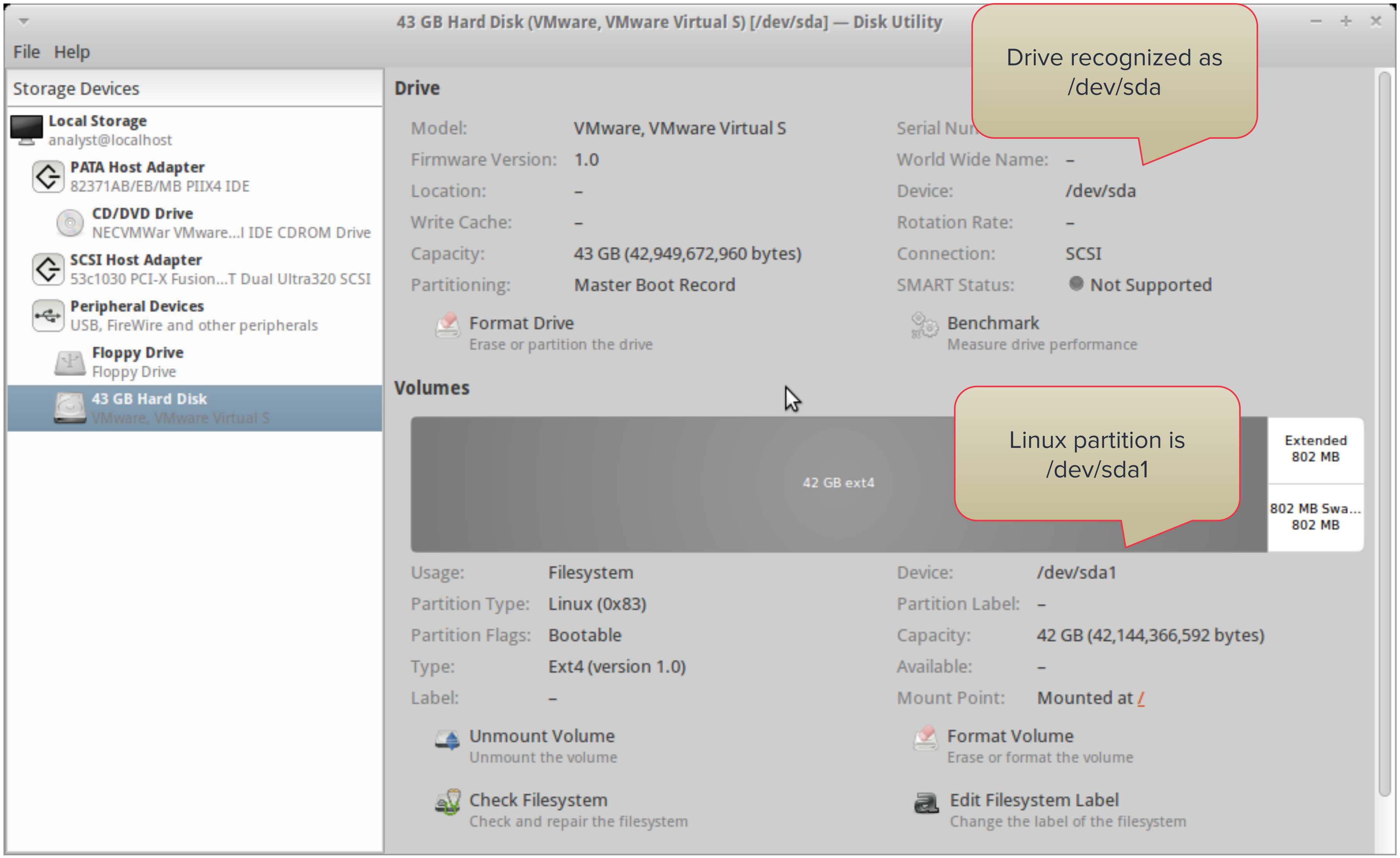
## Linux - Common Disks

Since we recognize that many people that are taking this course do not operate well in a “command line” interface, we would also like to point out that the device information is also available from within the graphical user interface.

To access the device (drive) information go to the main menu in your Linux distribution (mouse icon in Ubuntu) and select settings and then Disk Utility. A screenshot of what the Disk Utility looks like is on the next slide so you can see how the system organizes the attached drives, as well as the device name.

In our example we are running Ubuntu in a virtual machine so the main hard drive is recognized as /dev/sda, if you are not using our virtual machine your system may be different. The Linux partition is /dev/sda1.

# Linux - Common Disks



43 GB Hard Disk (VMware, VMware Virtual S) [/dev/sda] — Disk Utility

File Help

Storage Devices

**Local Storage** analyst@localhost
**PATA Host Adapter** 82371AB/EB/MB PIIX4 IDE
**CD/DVD Drive** NECVMWar VMware...l IDE CDROM Drive
**SCSI Host Adapter** 53c1030 PCI-X Fusion...T Dual Ultra320 SCSI
**Peripheral Devices** USB, FireWire and other peripherals
**Floppy Drive** Floppy Drive
**43 GB Hard Disk** VMware, VMware Virtual S

**Drive**

| | | | |
|---|---|---|---|
| Model: | VMware, VMware Virtual S | Serial Nu | |
| Firmware Version: | 1.0 | World Wide Name: | – |
| Location: | – | Device: | /dev/sda |
| Write Cache: | – | Rotation Rate: | – |
| Capacity: | 43 GB (42,949,672,960 bytes) | Connection: | SCSI |
| Partitioning: | Master Boot Record | SMART Status: | ● Not Supported |

Format Drive — Erase or partition the drive
Benchmark — Measure drive performance

**Volumes**

| | | | |
|---|---|---|---|
| Usage: | Filesystem | Device: | /dev/sda1 |
| Partition Type: | Linux (0x83) | Partition Label: | – |
| Partition Flags: | Bootable | Capacity: | 42 GB (42,144,366,592 bytes) |
| Type: | Ext4 (version 1.0) | Available: | – |
| Label: | – | Mount Point: | Mounted at / |

Unmount Volume — Unmount the volume
Format Volume — Erase or format the volume
Check Filesystem — Check and repair the filesystem
Edit Filesystem Label — Change the label of the filesystem

Drive recognized as /dev/sda

Linux partition is /dev/sda1

NowSecure™

## Linux - File Permissions

Having an understanding of how the directory structure is organized on Linux is key, along with being able to navigate through a terminal session using command line.

As demonstrated in the previous slides you can see that there is a fairly robust GUI interface that allows the user to accomplish most tasks with ease. For example, similar to Windows, you can right click a file in Linux and display the properties of that file.  From there, you can perform a few functions, such as view and modify file permissions.

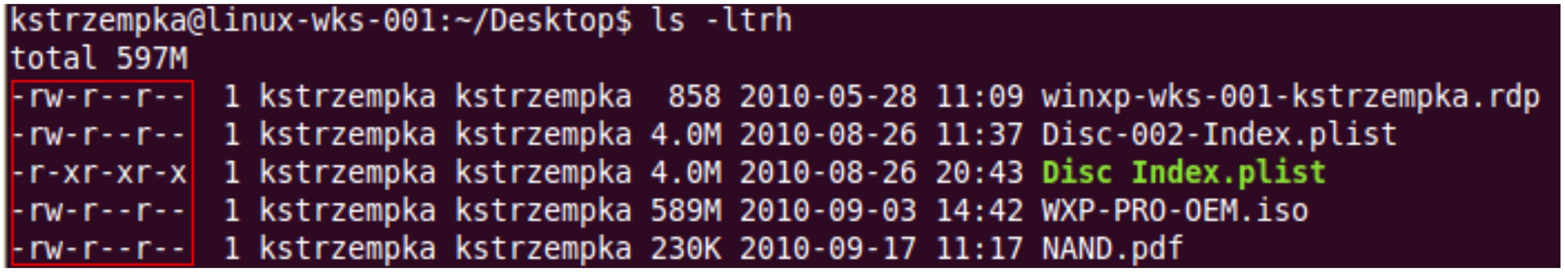
That said, we will focus primarily on the command line interface since many of the commands we run in the mobile forensics environment will be in a terminal session.

# Linux - File Permissions

We can take a look at file permissions in a terminal session by typing in the following command: **`ls -ltrh`**

You remember from a previous exercise that **`ls`** means listing and when we add the **`–l`** switch it means long listing format.  The other switches are as follows:

- **`-t`** means sort by modification time
- **`-r`** means reverse order while sorting
- **`-h`** means convert the sizes to human readable (i.e 4.0M, 230k, etc.)



```
kstrzempka@linux-wks-001:~/Desktop$ ls -ltrh
total 597M
-rw-r--r--  1 kstrzempka kstrzempka  858 2010-05-28 11:09 winxp-wks-001-kstrzempka.rdp
-rw-r--r--  1 kstrzempka kstrzempka 4.0M 2010-08-26 11:37 Disc-002-Index.plist
-r-xr-xr-x  1 kstrzempka kstrzempka 4.0M 2010-08-26 20:43 Disc Index.plist
-rw-r--r--  1 kstrzempka kstrzempka 589M 2010-09-03 14:42 WXP-PRO-OEM.iso
-rw-r--r--  1 kstrzempka kstrzempka 230K 2010-09-17 11:17 NAND.pdf
```

## Linux - File Permissions

You may be wondering what all the dashes and letters mean on the far left. Those are the actual file permissions and are broken into three security groups; owner, group, and world.

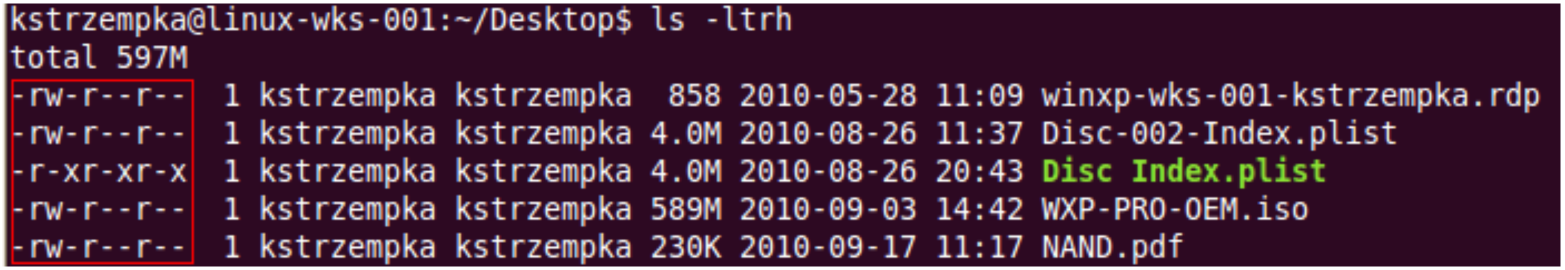
File permissions are shown in the following format:

- **-** = no permissions
- **r** = read
- **w** = write
- **x** = execute

So, above…

Line 1: read/write (owner), read (group), read (world)

Line 3: read and execute for all groups



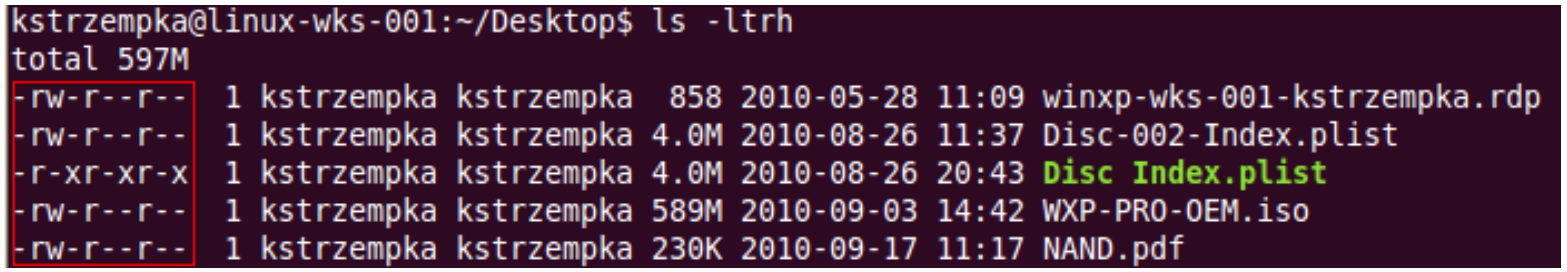
```
kstrzempka@linux-wks-001:~/Desktop$ ls -ltrh
total 597M
-rw-r--r--  1 kstrzempka kstrzempka  858 2010-05-28 11:09 winxp-wks-001-kstrzempka.rdp
-rw-r--r--  1 kstrzempka kstrzempka 4.0M 2010-08-26 11:37 Disc-002-Index.plist
-r-xr-xr-x  1 kstrzempka kstrzempka 4.0M 2010-08-26 20:43 Disc Index.plist
-rw-r--r--  1 kstrzempka kstrzempka 589M 2010-09-03 14:42 WXP-PRO-OEM.iso
-rw-r--r--  1 kstrzempka kstrzempka 230K 2010-09-17 11:17 NAND.pdf
```

NowSecure™

# Linux - File Permissions

The first character specifies type of file and uses the following designators:

- "**-**" = a file
- **d** = directory
- **b** = block devices
- **c** = character device
- **l** = link to another file/directory



```
kstrzempka@linux-wks-001:~/Desktop$ ls -ltrh
total 597M
-rw-r--r--  1 kstrzempka kstrzempka  858 2010-05-28 11:09 winxp-wks-001-kstrzempka.rdp
-rw-r--r--  1 kstrzempka kstrzempka 4.0M 2010-08-26 11:37 Disc-002-Index.plist
-r-xr-xr-x  1 kstrzempka kstrzempka 4.0M 2010-08-26 20:43 Disc Index.plist
-rw-r--r--  1 kstrzempka kstrzempka 589M 2010-09-03 14:42 WXP-PRO-OEM.iso
-rw-r--r--  1 kstrzempka kstrzempka 230K 2010-09-17 11:17 NAND.pdf
```

## Linux - File Permissions

Now that you're familiar with viewing the permissions of a file, let's take a look at how to modify them.

Various commands can be used to modify permissions, such as **`chmod`** and **`chown`** for example.  To change permissions, you must understand the numerical (or "octal") value for read, write, and execute assignments. Permissions are calculated based on the following:

| | |
|---|---|
| Read | = 4 |
| Write | = 2 |
| Execute | = 1 |

Read+write+execute = 7 so to assign full permissions for all 3 groups (owner, group, world), the command would be: **`chmod 777`** ***`filename`***

You will get some practical experience with this shortly!

## The “chown” Command

The **`chown`** command can be used to modify the owner or group permissions separately, or both can be modified within a single command.

## The “chown” Command

There are a few options when running the **`chown`** command with one only modifying the owner, and the other modifying both the group.  You can also modify the owner and group at the same time.

If only a user is given, the owner will change (not the group).

```
    $ chown jsmith image.dd
```

If *username:groupname* is given, both owner and group will change

```
    $ chown jsmith:forensics image.dd
```

NowSecure™

## The “chown” Command

When using chown to modify only the group a colon is used before the group name:

```
$ chown :forensics image.dd
```

*This is essentially the same as using the “chgrp” (change group) command:*

```
$ chgrp forensics image.dd
```

## Super User Permissions

On occasion, it may be necessary to run a command with escalated permissions, similar to in Windows where you may have to run a command as administrator.

In Linux we use what is called sudo. Sudo is short for "super user do," which is essentially saying do this command as admin. The system will prompt for your password when executing a command as sudo.

If you receive a permissions error when running a command, it likely means sudo is required.

NowSecure™

## Super User Permissions

The command below shows how to change the permissions of a file. In this example we change the permissions of image.dd:

```
analyst@ubuntu:~$ sudo chmod 755 image.dd

[sudo] password for analyst:

analyst@ubuntu:~$
```

The password is not visible as it is typed in but is required to accomplish the change. The password  in this case is the same as your login password.

Once the change is made it will just return you to the prompt.

NowSecure™

## Summary

In this section we discussed the basics of Linux, giving you an understanding of how the Linux file system is structured and where data is stored.  We also introduced file permissions and how to modify the security settings of a file or directory.

You are now ready to take the quiz for this section and then move on to the next module, “Basic Linux Commands.”

## Basics: Check Your Knowledge!

1. Linux is developed under the GNU (General Public License), which means that it's available for free download (T/F).
2. Which of the following are all distributions of Linux
   A. Red Hat, Mandriva, Knoppix, Debian, Gentoo
   B. Ubuntu, Mandriva, OpenSuSE, RoundTree
   C. Knoppix,  Fedora, Mandorin, Gentoo
   D. Options 1 & 3 above
3. The Application Space is bundled as part of the Linux kernel distribution and is not customizable (T/F).

## Basics: Check Your Knowledge!

4. When a new terminal window is open what will be the starting path by default?
   A. /home/<username>
   B. /usr/<username>
   C. /var/<username>
   D. /dev/<username>
5. Applications are stored in the /var directory (T/F)
6. To perform a directory “long” listing you would do the following:
   A. Open a terminal window, type ls at the prompt
   B. Open a terminal window, type dir  at the prompt
   C. Open a terminal window, type ls -l at the prompt
   D. Launch a GUI file manager from the desktop
7. You would find print spool documents located in
   A. /usr
   B. /var
   C. /etc
   D. Linux doesn’t have a print spooler as it outputs the file directly to the printer

## Basics: Check Your Knowledge!

8. You do a directory listing of /dev and find several instances of sdb (i.e sdb1, sdb2, etc…). What do these represent?
   A. Devices connected through USB or Firewire
   B. Device driver files
   C. Primary hard drives
   D. Partitions

You perform a long directory listing and see the following permissions associated with an object:

```
-rw--r--r--
```

9. What type of object is this?
   A. Directory
   B. File
   C. Symbolic link
   D. Named pipe

## Basics: Check Your Knowledge!

10. What are the group permissions associated with this object?
    A. Read only
    B. Read/write
    C. Read/write/execute
    D. No permissions

    You perform a long directory listing and see the following permissions associated with an object:

    ```
    -rw--r--r--
    ```

11. What command would you use to change the permissions of this file?
    A. **`chown`**
    B. **`sudo`**
    C. **`chmod`**
    D. **`ls -cp`**

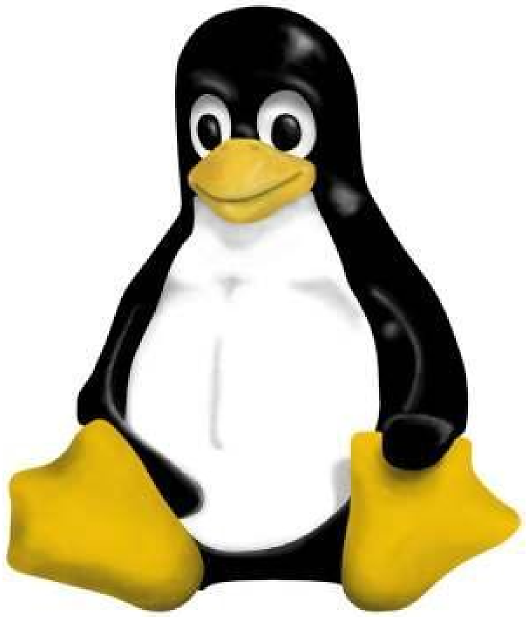
NowSecure™

## Basics: CYK! - Answer Key

1. True
2. A
3. False
4. A
5. False
6. C
7. B
8. D
9. B
10. A
11. C

NowSecure™

## Training Outline



## Intro to Command Line Interface (CLI)

Understanding the command line is important when working in the field of forensics. In fact, most of what we do in Android forensics is done in a Linux command line interface.

## Intro to Command Line Interface (CLI)

In this section, we will familiarize you with some essential command line commands and get some hands on practice with running the commands in your Ubuntu virtual machine.

These next commands are very basic, but they will be your core foundation for everything you do in Linux.  Be sure you understand the commands before moving on as it will make it easier as we progress through the training.

## The tab key

First and foremost you should know that the tab key on your keyboard is very useful when working in a terminal interface.

## The tab key

Linux will autocomplete a path or filename when you press the tab key, which relieves you from having to worry about case sensitivity or having to type out long file names or paths into the command line.

For example, below is a directory listing (**`ls`**) of the via folder on your Ubuntu machine:

```
/opt/via$ ls

Android
iPhone
live-exploits
live-recovery
```

If we were to type **`cd A`** and press the tab key it will auto fill the Android directory name. Notice we use a capital “A” as Linux is case sensitive.

## The tab key

Another example of the same directory output:

```
/opt/via$ ls

Android
iPhone
live-exploits
live-recovery
```

If we type **`cd l`** and press the tab key, it will autofill the live directory name up to the point where it is no longer common, which in this case is **`live-`**. At this point you could type an r or an e then hit tab again for Linux to autocomplete the rest of the directory or file name.

This is especially handy if there are spaces in the folder or file name (although you should never use spaces when naming folders and files in Linux if you can avoid it).

## The tab key

Let’s go ahead and practice on your Ubuntu virtual machine.

1. Open your virtual machine and log in using the username and password you created during VM setup.
2. Open a terminal window. There is a shortcut on the bottom toolbar and if you hover your mouse over the bottom of the screen you will see the black window icon
3. Notice that the default prompt is **`analyst@ubuntu:~$`** This is your home directory and is the default starting point for the command prompt.

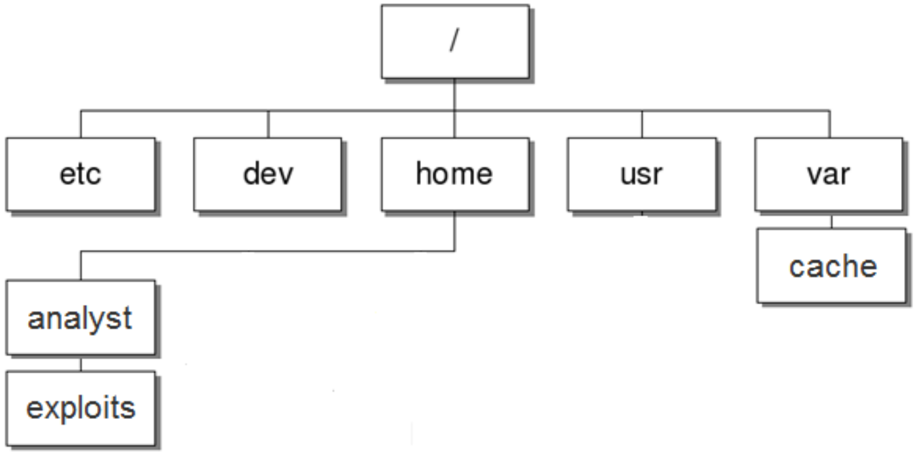
## The tab key

1. Practice doing a directory listing by typing **`lsx`** at the prompt. You will see all the folders within your home directory listed.
2. Now type **`ls -l`** (that is an L not a one) to get a long listing of the directory. You should now see each folder listed along with the permissions and MAC data for each folder.
3. CD into the var directory by typing **`cd /var`**. Remember that to go deeper into a directory you wouldn't use the / in front of the folder name. To change into an entirely new directory under root you would use the **`/`**, as in this case with changing into the var directory.

## The tab key

To solidify this point, assume you are in your home directory (/home/analyst). In the diagram below you can see that var falls under **/**, so when you cd into var you do this: **`cd /var`**

Once you are in the var directory and you want to go to the cache directory you can just type cd cache to get there. **`cd /cache`** will not work because cache doesn't fall under root.



# The tab key

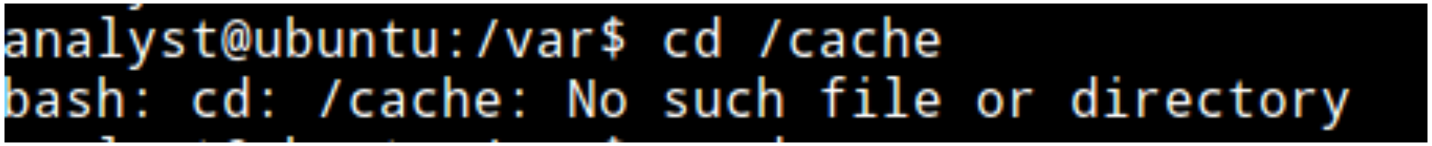
1. For practice let's try changing into the cache directory located in /var. In this case since cache falls under /var you would just type **`cd ca`** and hit the tab then the enter key. Notice that cache auto completes.
2. Now go back one directory to get back into /var. You can do this by typing **`cd ..`**
3. Note that the prompt shows you where you are. In this case you see ***`analyst@ubuntu:/var$`***

# The tab key

1. For fun, try going back into cache by typing **`cd /ca`** and press tab to see what happens. Note that Linux cannot auto complete the folder name.
2. Type **`cd /cache`** and hit enter.

   Notice that you get a bash error stating that there is **`No such file or directory`**. This is because cache does not fall directly under root.

   This is where tab will save you some time. Since tab didn't autocomplete **`/ca`** you know that something is wrong!!



```
analyst@ubuntu:/var$ cd /cache
bash: cd: /cache: No such file or directory
```

## The tab key

Now type **`cd`** and notice how you are taken back to your home directory. You know you are in your home directory because the prompt says

**`analyst@ubuntu:~$`**

Remember that the ~ is synonymous with the path /home/<username>

## The tab key

Hopefully you have a feel for how useful the tab key can be. We have one more tab function to cover before moving: the double tab feature.

In your terminal window, you should be in your home directory. Type **`cd /v`**, then hit the tab key and enter. This will change directories into the var folder.

Now type **`cd`**, hit the spacebar, then tap the tab key twice. You should see that it lists all the possible directories under /var.

If there were only one folder under /var, it would auto populate the directory name as the only choice.

## The tab key

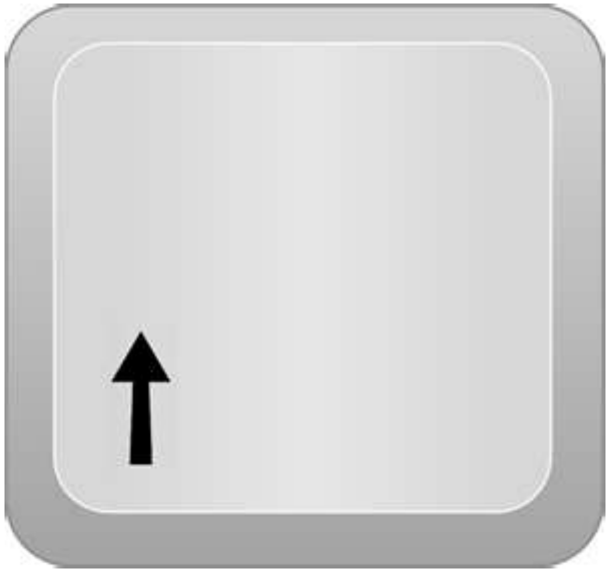
To recap, you have learned that using the tab key will autocomplete a directory or file name.

If you hit the tab key twice, it will list all the possibilities within the current directory.

Try to make a habit of using the tab key, because the entry will not autocomplete if the path or filename doesn’t exist. This will ultimately save you time!

## The up arrow

The up arrow is also a very useful tool since it allows you to scroll through previous commands, starting with the most recent.



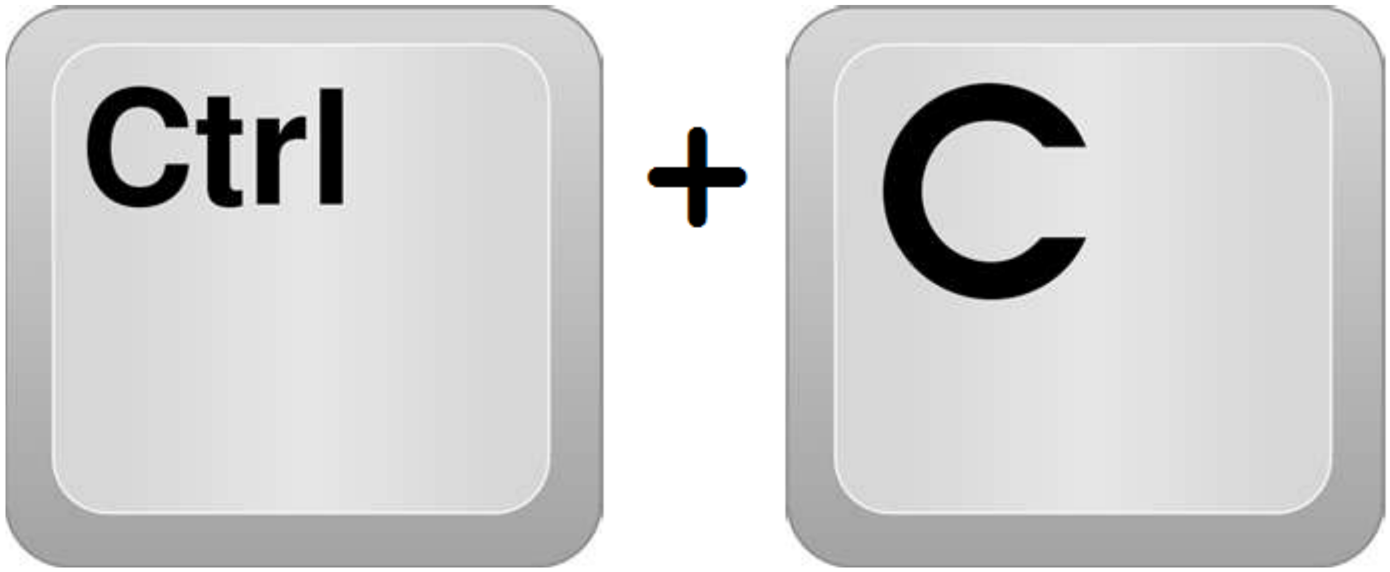
## The up arrow

Practice pressing the up arrow a few times in your terminal window to see all the commands that you have previously entered. This little feature can save you a bunch of typing if you run the same command repeatedly.

You can use the up arrow to select a previous command, then use the left arrow to go to any point of the command that needs to be modified!

## Ctrl + C

When it appears that you are stuck in an infinite loop or a command line program doesn't seem to have any way out...



NowSecure™

## Ctrl + C

When you press the control key and the c key at the same time, Linux will immediately terminate whatever command or program is running in the terminal. It will then return you to a normal command line prompt in whatever directory you were in when the command was executed.

This is a useful tool if you find yourself in a never-ending process!  When all else fails, ctrl + c your way out!

# manual: “man”

Ever wonder how something works and reference the manual? Linux has a manual for nearly all commands that can be referenced with simple command line syntax.

To take a look at a manual you would type the following:

```
$ man [-][-k keywords] command
```

Example: List information on the “mount” command

```
$ man mount
```

Example: Search all manuals containing “zip”

```
$ man –k zip
```

Go ahead and try these in your terminal window!

NowSecure™

## Help

Help is another good way to figure out the syntax or options of a specific command. If you are not sure how to run a command, simply ask for help!

Standard syntax:

```
    $ mount -h
```

or

```
    $ mount --help
```

**Give it a try in your virtual machine!**

## The q command

The q command will get you out of a manual or help page. In essence, **`q`** means “quit.” This is useful when you are viewing a manual and want to exit back to the prompt. Just type **`q`**!

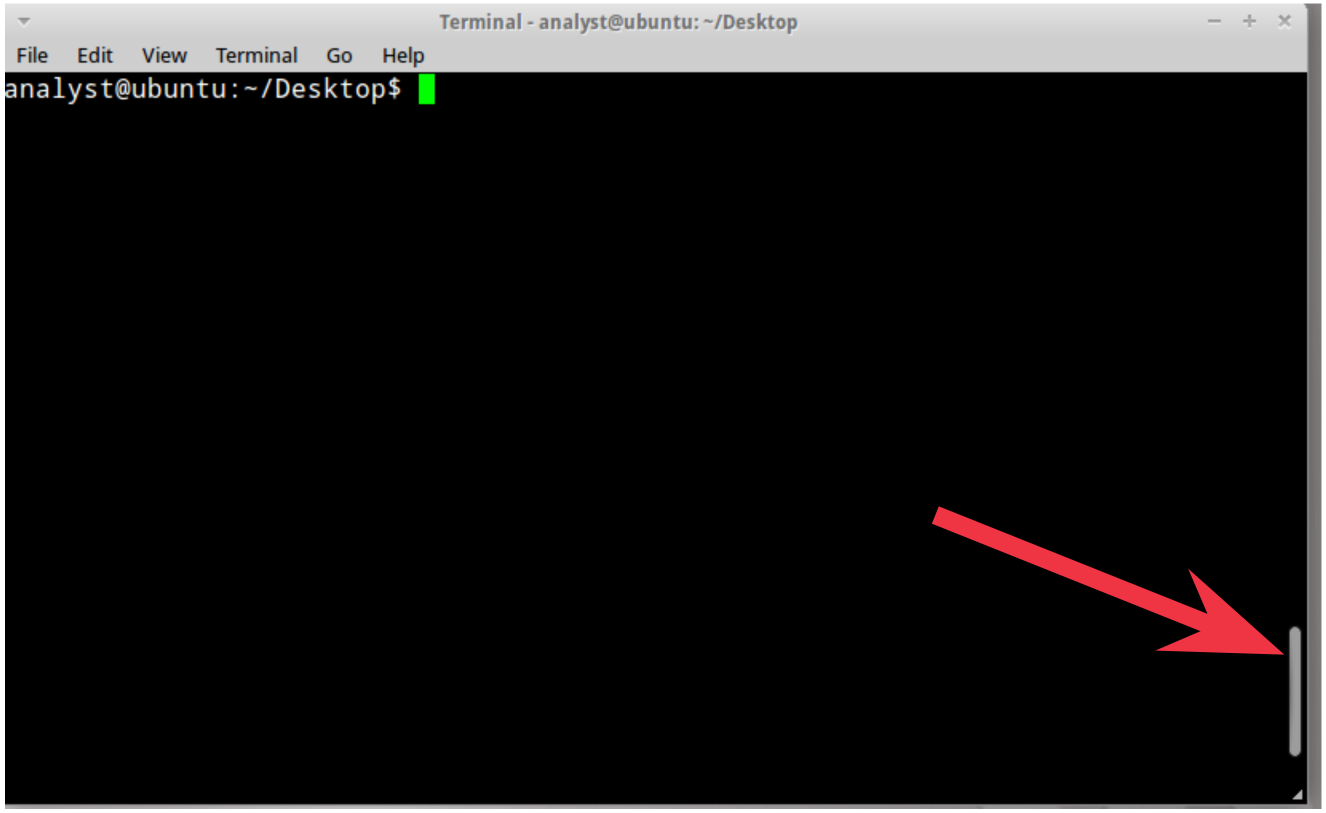
## The clear command

On occasion it is desirable to have a clean slate to work from in your terminal window.  Let’s say you want to clear the screen shown below, just type **`clear`** and hit enter!

NowSecure™

## The clear command

It is worth mentioning that the clear command will not erase everything but rather just puts the prompt at the top of the screen.  You can still scroll up to see the previous data.


Terminal - analyst@ubuntu: ~/Desktop
File Edit View Terminal Go Help

```
analyst@ubuntu:~/Desktop$ 
```

## Summary

Now that we have introduced you to a variety of simple commands, you should feel a little more comfortable navigating your way through the command line structure.

Remember that if you are ever wondering how a command works, just ask for help by typing **`-h`** after your command!

Next, we'll cover some more information on relative paths and advanced commands!

## Absolute vs. Relative Paths

The Absolute Path is the full path to a specific folder/file starting with root ("/"). We touched on this earlier but it is important to fully understand this concept. An example of an absolute path is below:

```
/home/jsmith/Desktop/folder
```

Notice how the path starts with a /. Anytime there is a / in the beginning of the path Linux will start at the root.

## Absolute vs. Relative Paths

The Relative Path is the path to a specific folder or file in relation to the location the user is currently in. In this case, we are not starting at the root but rather in the current location.

For example, Desktop/folder or ./Desktop/folder.

## Commands: Check Your Knowledge!

Open a terminal window in your virtual machine and change into “etc” folder located in the root directory. Your command should look like this:

```
analyst@ubuntu:~$ cd /etc
```

NowSecure™

## Commands: Check Your Knowledge!

1. What is one way to go back one directory? If you are not sure use your terminal to test the following commands to see which one will work.

    A. `$  cd ..`
    B. `$  dir ..`
    C. `$  cd..`
    D. `$  cd ../..`

## Commands: Check Your Knowledge!

2. Which of the following commands will take you back to your home directory? If you are not sure test each one in your terminal window.

   A. `$   cd ~`
   B. `$   cd`
   C. `$   cd /home/analyst (or current user name)`
   D. all the above

NowSecure™

## Commands: Check Your Knowledge!

3. What does the **`pwd`** command do?  Use your terminal to give it a try!

   A. Shows your password
   B. Shows your current machine name
   C. Shows your current directory
   D. This command doesn’t work!

## Commands: CYK! Answer Key

1. A (see below for explanation)
2. D
3. C

Comments on Question 1:

You found that only one of these commands will take you back one directory. Don't be confused by cd.. as that is a Windows command! Linux requires the space between **`cd`** and the `..`!

**`dir ..`** is incorrect because dir is a windows command.

**`cd..`** is incorrect because it is also a windows command. (remember the space!)

**`cd ../..`** is incorrect because it takes you back two directories.

## Additional Commands

Nice job!  You have made it through the basics and now it's time to cover some additional commands. The next few slides will show you some new tricks that will be extremely useful in mobile forensics!

## The “cat” command

There may be times where you need to see the contents of a specific file or script. In Android forensics we commonly cat out the partition table so we know what partition(s) we need to image.

To view the contents of a file there are two options: output to screen or output to a file.

Displays contents of a file to screen:

```
$ cat old.txt
```

## The “cat” command and redirecting output

Linux has a feature that allows you to redirect the output of a command into a file. By using the **>** symbol, you can output the results of the cat command into an entirely new file rather than outputting it on the screen. Here’s an example:

```
    $ cat old.txt > new.txt
    $ cat old.txt > ~/Documents/new.txt
```

As a side note, you can also use the redirect symbol for other functions such as listing out the results of a search in a file:

```
$  find ~/Documents -name “*.jpg” > my-pictures.txt
```

## The “cat” command

It may also be desirable to combine multiple files into one, such as when you have split images (commonly known as concatenating files). The cat command does the trick for this as well:

```
$ cat file1.txt file2.txt file3.txt > final.txt
```

NowSecure™

## The less and more commands

If you run the cat command against a very large file, you will find that the contents will fly through the screen without it stopping until the command reaches the end of the file.

Thankfully, there is a way to view the contents one page at a time – using the **`less`** and **`more`** commands.

**`more`** can be used to view the contents of a file, similar to the cat command, but displaying only one page at a time.  An example of the syntax is:

```
$   more default.prop
```

The space bar takes you to the next page.

NowSecure™

## The less and more commands

**`less`** expands on the **`more`** command in the sense that it allows you to scroll through the file with the up and down arrows, as well as to progress one page at a time with the space bar.  Here is an example of the syntax:

```
$   less default.prop
```

Take a few minutes to try to use the less and more commands in your terminal window.  You can run the command against scripts, text files, or anything that contains human readable text.

Tip:  When running the **`less`** command you may find the **`q`** key to be useful for exiting!

## “Pipeline” files

The pipeline key is located above the enter key and looks like this `|` . This key is used as a hand-off indicator that tells Linux to process the first command and then to process the command that follows the `|` .

This is useful if you would like one command to be sent to another for further processing, such as with the less command.  Here’s an example:

```
$  cat file.txt | less
```

In the above example, the contents of file.txt will be output to the screen one page at a time.

## Find/search files and folders

Similar to Windows, Linux also has a find command that will let you locate various files and folders.  In the GUI interface, you can do a CTRL+f from anywhere and search for a file. Since the primary focus of this training is in the command line environment we'll cover some more powerful search features.

Let's say we want to list out all the files and their path in the users' documents folder.  The following command returns all files and paths, including hidden files.

```
$ find /home/analyst/Documents
```

## Show files and directory path

If we are looking for a specific type of file we can use the following command:

```
$ find . -type f -name "*.jpg"
```

- Find a regular file (**`-type f`**)
- In the current directory (**`.`**)
- Which has a JPG extension (**`-name "*.jpg"`**)

Again, if you are ever not sure on what those switches mean you can type **`find --help`** for quick reference OR **`man find`** for a full manual of the command.

## Create and remove files and folders

Once again, with the GUI you can create folders and files just as in Windows. In all practicality, the GUI is just fine, but since we are focused on the command line interface, we'll touch on some basic commands.

Follow along in your terminal window by first starting out in your home directory (remember how to get to your home directory?) and typing the following commands:

```
mkdir Linux
```

The above command will create a Linux directory inside of your home directory.

## Create and remove files and folders

Now let’s create a couple of directories within the new Linux folder we just created.  You can do this several different ways but the easiest is to use the **`-p`** switch, which creates all the folders in the specified path.

```
mkdir -p Linux/Forensics/Training
```

In this example, both the Forensics folder and the Training folder were created in one command.

## Create and remove files and folders

Now that we have some data in our new directory structure, we will discuss removing directories and files.  Removing folders through the command line is a little different than using the GUI since files deleted through GUI most often go to the trash, whereas command line they are just deleted.

To start, let's delete a folder that has data in it. Start by backing out into your home directory (just type **`cd`** and hit enter):

```
$  rmdir Linux
```

## Create and remove files and folders

To set up for the next exercise, we will need to create some files in the Forensics directory.  Perform the following in your terminal window (note: **`pico`** starts a text editor):

```
cd ~/Linux/Forensics
pico
```

Now type some text in the file and select CTRL+x, type Y when prompted to save, and save it as test1.txt.

Make two more files called test2.txt and test3.txt.

## Create and remove files and folders

Notice that Linux tells you that the directory is not empty and therefore will not delete it. This functionality is in place to protect you from accidentally deleting data, although there is a simple way around it that we will discuss next.

## Create and remove files and folders

You can also delete the entire directory structure with one command, similar to how we created it earlier.  Keep in mind that once again the directories must be empty or this will not work!

```
$  rmdir -p Linux/Forensics/Training
```

The command line returns a message saying that it failed to remove directory ‘Linux/Forensics’ : Directory not empty. Note that the Training folder was deleted because it was empty!

## Create and remove files and folders

Files can be removed the same way that you would remove an empty directory. The **`rm`** command can remove a single file or multiple files at one time, depending on how the command is used. Perform the following from within your Linux/Forensics directory where we created the text files earlier (you have to cd into Linux/Forensics first):

```
$   rm test1.txt  = deletes the specified file
$   rm *.txt  = deletes all .txt files within the current directory
$   rm *  = deletes all files in the current directory
```

Since you already removed all the files with the last command the **`rm *`** command doesn't have anything to do but just know that it will remove everything within the current directory!

## Create and remove files and folders

We have now covered how to delete an individual file or the contents of a folder, but what about the previous example where we tried to delete an entire directory containing active files?

Remember that the system generated an error stating that the directory wasn't empty and we told you that there was a way around it? By placing a **-r** or **-R** after the command, it will recursively remove all the contents, including subfolders.

In our example, we could have typed the following and all the files and subdirectories would have been deleted:

```
rm -r ~/Linux
```

## Create and remove files and folders

So you can see that the **`-r`** switch can really get you in trouble if you are not careful.

There is one last thing to point out about removing files through the command line.  As discussed, the files do not ever hit the trash folder, but they are still recoverable using undelete software or forensic methods. If you desire to remove the files permanently you can use the shred command.

The shred command will not only remove the file but will overwrite it repeatedly, making it difficult if not impossible to recover.

```
$  shred [options] file.ext
```

## Copy and Move Files

Now that you understand how to create and delete files and directories, let’s talk about copying and moving them around.

A copy will simply copy a file from one location into a specified location. Again, you can do this in the GUI by right clicking the file and selecting copy/paste but we are going to show you how to manage this through the command line interface.

## Copy and Move Files

There are several switches that can be used with the **`cp`** command and they are all explained in the command help.  For example, if you run this command:

```
$ cp -p image.dd ~/Desktop/client/case/
```

The file metadata will be preserved as it's copied to the new location as **`-p`** means preserve.

## Copy and Move Files

Linux also allows you to move files through the command line interface. In the example below, we are moving the image.dd file from the current directory to the case folder within client on the desktop.

```
$ mv image.dd ~/Desktop/client/case/
```

## Copy and Move Files

**mv** can also be used to rename a file or folder.  For example, if we run the 12m exploit on a Motorola Android phone, we first need to create a backup of the directory before going any further in the exploit:

```
$ mv /data/local/12m /data/local/12m.bak
```

Specify the directory we want to move

Then specify where we want to move it to

This is the same as right clicking the file in windows and renaming it from 12m to 12m.bak.

## Copy and Move Files

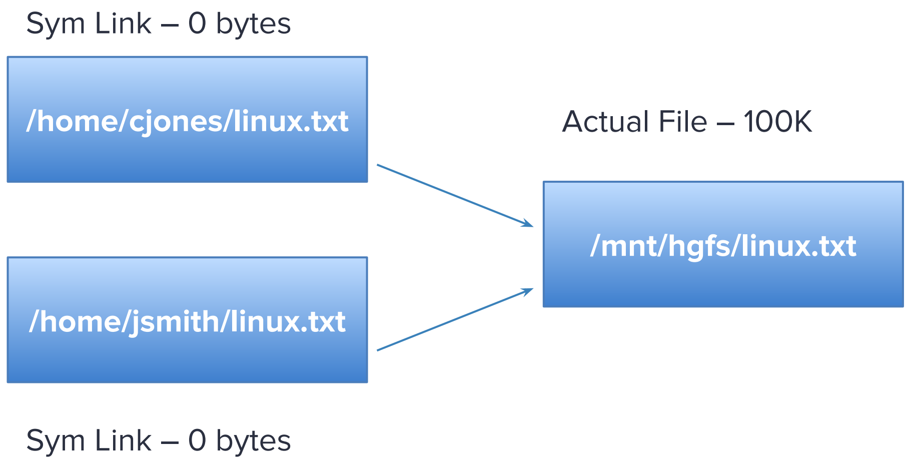
You can see that moving and copying files would be much simpler in the GUI interface, but it’s important to understand command line since we will spend much of our time in a terminal window while doing mobile forensics.

So we have talked about creating, moving, and copying files in the command line interface but what about creating a link to a file?  There are two different types of links, symbolic and hard links, and we will touch on them now.

## Symbolic Links

Symbolic links are files that contain a reference to another file or directory in the form of an absolute or relative path. For example, in Linux we can replace a file with a symbolic link that will point to a different file.

In essence, a symbolic link allows you to have several instances of the same file in different directories that don't take up any space.



## Symbolic

Why would you ever want to do that? There are several reasons, but a practical one that applies to mobile forensics is our Motorola 12m exploit that we touched on in the last slide.

```
$ mv /data/local/12m /data/local/12m.bak
```

NowSecure™

## Symbolic Links

Keep in mind that Android uses a Linux kernel so much of what you learn in this course will directly relate to Android.  In the case of our Motorola 12m exploit, by replacing the original 12m directory with a symbolic link, we can gain write access to a part of the file system that is normally not accessible without “root” privileges.   We first backup the original 12m directory as previously shown with the mv command:

```
$ mv /data/local/12m /data/local/12m.bak
```

Then we create a symbolic link to a different directory:

```
$ ln -s /data /data/local/12m
```

## Symbolic Links

Just to re-emphasize, the command:

```
$ ln -s /data /data/local/12m
```

Writes a symbolic link file that contains the reference /data and will redirect anything that references /data/local/12m to the new location.

Another use of a symbolic link is to store multiple “copies” of the same file that all reference the original file.  When you delete a symbolic link the original file remains intact.

## Symbolic Links: Check Your Knowledge!

1. Which one of the following commands would you type to create a symbolic link to Linux.txt located in the /opt/via folder?

   A. `ln -I /home/analyst/linux.txt /opt/via/Linux.txt`
   B. `ln -s /home/analyst/linux.txt /opt/via/linux.txt`
   C. `ln -s /home/analyst/linux.txt /opt/via/Linux.txt`
   D. None of the above

2. True or false: the two types of links used in Linux are symbolic and virtual.

NowSecure™

## Symbolic Links: CYK! Answer key

1. C
2. False, because the two types are symbolic links and hard links.

Explanations:

Question 1: Line C is the only correct answer because `-I` is not a valid switch and in the second option Linux is not capitalized. Remember, Linux is case sensitive!

Question 2: We didn't cover hard links in detail but here are some basic differences:

1. Hard links cannot be used to link to a directory, whereas symbolic links can.
2. Hard links cannot cross file system boundaries, making them useless in a large scale network environment where multiple file systems exist.

NowSecure™

## Symbolic Links - Test Your Knowledge

2. False, because the two types are symbolic links and hard links.

We didn’t cover hard links in detail but here are some basic differences:

1. Hard links cannot be used to link to a directory, whereas symbolic links can.
2. Hard links cannot cross file system boundaries, making them useless in a large scale network environment where multiple file systems exist.

## Installing Programs

Linux comes standard with several useful forensic tools such as mount, md5sum, dd, xxd, etc.  If a program isn't installed you can download from any source and install or use the software center (some distributions).

For example, Ubuntu has a software center that you can check out here: https://apps.ubuntu.com/cat/applications/software-center/

NowSecure™

## The “which” command

Before installing a program it may be wise to do a quick check to see if it is already installed.  This can be done quickly by using the “which” command.

In your terminal window, type the following commands to see if these programs are installed.

```
$ which fls
$ which hexedit
```

In the examples above, you will see that fls is installed in /usr/local/bin/fls but hexedit is not installed as it returns nothing to the prompt.

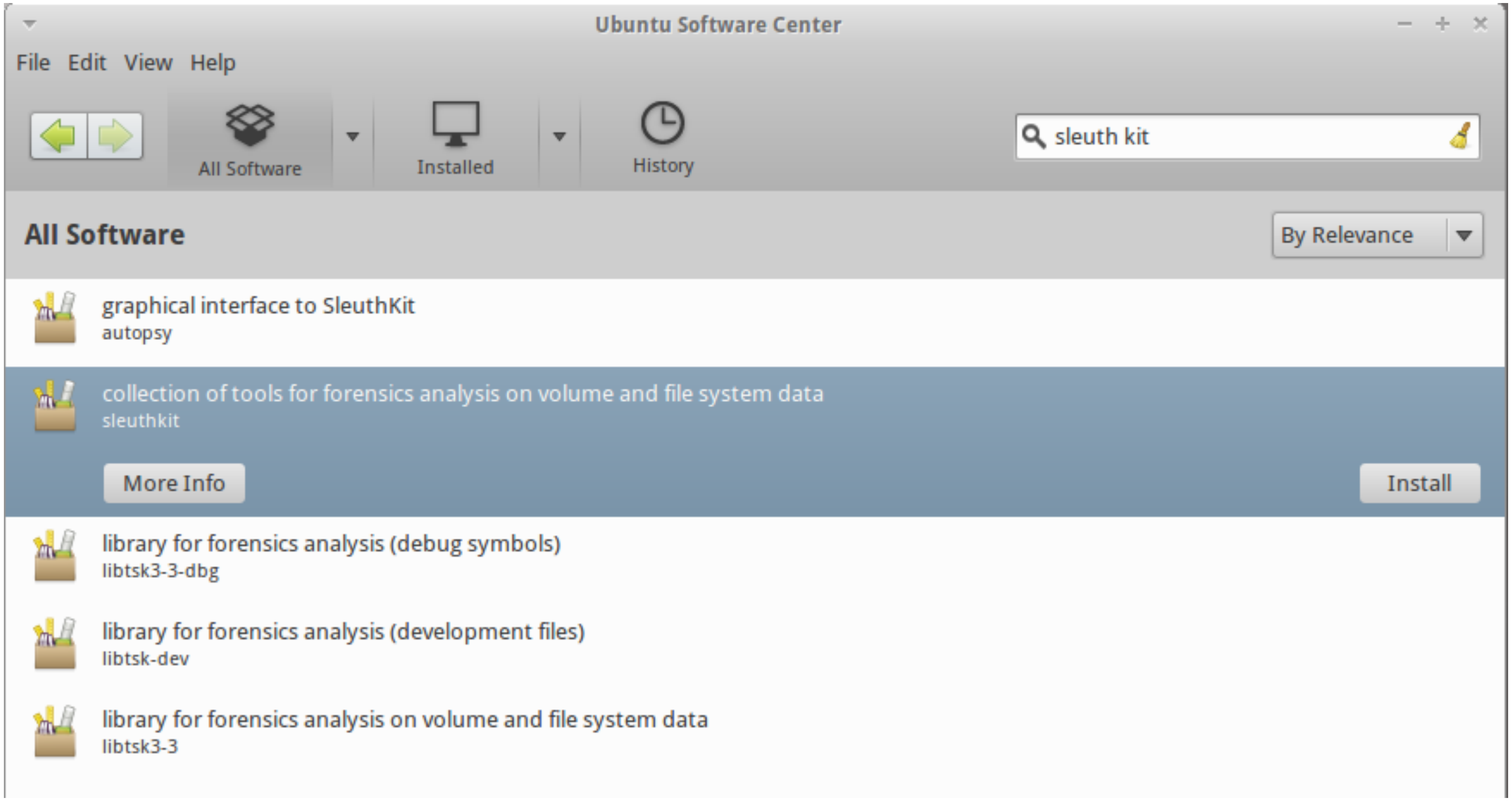
## Installing Programs

Let’s install the Sleuth Kit (TSK) by using the Ubuntu Software Center. First, click on the blue arrow as mentioned previously to open the Software Center. Next, search for “sleuth kit” in the search window.

All of the results will be displayed. See the graphic on the next slide.

# Installing Programs

Click the install button to begin the installation process.



graphical interface to SleuthKit
autopsy

collection of tools for forensics analysis on volume and file system data
sleuthkit

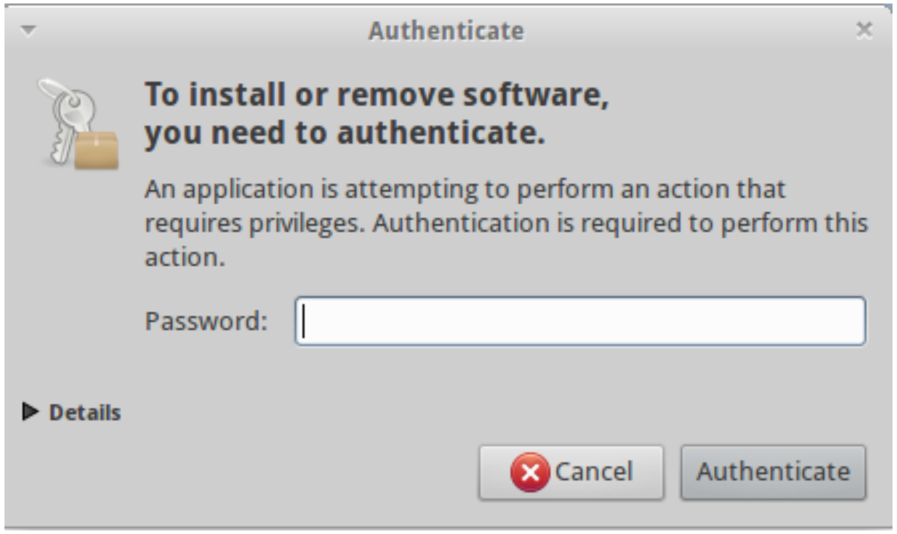
library for forensics analysis (debug symbols)
libtsk3-3-dbg

library for forensics analysis (development files)
libtsk-dev

library for forensics analysis on volume and file system data
libtsk3-3

## Installing Programs

You will need to enter your login password to authenticate the installation. This will be the password you created during the initial VM setup.


Authenticate

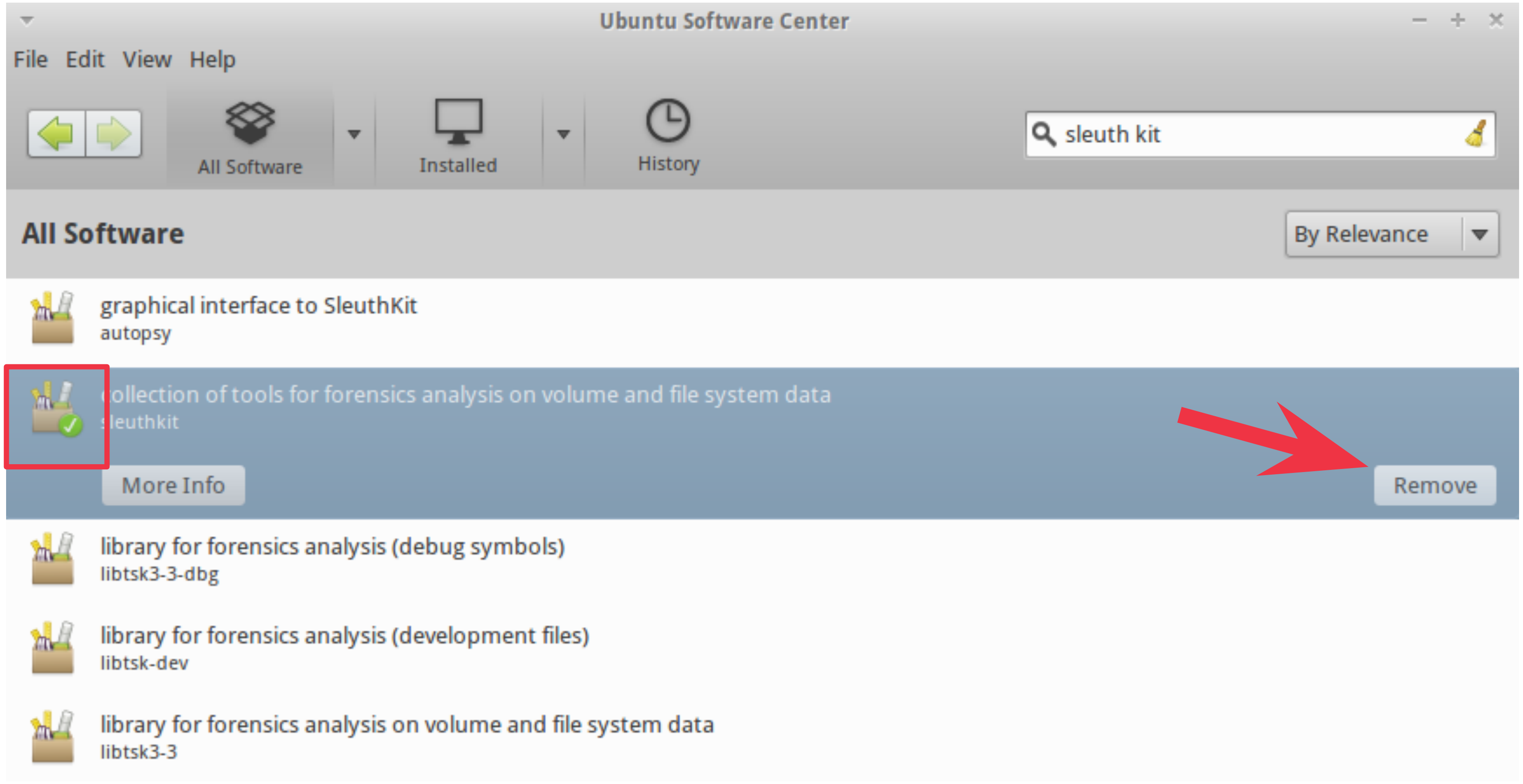
**To install or remove software, you need to authenticate.**

An application is attempting to perform an action that requires privileges. Authentication is required to perform this action.

Password:

## Installing Programs

Once installation is finished there will be a green checkmark next to the package icon and the button will change to “remove”.



graphical interface to SleuthKit
autopsy

collection of tools for forensics analysis on volume and file system data
sleuthkit

library for forensics analysis (debug symbols)
libtsk3-3-dbg

library for forensics analysis (development files)
libtsk-dev

library for forensics analysis on volume and file system data
libtsk3-3

NowSecure™

## Installing Programs

Another method for installing tools is to use apt-get.  This is not as user friendly but it still gets the job done.  Here are some examples of how you would install The Sleuth Kit using the apt-get command:

```
$ sudo apt-get install sleuthkit
```

If you want to uninstall The Sleuth Kit:

```
$ sudo apt-get remove sleuthkit
```

# Installing Programs

Some software has to be downloaded, compiled, then installed. For example, here is how you would install The Sleuth Kit when compiling from source:

1. Download from http://sleuthkit.org/sleuthkit/download.php

```
$ cd Downloads/
$ tar xvf sleuthkit-3.2.3.tar.gz
$ cd sleuthkit-3.2.3/
```

2. Compile the source:

```
$ ./configure
$ make
```

3. Install

```
$ sudo make install
```

NowSecure™

## Installing Programs

Now you can see that using the Software center is so much easier in comparison. Remember, the utility that you need may not be available in the Software Center, so you will then be required to install it from source!

Let's practice installing hexedit, a useful hex editor, from source code.

## Installing Programs

Before we begin this process we will need to make a quick update to our terminal library.  Note that this process is only required to run once.

Type the following into your terminal window:

```
$ sudo apt-get install libncurses5-dev libncursesw5-
dev
```

Be sure not to mistype anything or Linux will generate an error.  Hit enter on the installation prompts to accept the default.

## Installing Programs

1. Using the web browser, download the source from http://rigaux.org/hexedit-1.2.12.src.tgz Then perform the following from your home directory:

```
$ cd Downloads/
$ tar xvf hexedit-1.2.12.src.tgz
$ cd hexedit
```

2. Compile the source:

```
$ ./configure
$ make
```

3. Install

```
$ sudo make install
```

4. Test

```
$ which hexedit
```

NowSecure™

## Installing Programs

To run the application you can either be in the directory that contains the executable and run the program from there, or type the full path to the executable.

When executing a program from the current directory you need to use a `./` in front of the command.

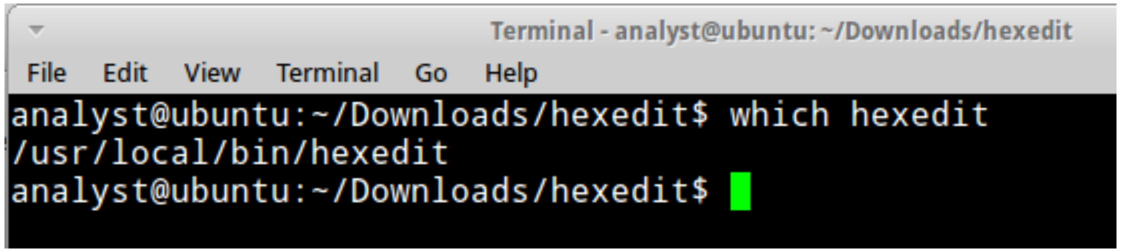
For example, to run mactime you would go to the containing directory and type **`./mactime`** or **`./fls`** to run the fls program. If you are not in the containing directory then you can just type the full path to the executable:

```
/Downloads/sleuthkit-3.2.3/tools/mactime
```

## Installing Programs

However you can also make a program executable from any directory by simply copying the executable into /usr/local/bin.  This feature is similar to Windows when you set the path in the environment variable.

You may have noticed that when you ran the "which" command that all of the utilities so far have been installed to /usr/local/bin by default.


Terminal - analyst@ubuntu: ~/Downloads/hexedit
File Edit View Terminal Go Help

```
analyst@ubuntu:~/Downloads/hexedit$ which hexedit
/usr/local/bin/hexedit
analyst@ubuntu:~/Downloads/hexedit$ 
```

## Installing Programs

If the program did not install to /usr/local/bin by default or it is just a binary utility that you want to be able to execute from anywhere then you can simply copy the utility or executable into the bin directory as shown below:

```
$ sudo cp ~/tools/fastboot  /usr/local/bin
```

## Installing Programs

Once copied you can execute the program by just typing the name of the executable.  No ./ or path required!

Just to recap, here are your options for running a program:

1. From within the containing directory

```
$ ./fastboot …
```

2. From outside the containing directory

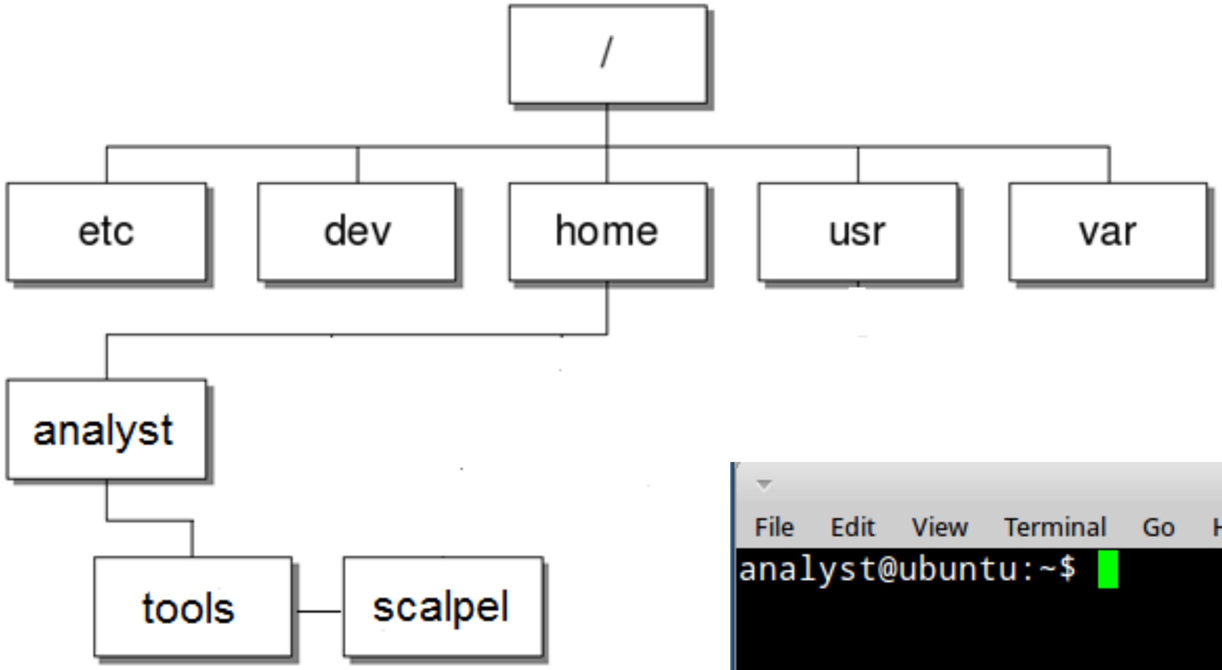
```
$ /<full>/<path>/<to>/fastboot …
```

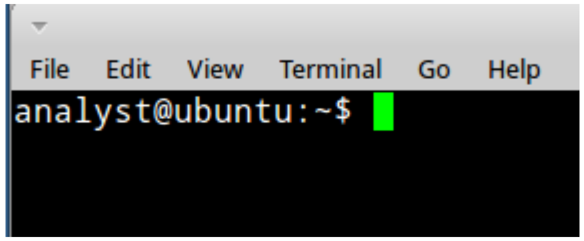
3. If executable is copied to /usr/local/bin

```
$ fastboot …
```

# Installing Programs - Check Your Knowledge

Use these graphics to help you answer the question on the following slide:

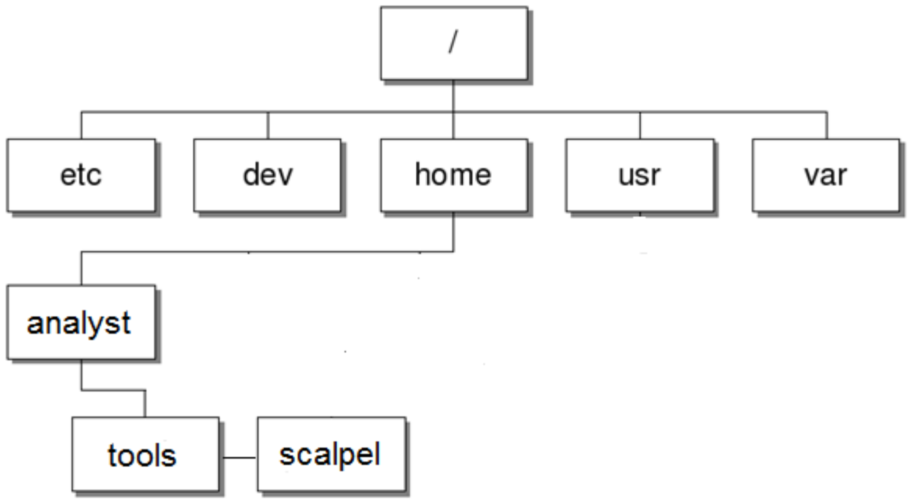


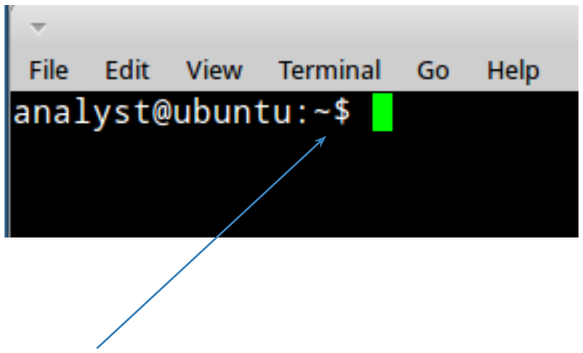


```
analyst@ubuntu:~$
```

# Installing Programs - Check Your Knowledge

Given the following information, what would you type into the command line in order to run scalpel? Note: assume scalpel is loaded in /usr/local/bin





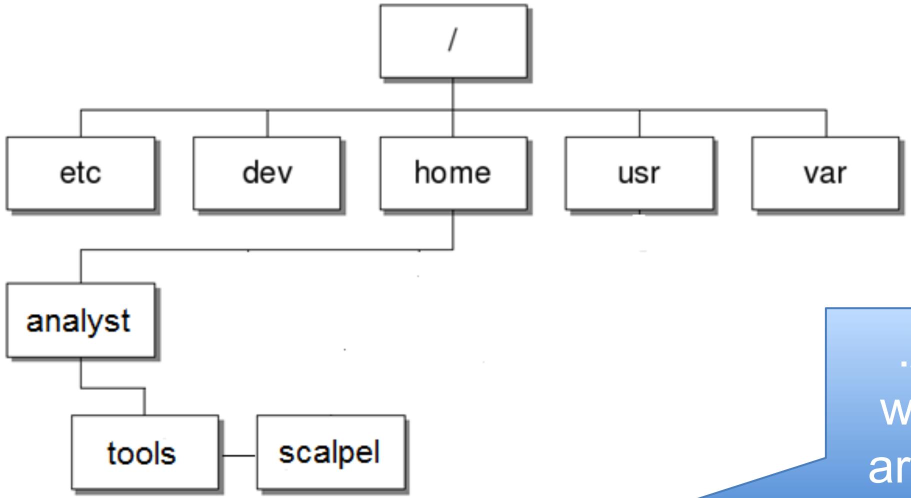
```
analyst@ubuntu:~$ 
```

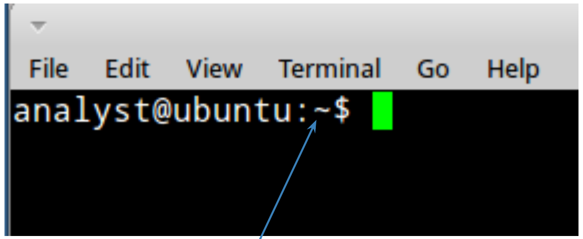
A. **`./scalpel`** …
B. **`scalpel`** …
C. **`tools/scalpel`** …
D. Either **`scalpel`** or **`tools/scalpel`** will work

NowSecure™

# Installing Programs - Check Your Knowledge

Given the following information, what would you type into the command line in order to run scalpel? Note: assume scalpel is loaded in /usr/local/bin





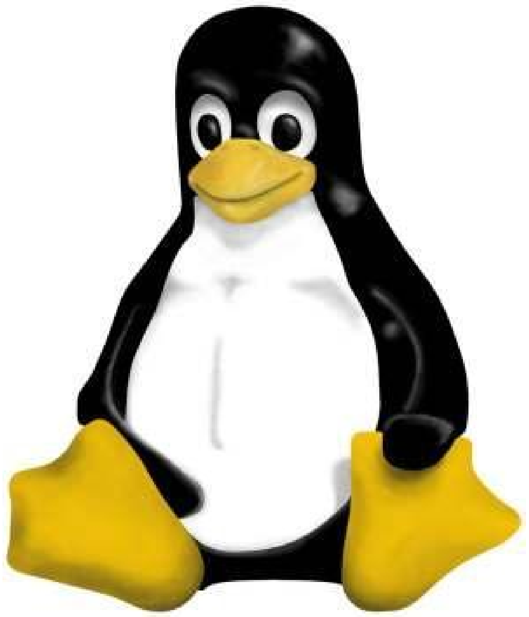
```
analyst@ubuntu:~$
```

./scalpel will not work because we are not in the tools directory!

A. **`./scalpel`** …
B. **`scalpel`** … (If scalpel is loaded in /usr/local/bin)
C. **`tools/scalpel`** …
D. Either **`scalpel`** or **`tools/scalpel`** will work

## Training Outline

1) Linux OS Overview
    a. File system
    b. Basic directory structure
    c. File Permissions
2) Linux Commands
    a. Basics
    b. **Quiz: Linux Commands**
3) Forensic Analysis Tools
    a. Lab: Analysis Part 1
    b. Lab: Analysis Part 2
    c. Lab: Analysis Part 3



## Exam 1: Linux Commands

For the following exam run through the commands on your virtual machine and answer the question with the best possible choice.

## Exam 1: Linux Commands

1. Create a new directory called Linux-course on your desktop. To do this you will perform which of the following?:

    A. Open a terminal window, type **`cd desktop`**, type **`mkdir Linux-course`**
    B. Open a terminal window, type **`cd Desktop`**, type **`mkdir -l Linux-course`**
    C. Open a terminal window, type **`cd desktop`**, type **`mkdir -r Linux-course`**
    D. Open a terminal window, type **`cd Desktop`**, type **`mkdir Linux-course`**

NowSecure™

# Exam 1: Linux Commands

2. Create a new text file that contains the word "test" in the Linux-course directory and name it exam1.txt.  To do this you will perform which of the following commands?:

   A. Type **`cd Linux-course`**, type **`nano`**, type the word "test", hit ctrl-x to exit, Y to save, type exam1.txt as the file name and hit enter
   B. Type **`nano`**, type the word "test", hit ctrl-x to exit, Y to save, type exam1.txt as the file name and hit enter
   C. Type **`cd Linux-course`**, type **`./nano`**, type the word "test", hit ctrl-x to exit, Y to save, type exam1.txt as the file name and hit enter
   D. Type **`./nano`**, type the word "test", hit ctrl-x to exit, Y to save, type exam1.txt as the file name and hit enter

# Exam 1: Linux Commands

3. Change the permissions of exam1.txt to the following:

   Owner – read/write

   Group – read only

   World – read only

To do this, you will type which of the following commands?:

A. `chmod 777 exam1.txt`

B. `chown analyst: exam1.txt`

C. `chmod 455 exam1.txt`

D. `chmod 644 exam1.txt`

NowSecure™

## Exam 1: Linux Commands

4. Change the owner of the exam1.txt file to root. To do this you will type which of the following commands?

   A. `chown root exam1.txt`
   B. `chown root: exam1.txt`
   C. `sudo chown root exam1.txt`
   D. `sudo chown root: exam1.txt`

## Exam 1: Linux Commands

5. Simultaneously change the owner and group of the exam1.txt back to analyst. To do this you will type which of the following commands?

   A. `chown analyst exam1.txt`
   B. `chown analyst: exam1.txt`
   C. `sudo chown analyst exam1.txt`
   D. `sudo chown analyst: exam1.txt`

# Exam 1: Linux Commands

6. Change the group of exam1.txt to root. To do this you will type which of the following commands?

   A. `chown :root exam1.txt`
   B. `chown root: exam1.txt`
   C. `sudo chown :root exam1.txt`
   D. `sudo chown root: exam1.txt`

## Exam 1: Linux Commands

7. Delete the entire Linux-course directory from your desktop, including all of the files within this directory. To do this you will type which of the following commands? (Note: Assume you are currently in the Linux-course folder in your terminal window)

   A. `rmdir Linux-course/`
   B. `rmdir ../Linux-course/`
   C. `rm Linux-course/`
   D. `rm -r ../Linux-course/`

NowSecure™

# Exam 1: Linux Commands

8. Create a symbolic link on your desktop that points to /usr/games/gnome-sudoku. To do this you will type which of the following commands?

   A. `ln -s /usr/games/gnome-sudoku`
   B. `ln -r /usr/games/gnome-sudoku`
   C. `ln -s /Desktop /usr/games/gnome-sudoku`
   D. `ln -s /usr/games/gnome-sudoku ~/Desktop`

# Exam 1: Linux Commands

9. Using the command line, install the note application. To do this you will type which of the following commands?

   A. `sudo apt-get note`
   B. `sudo apt-get install note`
   C. `apt-get install note /applications`
   D. `sudo apt-get install note /applications`

## Exam 1: Linux Commands

10. If there were a document that you wanted to permanently delete what command would you use?

    A. `rm <filename>`
    B. `rmdir <filename>`
    C. `shred <filename>`
    D. `None of the above`

# Exam 1: Linux Commands

11. Using your command line, what would be the fastest way to look at the contents of the file /boot/abi-3.0.0-12-generic?

    A. **`dir /boot/abi-3.0.0-12-genericrmdir`**
    B. **`cat /boot/abi-3.0.0-12-generic`**
    C. **`open /boot/abi-3.0.0-12-generic`**
    D. **`rm /boot/abi-3.0.0-12-generic`**

NowSecure™

# Exam 1: Linux Commands

12. Which of the following commands will view the contents of /boot/abi-3.0.0-12-generic one page at a time?

    A. **`cat /boot/abi-3.0.0-12-generic | more`**
    B. **`cat /boot/abi-3.0.0-12-generic | less`**
    C. **`open /boot/abi-3.0.0-12-generic | less`**
    D. Both a and b

## Exam 1: Linux Commands

13. Which of the following commands will allow you to concatenate the contents of two files into one text file located in your home directory?

    A. `cat /boot/abi-3.0.0-12-generic /boot/abi-3.0.0-15-generic > ~/abi.txt`
    B. `cat /boot/abi-3.0.0-12-generic /boot/abi-3.0.0-12-generic | less`
    C. `open /boot/abi-3.0.0-12-generic /boot/abi-3.0.0-12-generic > /users/<profilename>/abi.txt`
    D. Both a and c

NowSecure™

## Exam 1: Answer Key

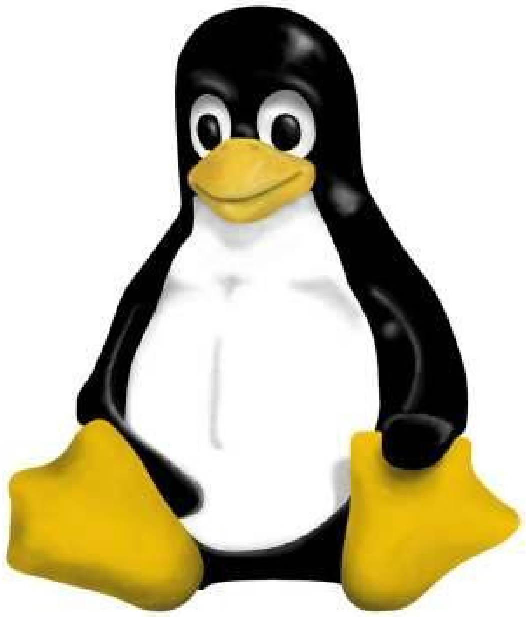
1. D
2. A
3. D
4. C (see more below)
5. D
6. C
7. D
8. D
9. B
10. C
11. B
12. D
13. A

Additional comments for 4: **`Sudo chown root:`** will change the owner and group, but we just want to change the owner.

NowSecure™

## Training Outline

1) Linux OS Overview
   a. File system
   b. Basic directory structure
   c. File Permissions
2) Basic Linux Commands
   a. Quiz: Linux Commands
3) **Forensic Analysis Tools**
   a. Lab: Analysis Part 1
   b. Lab: Analysis Part 2
   c. Lab: Analysis Part 3



NowSecure™

## Overview of Tool Uses

Now that we have covered some commands that are essential to navigating your way through Linux, let's talk about some tools that will be useful in forensics.

## Overview of Tool Uses

Before we continue we need to make a quick update to your virtual machine. Download this file: via.zip and unzip it. Copy it to your Santoku VM's desktop. After that, open a terminal window, then type:

1) `cd /opt`
2) `sudo rm -r via`
3) `mkdir via`
4) `cp ~/Desktop/via/* via/`

That will copy the files from the downloaded folder to the /opt/via folder. To double check, next type

```
ls -lh /opt/via
```

It should show these files:

Linux-101.sh
scalpel.conf
sms.db
viaForensics-Training-Readme

NowSecure™

## Overview of Tool Uses

Many tools are available for Linux that will make your life as a forensic investigator easier, such as:

- Creating a disk image
- Image verification
- Viewing image with Hex editor
- Mounting/Unmounting an image
- Searching across entire disk image
- Data carving
- Timeline creation

## Overview of Tool Uses

In the next section we will explain some of the more common tools and later let you get some practical experience with using the tool in a forensic environment.

## Partitions

The first topic of discussion is partitions. Any electronic media contains partitions (dividing a disk into logical sections) and each partition contains a file system. Some examples of file systems include:

- Windows: NTFS partition
- Linux: ext, ext2, ext3 (among others)
- Mac: HFS or HFS+
- Android: YAFFS2, EXT, RFS
- iPhone: HFS+
- USB/SD Card: Typically FAT, FAT32

NowSecure™

## Viewing Partitions

fdisk (fixed disk) is a Linux utility that can be used to display partitions on a device. To run fdisk against an image file you would type: **`$ fdisk -lu disk-image.dd`**

- "**`-l`**" tells it to list details
- "**`-u`**" tells it to output in sectors

NowSecure™

## Viewing Partitions

mmls (media management listing) is another Linux utility that has similar functions but also displays the unused sectors and works on non-Linux systems. Below are a couple of usage examples of mmls:

```
$ mmls /dev/sdb
$ mmls image.dd
```

Remember that in Linux if you were looking at a partition using mmls that you would need to add the partition number, i.e. sdb1, sdb2, etc.

## mmls - Example

```
username@linux-wks-001:/$ mmls hard-drive-image.dd
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

     Slot    Start        End          Length       Description
00:  Meta    0000000000   0000000000   0000000001   Primary Table (#0)
01:  -----   0000000000   0000000062   0000000063   Unallocated
02:  00:00   0000000063   0000224909   0000224847   Dell Utilities FAT (0xde)
03:  00:01   0000224910   0312576704   0312351795   NTFS (0x07)
04:  -----   0312576705   0312581807   0000005103   Unallocated
```

NowSecure™

## Creating a Disk Image - dd/dc3dd

In forensics obtaining a disk image of a media is an everyday task.  Linux builds in the dd disk imaging utility making life a little easier for the examiner.

Let's take a look at some imaging options.

## Creating a Disk Image - dd/dc3dd

dd or dc3dd can be used to image a hard drive, SD card, or other media. Though a write blocker is always desirable it might not be an option, as in the case of NAND memory.  Again, dd is installed in Linux by default but dc3dd needs to be installed using apt-get or software center.

dd or dc3dd image the device and allow you calculate hash values, create error logs, and specify input and output files/paths.

NowSecure™

# Creating a Disk Image - dd/dc3dd

Here’s how to use dd:

```
$ dd if=/path/to/source of=/path/to/destination

$ dd if=/dev/sdb of=~/Desktop/SD-Card/image.dd
```

The syntax is simple, **`if=`** stands for input file and **`of=`** is the destination image file that you want to create.

dd and dc3dd can image either an entire disk, or just a specific partition. In computer forensics it’s more common to get an entire disk whereas in mobile forensics it is more likely to image a specific partition.

## Verifying a Disk Image - dd/dc3dd

Generating the hash signature of a disk image is an essential part of the imaging process. We can use a simple utility in Linux called md5sum to calculate the hash signature of a disk or partition both before and after the image is created. For example, here is how you get the md5 of a disk before it is imaged:

```
$ md5sum /dev/sdb
```

After the device is imaged, the syntax is the same, but the path will point to the new image file. The hash signatures should match or something was modified during the imaging process!

```
$ md5sum /home/analyst/image.dd
```

## Viewing Image Contents

So you've gone through the process of obtaining a disk image and verifying the hash signature to make sure it is forensically sound. What next?

There are several ways to parse and view the data but for now let's focus on using a simple hex editor to take a quick look at the contents.

Earlier you went through the process of installing hexedit, which is a simple free hex editor that will run great in your Linux command line environment. We will now go over some hexedit essentials so you can find your way around.

## Using hexedit to view hex source

The first step in using hexedit is to launch a terminal window. The program will launch from command line but will present you with a different interface in the terminal window that will allow for some search and scrolling features.

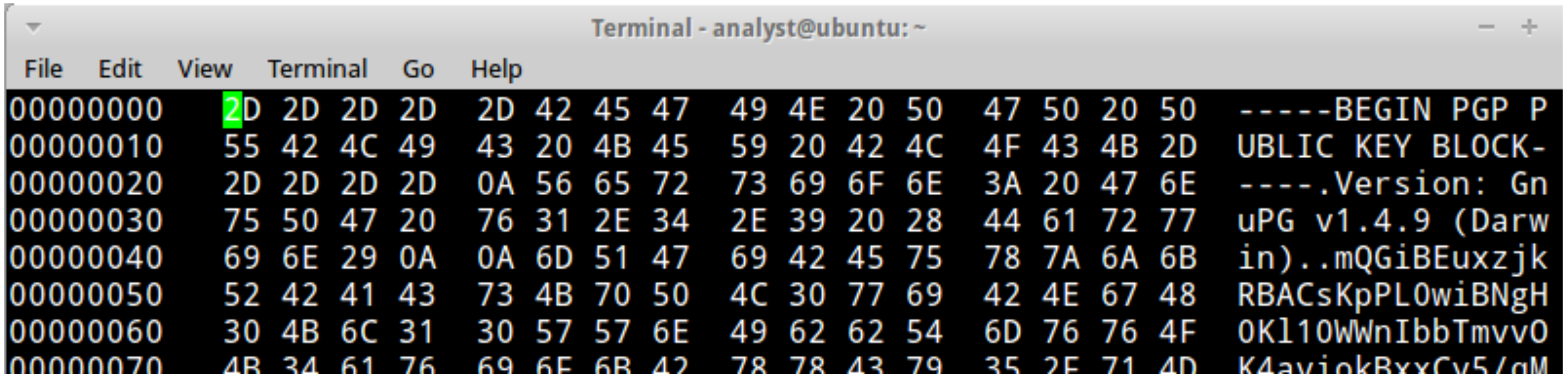
To launch hexedit you simply would type hexedit followed by the path to the image file:

```
$ hexedit /home/analyst/android-userdata.dd
```

You will get some practical experience with this later but for now we just want to cover some basics.

# Using hexedit to view hex source

Hexedit will open and show you the data sequentially in the image file.  You can scroll down through the pages or you can hit the “/” key to bring up the search option.

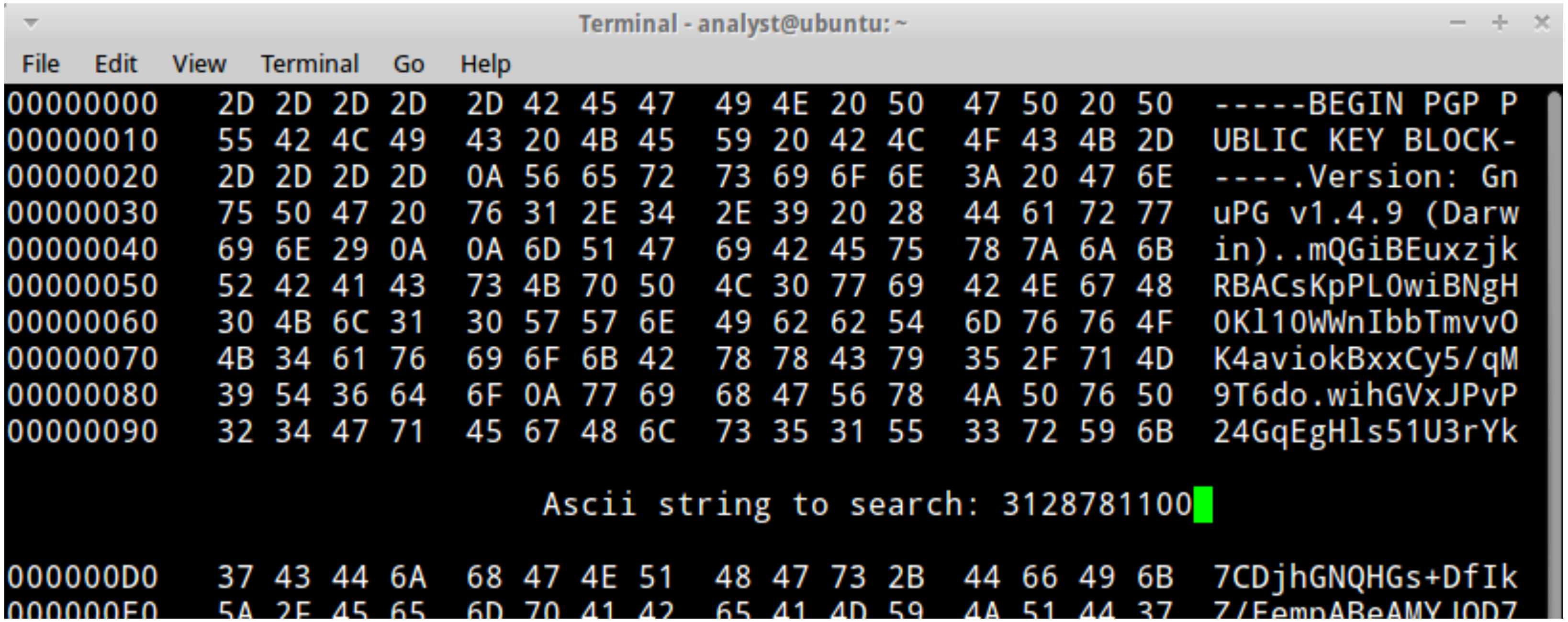


```
00000000    2D 2D 2D 2D  2D 42 45 47  49 4E 20 50  47 50 20 50  -----BEGIN PGP P
00000010    55 42 4C 49  43 20 4B 45  59 20 42 4C  4F 43 4B 2D  UBLIC KEY BLOCK-
00000020    2D 2D 2D 2D  0A 56 65 72  73 69 6F 6E  3A 20 47 6E  ----.Version: Gn
00000030    75 50 47 20  76 31 2E 34  2E 39 20 28  44 61 72 77  uPG v1.4.9 (Darw
00000040    69 6E 29 0A  0A 6D 51 47  69 42 45 75  78 7A 6A 6B  in)..mQGiBEuxzjk
00000050    52 42 41 43  73 4B 70 50  4C 30 77 69  42 4E 67 48  RBACsKpPL0wiBNgH
00000060    30 4B 6C 31  30 57 57 6E  49 62 62 54  6D 76 76 4F  0Kl10WWnIbbTmvvO
00000070    4B 34 61 76  69 6F 6B 42  78 78 43 79  35 2F 71 4D  K4aviokBxxCy5/qM
```

Notice where the green cursor is in the screenshot above.  If we hit the “/” key now, the search will look through hex strings for your search terms.  The TAB key will put the cursor to the ASCII side of things.

# Using hexedit to view hex source

After hitting tab the green cursor moved over to the ASCII side of the display and now we hit the "/" key to begin our search.



```
00000000   2D 2D 2D 2D  2D 42 45 47  49 4E 20 50  47 50 20 50  -----BEGIN PGP P
00000010   55 42 4C 49  43 20 4B 45  59 20 42 4C  4F 43 4B 2D  UBLIC KEY BLOCK-
00000020   2D 2D 2D 2D  0A 56 65 72  73 69 6F 6E  3A 20 47 6E  ----.Version: Gn
00000030   75 50 47 20  76 31 2E 34  2E 39 20 28  44 61 72 77  uPG v1.4.9 (Darw
00000040   69 6E 29 0A  0A 6D 51 47  69 42 45 75  78 7A 6A 6B  in)..mQGiBEuxzjk
00000050   52 42 41 43  73 4B 70 50  4C 30 77 69  42 4E 67 48  RBACsKpPL0wiBNgH
00000060   30 4B 6C 31  30 57 57 6E  49 62 62 54  6D 76 76 4F  0Kl10WWnIbbTmvvO
00000070   4B 34 61 76  69 6F 6B 42  78 78 43 79  35 2F 71 4D  K4aviokBxxCy5/qM
00000080   39 54 36 64  6F 0A 77 69  68 47 56 78  4A 50 76 50  9T6do.wihGVxJPvP
00000090   32 34 47 71  45 67 48 6C  73 35 31 55  33 72 59 6B  24GqEgHls51U3rYk

                          Ascii string to search: 3128781100

000000D0   37 43 44 6A  68 47 4E 51  48 47 73 2B  44 66 49 6B  7CDjhGNQHGs+DfIk
000000E0   5A 2F 45 65  6D 70 41 42  65 41 4D 59  4A 51 44 37  Z/EempABeAMYJQD7
```

In this example, we just search for a phone number, but you can look for anything!

NowSecure™

## Using hexedit to view hex source

To exit out of hexedit you simply hit CTRL+c on your keyboard.  Remember that that key combo will stop any running process in a terminal window.

Some other useful key combo's for hexedit include:

- > = go to end of file
- < = go to beginning of file
- Ctrl-S = search forward
- Ctrl-R = search backward
- Page-down, Page-up moves one page at a time

NowSecure™

## Using hexedit to view hex source

Viewing the image in a hex editor may not be the easiest method of obtaining evidence, but in some cases it is essential since forensic carving tools are limited in the type of data they can pull.

Just remember that a hex editor will only display the raw hex and ASCII data. It's not a very useful way to look at the contents of files contained within the disk image.

NowSecure™

## Manually Mounting a Drive or Image

Mounting the image, on the other hand, is a great way to browse through the logical files in the raw image file.  Keep in mind that a disk image must be mounted as a loopback device, that is the mechanism that is used to interpret files as real devices.

If the image file is of an entire disk it will likely contain multiple partitions.

## Manually Mounting a Drive or Image

Since only certain partitions contain actual user data we need to first determine which partitions we want to mount.  mmls is a great tool that will display all the partition information and give us the details we need to effectively mount the image so we can access the data.

Remember our example mmls output?  Here's what it looked like:

```
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

     Slot    Start        End          Length       Description
00:  Meta    0000000000   0000000000   0000000001   Primary Table (#0)
01:  -----   0000000000   0000000062   0000000063   Unallocated
02:  00:00   0000000063   0000224909   0000224847   Dell Utilities FAT (0xde)
03:  00:01   0000224910   0312576704   0312351795   NTFS (0x07)
04:  -----   0312576705   0312581807   0000005103   Unallocated
```

## Manually Mounting a Drive or Image

```
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

     Slot    Start        End          Length       Description
00:  Meta    0000000000   0000000000   0000000001   Primary Table (#0)
01:  -----   0000000000   0000000062   0000000063   Unallocated
02:  00:00   0000000063   0000224909   0000224847   Dell Utilities FAT (0xde)
03:  00:01   0000224910   0312576704   0312351795   NTFS (0x07)
04:  -----   0312576705   0312581807   0000005103   Unallocated
```

mmls tells you the sector size, which plays an important role because we have to tell mount where the partition starts in order to mount it properly. We do this by calculating an offset.

In this example, we will want to image the “NTFS” partition as it is the largest (note: look at the “length” column in the output). With 512 byte sectors we would multiply the partition start by 512 (224910 x 512) to get an offset of 115153920.

## Mounting an Image

Once you have computed the offset you can proceed with the mount command.  Here are some examples on how to mount various media:

Hard drive:

```
    $ sudo mount -t ntfs -o ro,loop,offset=115153920 drive.dd ~/Desktop/mount
```

SD Card image:

```
    $ sudo mount -t vfat -o ro,loop,offset=32256 sdcard.dd ~/Desktop/mount
```

Notice the ro switch for READ ONLY!  You must also specify a mount location so be sure that you have created a mount directory first, as in this case ~/Desktop/mount.

## Mounting an Image

To mount an external drive (HD, USB, etc.) the process is similar but you must specify the partition by name.  Since the partition name (sdb1 in this example) is defined you would not need to mount as loopback or specify the offset in the command:

```
$ sudo mount -t ntfs -o ro /dev/sdb1 ~/Desktop/mount
```

NowSecure™

## Unmounting a Drive or Image

When you are finished with the data and no longer need the image file mounted, you can simply unmount:

```
$ sudo umount ~/Desktop/mount
```

You can see that mounting an image file in Linux is not that complicated. Certain file systems can make the task slightly more difficult, such as YAFFS2 in Android, but Linux supports many of the basic file systems that you will run across as a forensic examiner.

## Disable Automount

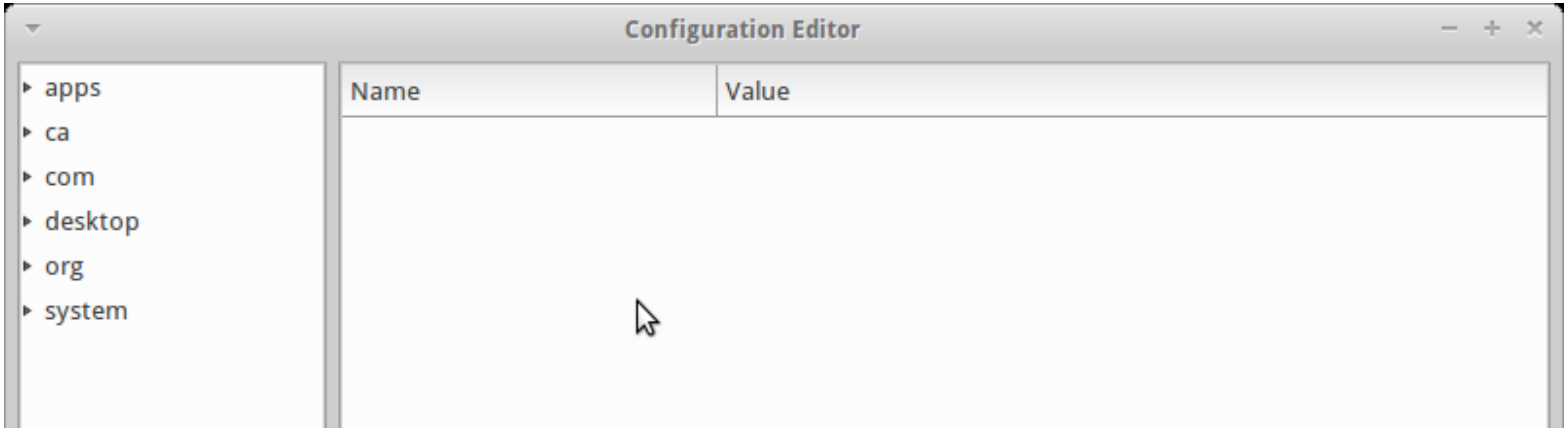
One other thing to mention is that by default any device that is attached to the system will mount automatically. However, there may be many cases where examiners will NOT want external drives to automatically mount on their forensic machine.

Automount can be easily be disabled on Linux to prevent devices from attaching automatically. To access this configuration setting you can either type **`dconf-editor`** in a terminal window and hit enter or go to System and then dconfEditor in the GUI menu.

If the configuration editor is not installed you will have to install it first by using sudo apt-get install dconf-tools or software center.

## Disable Automount

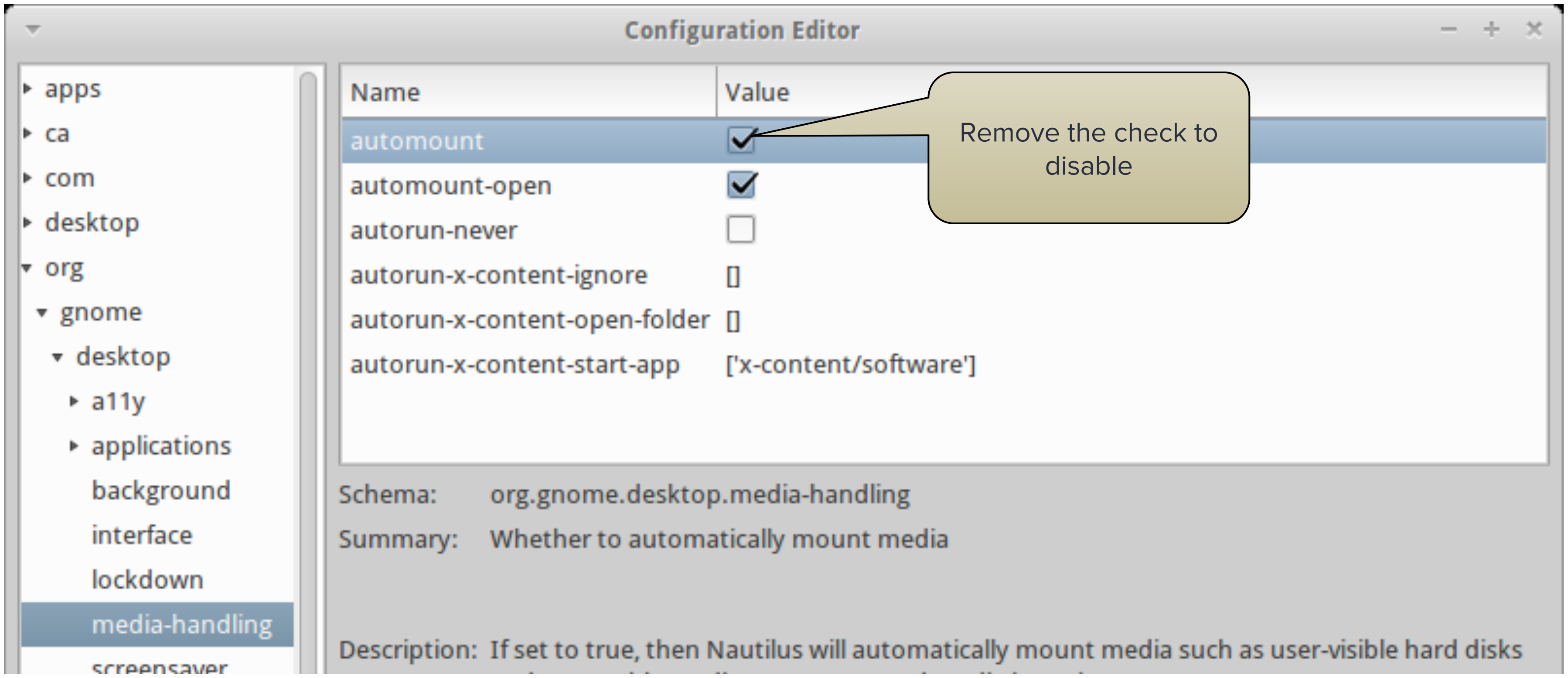
Here is what the configuration editor will look like:



# Linux- Disable Automount

Next, navigate to org > gnome > desktop > media-handling, then uncheck automount. You must now physically mount the media in order to process it.

Note: when doing mobile forensics, you may want to re-enable this option as acquisition tools will require this option to "see" the device



| Name | Value |
|---|---|
| automount | ✓ |
| automount-open | ✓ |
| autorun-never | ☐ |
| autorun-x-content-ignore | [] |
| autorun-x-content-open-folder | [] |
| autorun-x-content-start-app | ['x-content/software'] |

Schema: org.gnome.desktop.media-handling

Summary: Whether to automatically mount media

Description: If set to true, then Nautilus will automatically mount media such as user-visible hard disks

## Forensic Tools in Linux

Obtaining the physical image and mounting it are only part of the overall investigative process. Since we have a full physical disk image, we can explore both the allocated and unallocated space (in most cases) of the disk image. We have to caveat the unallocated space because disk encryption may not allow access.

Next, let's explore some of the great tools that you can use to extract data from the image file.

## “Strings” Command

**`Strings`** is a powerful utility to extract ASCII or Unicode strings from binary data.  That means we can run the utility against the full disk image and extract all of the human readable text into a single text file. Once extracted, you can use your favorite text editor’s search function to look for something specific, or just browse through various sections.

You can run strings against a file or a full disk image.  The next slide will walk you through a few examples of executing the command.

## “Strings” Command

Strings can be run against a full disk image as shown in the first two examples, or against a file as we do in the last command shown here:

```
$ strings Training-image.dd > disk-strings.txt
$ strings iPhone.dmg | less
$ strings sms.db
```

NowSecure™

## “Strings” Command

In this example we output the strings to a new text file called disk-strings.txt:

```
$ strings Training-image.dd > disk-strings.txt
```

You can also output the strings directly to the screen:

```
$ strings sms.db
```

Or view it one page at a time by piping out to the less command:

```
$ strings iPhone.dmg | less
```

# “Strings” Command

Searching through the results can be as easy as opening in a text editor and performing a search using the GUI interface, or you can use grep through the command line. In this example we are searching the nanddump file for “viaForensics” by using grep:

```
$ strings android.nanddump | grep viaForensics
```

Remember that the pipe command tells Linux that there is more than one task to be performed. In this case we first run strings, then we search for viaForensics. Let’s talk a bit more about grep and how to use it.

## Grep Command

As we just demonstrated, grep is a useful tool that will search through a file (or many files/folders) for a specified keyword.  It's a bit more robust than a basic text search because you can add some criteria that will allow you to narrow your search.

First off, grep is case sensitive by default so you need to make sure you use the appropriate case when entering the search parameters.  Here's an example:

```
$ grep viaForensics file.txt
```

You can add the –i switch  to make the command case-insensitive but the search is more time consuming:

```
$ grep -i viaforensics file.txt
```

## Grep Command

Grep will also give you the ability to search for an entire phrase by placing it in quotes:

```
$ grep “Linux Training” file.txt
```

We touched slightly on the recursive switch when we covered deleting files and directories.  Recursive also works with grep as an option to search through all files and folders for a specific keyword.  In the following example we are searching for part of a phone number in any file containing sqlite in the name:

```
$ grep –r 312878 sqlite*
```

NowSecure™

## Grep Command

Grep offers even more with the ability to retrieve lines before and after keyword. For example, we know that the latitude and longitude often appear about 5 lines after any instance of HDTA in an iOS image file. We could set up a grep command to pull out all the latitude and longitude entries from the image file by using the following command:

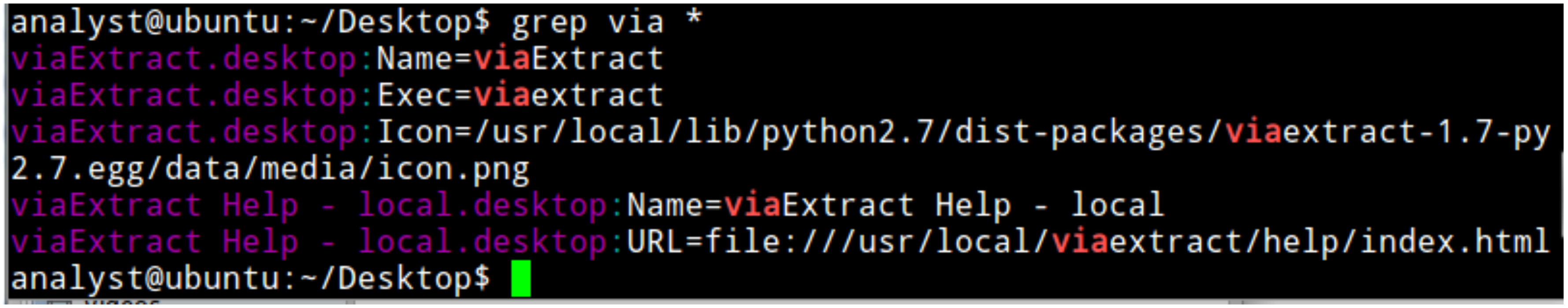
```
$ grep -B 3 -A 8 HDTA iOSimage.img
```

In this example we are grabbing the 3 lines before every instance of HDTA and 8 lines after HDTA giving us a complete picture of the results.

## Grep Command

The final grep feature that we will cover is that it has the ability to search through the entire file system and return the names of the files that contain the specific term you are looking for.

For example, we will grep through all files on user's desktop for "via":



```
analyst@ubuntu:~/Desktop$ grep via *
viaExtract.desktop:Name=viaExtract
viaExtract.desktop:Exec=viaextract
viaExtract.desktop:Icon=/usr/local/lib/python2.7/dist-packages/viaextract-1.7-py
2.7.egg/data/media/icon.png
viaExtract Help - local.desktop:Name=viaExtract Help - local
viaExtract Help - local.desktop:URL=file:///usr/local/viaextract/help/index.html
analyst@ubuntu:~/Desktop$ 
```

NowSecure™

## Grep Command

So you can see that the grep command is very useful and we have only touched on its capabilities.  Take a look at the help file for additional information that will help narrow your searches even more!

NowSecure™

## Data Carving- Scalpel

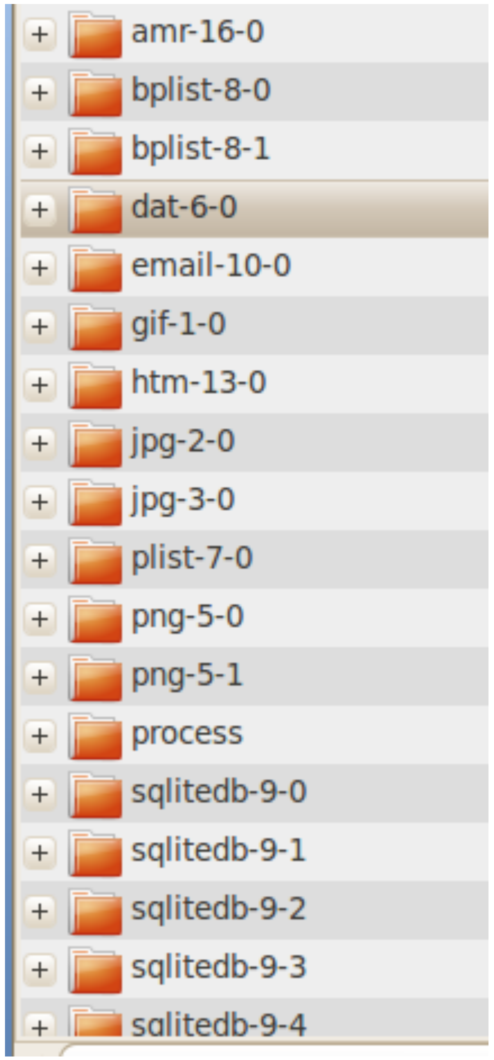
You're almost on the home stretch in terms of putting your newfound skills to the test. But before the lab exercises we need to cover one last important tool: the scalpel.

Scalpel is a useful utility that allows you to carve image files for specific files such as .jpg, .doc, .xls, or virtually any other file. Scalpel is installed as part of the sleuth kit or can be installed separately through the software center or source download as described previously. We have already installed it for you on your virtual machine so it will not be necessary to install it again for the purpose of this course.

# Data Carving- Scalpel

Scalpel is open source and reads a database of header and footer definitions then extracts the defined files from a raw image.

It's file system-independent, so it works on FATx, NTFS, ext2/3, or raw partitions. The extracted files are categorized into folders as shown in the image to the right.



amr-16-0
bplist-8-0
bplist-8-1
dat-6-0
email-10-0
gif-1-0
htm-13-0
jpg-2-0
jpg-3-0
plist-7-0
png-5-0
png-5-1
process
sqlitedb-9-0
sqlitedb-9-1
sqlitedb-9-2
sqlitedb-9-3
sqlitedb-9-4

## Data Carving- Scalpel

Scalpel will extract both logical and unallocated files from the raw image based on the defined header or extension contained in the scalpel configuration file.  Without the configuration file scalpel will not be able to carve anything.

An example of a scalpel.conf is on the next slide.

## Sample Scalpel Configuration

```
#             case      size          header                        footer
#extension    sensitive
     gif       y        5000000       \x47\x49\x46\x38\x37\x61      \x00\x3b
     jpg       y        200000000     \xff\xd8\xff\xe0\x00\x10      \xff\xd9
     png       y        102400        \x50\x4e\x47?                 \xff\xfc\xfd\xfe
     email     y        10240         From:
     doc       y        10000000      \xd0\xcf\x11\xe0\xa1\xb1
#    zip       y        10000000      PK\x03\x04                    \x3c\xac
```

Each entry starts with an extension, defines whether or not it is case sensitive, defines a file size, and defines the header and footer.

In some cases the file may not have a footer in which case the file size will be important as it will start carving at the header and either end at the footer or defined file size.

NowSecure™

## Sample Scalpel Configuration

You can modify the configuration file in any text editor so additional file extensions can be added or others removed.  You can also comment out an entire line by using the # symbol and scalpel will overlook that entry.

```
#    zip      y       10000000        PK\x03\x04               \x3c\xac
```

The trick is that you must have a logical file to examine in a hex editor in order to get the file header/footer information. A good habit would be to grab the header and footer from random files when you have the opportunity since a future investigation might not have any logical files available for the file type you are looking for!

## Scalpel- Process

To run scalpel you need to first make sure it is installed in /usr/local/bin. Remember that any file in /usr/local/bin can be executed from anywhere? Take a look to see where scalpel is installed on your virtual machine by typing:

```
$ which scalpel
```

The terminal will return the path for you to confirm that it is in /usr/local/bin.

NowSecure™

## Scalpel- Process

The syntax for scalpel is very simple.  You must point the executable to the path of the configuration file, in the example below the configuration file is in /opt/via/scalpel.

After defining the configuration file you simply specify the path to the raw image, as in our example ~/Desktop/lab1/image.dd

```
    $ scalpel -c /opt/via/scalpel.conf ~/Desktop/lab1/image.dd
```

## Scalpel- Process

Examine data in the “scalpel-output” directory, which is created in the folder where scalpel was executed from.  In this example the scalpel-output directory would be located in /opt/via/scalpel-output because we were in /opt/via at the time of execution as shown below:

```
analyst:/opt/via$ scalpel –c /opt/via/scalpel.conf ~/Desktop/image.dd
```

One other important factor to note about the scalpel-output directory is that once it’s created scalpel will not run in the same directory until the folder is renamed or deleted.  This is a built-in feature that prevents you from overwriting scalpel-output by accidentally running scalpel again against a different raw image file.

# Scalpel- Screen Output

```
analyst@ubuntu:~$ scalpel -c ~/scalpel.conf /mnt/ntfs/hard-drive.dd
Scalpel version 2.0
Written by Golden G. Richard III and Lodovico Marziale.
Multi-core CPU threading model enabled.
Initializing thread group data structures.
Creating threads…
Thread creation completed.

Opening target "/mnt/ntfs/hard-drive.dd"

Image file pass 1/2.
/mnt/ntfs/hard-drive.dd: 100.0% |**********************************|    7.1 GB    00:00 Allocating work
    queues...
Work queues allocation complete. Building work queues...
Work queues built.  Workload:
gif with header "\x47\x49\x46\x38\x37\x61" and footer "\x00\x3b" --> 0 files
gif with header "\x47\x49\x46\x38\x39\x61" and footer "\x00\x3b" --> 15 files
jpg with header "\xff\xd8\xff\xe0\x00\x10" and footer "\xff\xd9" --> 303 files
jpg with header "\xff\xd8\xff\xe1" and footer "\x7f\xff\xd9" --> 107 files
<output omitted>
** PREVIEW MODE: GENERATING AUDIT LOG ONLY **
** NO CARVED FILES WILL BE WRITTEN **
Carving files from image.
Image file pass 2/2.
/mnt/ntfs/hard-drive.dd: 100.0% |**********************|    7.1 GB    00:00 ETA Processing of image file
    complete.
Cleaning up...
Done.
Scalpel is done, files carved = 2400, elapsed  = 357 secs.
```

NowSecure™

## Scalpel- Summary

Scalpel is a powerful carving tool and can be used to carve any type of file as long as the header or file extension is known.  We talk more about scalpel in our Advanced File Carving course where we spend a good amount of time modifying the configuration file in an attempt to recover various deleted items from an Android phone.  For now, we have introduced you to the basics so you can start carving away on your next forensic case!

Let's move on to creating timelines!

## Creating a Timeline- The Sleuth Kit (TSK)

Many files and directories within a raw image have times associated with them. A timeline is essential as it will provide a high-level look at system activity, such as when files were compiled and when archives were opened.

Running a timeline is a two-step process:

1. Using the “fls” tool, data is gathered from sources such as file systems, registries, logs, etc. and saved to a the “body file” format
2. The “mactime” script is then used to sort and merge this data into a timeline

## Download, Compile and Install TSK

The timeline utilities are all part of The Sleuth Kit, which we installed earlier. Remember that the easiest way to install software is through the software center GUI interface!

Since everything is already installed we can move on and introduce you to the first part of the process, fls.  We already talked about what fls does so let's take a look at how to run the utility.

## Creating a Timeline- TSK

```
$ fls -o 63 -z CST6CDT -s 0 -m 'C:/' -f ntfs -r ~/Desktop/hard-
   drive.dd > ~/Desktop/harddrive-timeline.body
```

- -o 63 = offset, or starting sector of the partition
- -z CST6CDT = central time zone -6 hours from Greenwich Mean Time
- -s 0 = no time skew, no minutes added or subtracted to the times (used to fix a clock that is "off")
- -m 'C:/' = all paths to files will begin with C:/
- -f ntfs = the file system type will be NTFS
- -r = recursively follows directories
- hard-drive.dd = disk image name
- > ~/Desktop/harddrive-timeline.body = direct the output to a body file on the Desktop of the current user

Note: In the command above, there is a single space between **`-r`** and **`~/Desktop`**

NowSecure™

## Creating a Timeline- TSK

Once you have the body file from the fls output you will need to run the mactime script to format and merge into a timeline:

```
$ mactime -b ~/Desktop/harddrive-timeline.body -z
  CST6CDT -d > ~/Desktop/harddrive-timeline.csv
```

- -b = specify body file
- -z CST6CDT = timezone
- -d = delimited (i.e. can open in Excel)
- > **`~/Desktop/harddrive-timeline.csv`** = direct the output to a csv file named “harddrive-timeline.csv” on the Desktop of the current user

Note: In the command above, there is a single space between **`-z`** and **`CST6CDT`**

## Creating a Timeline- log2timeline

Another useful timeline tool is log2timeline. This handy utility will allow you to create a body file similar to fls, but in this case you can target specific files instead of having to run it on the entire raw image.  For example, you may want to create a timeline on the recycle bin or a Firefox SQLite web history file and log2timeline will accomplish it with ease.

To see all the file types that log2timeline supports run the following command:

```
$ log2timeline -f list
```

## Creating a Timeline- log2timeline

Here are some examples on how to run log2timeline:

```
$ log2timeline -f recycler -o sqlite -w /tmp/rec.sql INFO2
$ log2timeline -f prefetch WINDOWS/Prefetch >> case.body
```

Once the body file is created you can import it into other tools to generate the actual timeline, such as mactime.

## Analyzing a Timeline

Capturing the timeline is probably the easiest step of the entire process since interpreting the data can be daunting at times. It is best to focus on a specific timeframe and narrow your search from there.

When a file is created the modified/accessed/changed/birth times are initially set.  Here are the definitions of each of these MACB entries:

- m: modified (metadata modified about the file)
- a: accessed (file itself has been accessed)
- c: changed (content of the file has been changed)
- b: birth (file created)

## Analyzing a Timeline

Keep in mind that normal system routines will access files frequently, such as a virus scan or system recovery checkpoint.  However, if content is modified there will be a “c” so that will help narrow it down a bit.

When you see a file of interest in the timeline, you can track it down in the mounted image.  The full path is displayed in the timeline so locating it in the image file will be simple in most cases.  For example, you see a picture called “IMG_001.jpg” was created at a certain time.  Go to the image file to locate the photo and see what the picture contains.

## Analyzing a Timeline

Another example is if you want to determine if files were transferred to a USB drive. You can compare the USB connection times with the timeline to see what files were accessed to help make this determination.

## Analyzing a Timeline

As you can see, a timeline is essential to gain a thorough understanding of what transpired on a device during a certain time period. Though the search can be time consuming, the information gained can make or break the case.

You are now on your way to getting some hands-on experience with all of these new commands in the lab portion of this training.  But first, let's take a short quiz to see how much you've learned!

Keep in mind the commands in these questions will not work on your VM since the paths are different.  This quiz is intended to get you thinking about each command logically rather than just testing them until you find one that works!

## Forensic Tools: Check Your Knowledge!

1. What tool can be used to display partition information?

   A. Scalpel
   B. Fls
   C. Fdisk
   D. mmls
   E. Both C and D

## Forensic Tools: Check Your Knowledge!

2. Creating a disk image can be done using dd or dc3dd. The nice part about using dc3dd is that it incorporates a built-in write blocker. True or false?

## Forensic Tools: Check Your Knowledge!

3. You have successfully extracted the physical hard drive from the computer and now you are ready to image the disk. You plug the device into a write blocker and open a terminal window in Linux. After running ls –l on /dev you see that the external disk is mounted as sdb2. What would be the correct syntax for performing a dd image of the hard drive and saving to your home directory?

    A. `dd of=/home/analyst/disk-image.dd if=/sdb2`
    B. `dd if=/dev/sdb2 of=/dev/home/disk-image.dd`
    C. `dd if=/etc/sdb2 of=/home/analyst/disk-image.dd`
    D. `dd if=/dev/sdb2 of=~/disk-image.dd`

## Forensic Tools: Check Your Knowledge!

4. How would you verify the forensic integrity of the image?

   A. `sudo md5sum /dev/sdb2` before imaging
   `sudo md5sum ~/disk-image.dd` after imaging

   B. `sudo hashsum /dev/sdb2` before imaging
   `sudo hashsum ~/disk-image.dd` after imaging

   C. `sudo md5 /dev/sdb2` before imaging
   `sudo md5 ~/disk-image.dd` after imaging

   D. `sudo checksum /dev/sdb2` before imaging
   `sudo checksum ~/disk-image.dd` after imaging

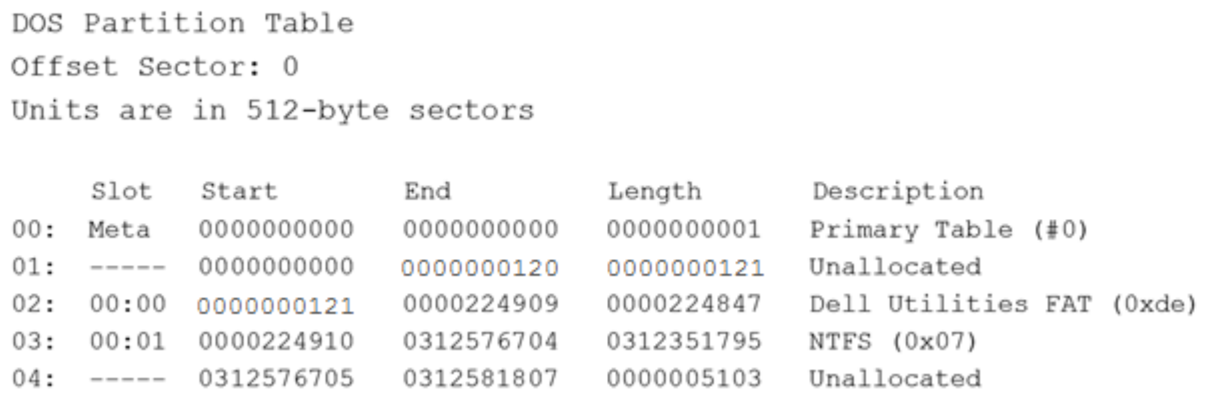
## Forensic Tools: Check Your Knowledge!

5. You fired up your favorite hex editor, hexedit, and are about to begin the tedious task of locating a text message from a specific phone number 312-878-1100. How would you search for this number in hexedit?

   A. Ctrl+f on the keyboard and type the phone number
   B. "/" on the keyboard and type the phone number
   C. Tab on the keyboard and type the phone number
   D. Tab, then "/", then type the phone number

# Forensic Tools: Check Your Knowledge!

6. Looking at the mmls output below, choose the correct command for mounting the image file, specifically the “NTFS” partition:



```
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

     Slot    Start        End          Length       Description
00:  Meta    0000000000   0000000000   0000000001   Primary Table (#0)
01:  -----   0000000000   0000000120   0000000121   Unallocated
02:  00:00   0000000121   0000224909   0000224847   Dell Utilities FAT (0xde)
03:  00:01   0000224910   0312576704   0312351795   NTFS (0x07)
04:  -----   0312576705   0312581807   0000005103   Unallocated
```

A. `$ sudo mount -t ntfs -o ro,loop,offset=32256 drive.dd ~/Desktop/mount`
B. `$ sudo mount -t ntfs -o ro,loop,offset=115153920 drive.dd ~/Desktop/mount`
C. `$ sudo mount -t ntfs -o ro,loop,offset=61440 drive.dd ~/Desktop/mount`
D. `$ sudo mount -t ntfs -o ro,loop,offset=512 drive.dd ~/Desktop/mount`

NowSecure™

## Forensic Tools: Check Your Knowledge!

7. You are investigating an SD card and are about to mount it in Linux to take a look at the logical files. What is wrong with this mount command from a forensic standpoint?

```
$ sudo mount -t vfat -o rw,loop,offset=32256 sdcard.dd ~/Desktop/mount
```

A. The mount command does not require sudo
B. This mount command would mount the partition with write access
C. The file system type should be fat32
D. There is nothing wrong with this mount command from a forensic standpoint.

NowSecure™

## Forensic Tools: Check Your Knowledge!

8. You performed a physical image of an SD card out of a Samsung Galaxy S. You mounted the image and pulled out all the logical files but you also want to see if there are any deleted pictures in the unallocated space. You remember that scalpel is a great tool for carving files so you decide to give it a try. You are currently located in opt/via, which is where your configuration file and disk image are located. "Scalpel" has already been pre-installed and is located in /usr/bin. What command would you type to run scalpel?

   A. `$ ./scalpel -c scalpel.conf disk-image.dd`
   B. `$ scalpel -c /opt/via/scalpel.conf ~/disk-image.dd`
   C. `$ scalpel -c scalpel.conf disk-image.dd`
   D. `$ ./scalpel -conf scalpel.conf /disk-image.dd`

## Forensic Tools: Check Your Knowledge!

9. Where did your previous scalpel command output the carved data?

   A. /opt/via/scalpel-data
   B. /opt/via/scalpel-output
   C. /home/analyst
   D. ~/

NowSecure™

## Forensic Tools: Check Your Knowledge!

10. You decided to perform a strings search on the entire image file so  you can find more information relating to that text message from earlier. What command would you run to get a strings output file from the image file?

    A. `$ strings /opt/via/disk-image.dd | disk-strings.txt`
    B. `$ strings /opt/via/disk-image.dd disk-strings.txt`
    C. `$ strings /opt/via/disk-image.dd > disk-strings.txt`
    D. `$ ./strings /opt/via/disk-image.dd | disk-strings.txt`

## Forensic Tools: Check Your Knowledge!

11. You want to run a search against the entire dd image for viaForensics and viaforensics. Since grep is case sensitive, what switch could you add to search for uppercase and lowercase all in one command?

    A. `grep –c viaforensics ~/disk-image.dd`
    B. `grep -i viaforensics ~/disk-image.dd`
    C. `grep –LU viaforensics ~/disk-image.dd`
    D. `grep –r viaforensics ~/disk-image.dd`

## Forensic Tools: Check Your Knowledge!

12. How do you perform a grep search to grab 3 lines before your search term and 8 lines after?

    A. `grep -B 3 -A 8 viaforensics ~/disk-image.dd`
    B. `grep -3 +8 viaforensics ~/disk-image.dd`
    C. `grep -b 3 -a 8 viaforensics ~/disk-image.dd`
    D. **Either A or C will work**

## Forensic Tools: Check Your Knowledge!

13. Timelines are a useful forensic tool and are often run on an entire disk image. What timeline tool could you use if you only wanted a timeline on the Windows recycle bin?

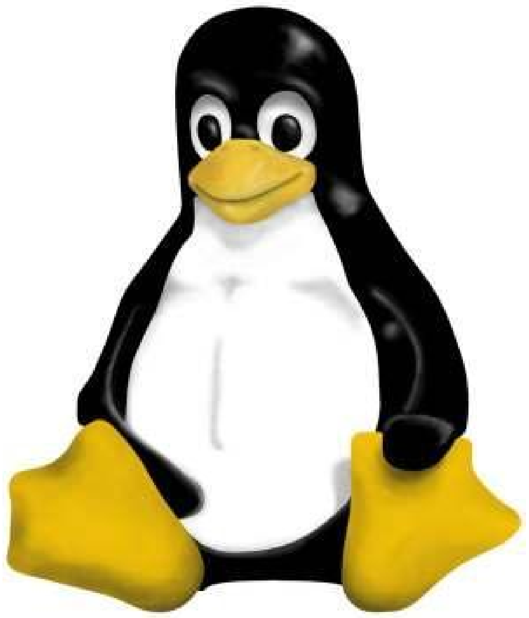
    A. fls
    B. log2timeline
    C. Neither

# Forensic Tools: CYK! Answer Key

1. E
2. false
3. D
4. A
5. D
6. B
7. B
8. C
9. B
10. C
11. B
12. A
13. B

## Training Outline

1) Linux OS Overview
    a. File system
    b. Basic directory structure
    c. File Permissions
2) Basic Linux Commands
    a. Quiz: Linux Commands
3) Forensic Analysis Tools
    a. **Lab: Analysis Part 1**
    b. Lab: Analysis Part 2
    c. Lab: Analysis Part 3



NowSecure™

## Lab: Forensic Analysis, Part 1

To run the forensics part of this training, you first need to install some tools on your Santoku VM.

Type in this command to install these tools:

```
sudo apt-get install dconf-tools
type in your sudo password, and hit enter
```

NowSecure™

# Lab: Forensic Analysis, Part 1

Connect a USB stick to your virtual machine using a write blocker (if available). If you don't have a write blocker you could disable auto-mount in Linux and manually mount the USB drive as read only.

To disable auto-mount type **`dconf-editor`** in a terminal window. Next go to org, gnome, desktop, media-handling, and remove the checkmarks next to auto mount and automount-open.

To manually mount the image file you must first go into the disk utility (under Applications/Preferences) to see how Linux defines the external media. In our case it is /dev/sdc but it may be different on your system.

To mount the external drive as read only:

```
$ mkdir ~/Desktop/mnt
$ sudo mount -t vfat -o ro /dev/sdc1 ~/Desktop/mnt
```

NowSecure™

## Lab: Forensic Analysis, Part 1

When we discussed disk imaging we brought up the importance of performing a md5 hash against the disk image to verify forensic integrity.  If the disk hash doesn't match the image hash we know that something changed during the process.

1. Obtain a hash signature of the image you downloaded earlier in the training.

You first need to determine where Linux stored the downloaded image.  You can use the disk utility to determine this.  Next, perform the md5sum in your terminal window:

```
$ sudo md5sum linux-training-sdcard-image.dd
```

NowSecure™

## Lab: Forensic Analysis, Part 1

Once we have the hash signature we can get a physical image file.

2. Create a dd image of the file and save it to /Desktop/lab1. (You may have to create this folder)

```
$ sudo dd if=~/Desktop/linux-training-sdcard-image.dd
of=~/Desktop/lab1/sdimage.dd
```

Note that you will need to replace /Desktop/linux with the actual download point of your image… as you determined in the previous step.

NowSecure™

# Lab: Forensic Analysis, Part 1

Next we want to mount the partition from the image we just created. But first we need to determine a few things:

3. Determine partition type and sector

   What is the partition type? ________

   What is the starting sector? _______

   What is the offset? _______

Run $ `mmls ~/Desktop/lab1/sdimage.dd`
Determine partition type (NTFS, FAT16, etc)
Locate starting sector for that partition
Multiple starting sector by 512 to determine offset.

## Lab: Forensic Analysis, Part 1

Now that we know the offset we can mount the partition to browse the logical files.

4. Mount the above partition as read-only

First you need to create a mount folder. Remember that the mount command requires a location to mount the image so create that first. It can be created anywhere and named anything you wish. In this example we created a folder on our desktop called mount.

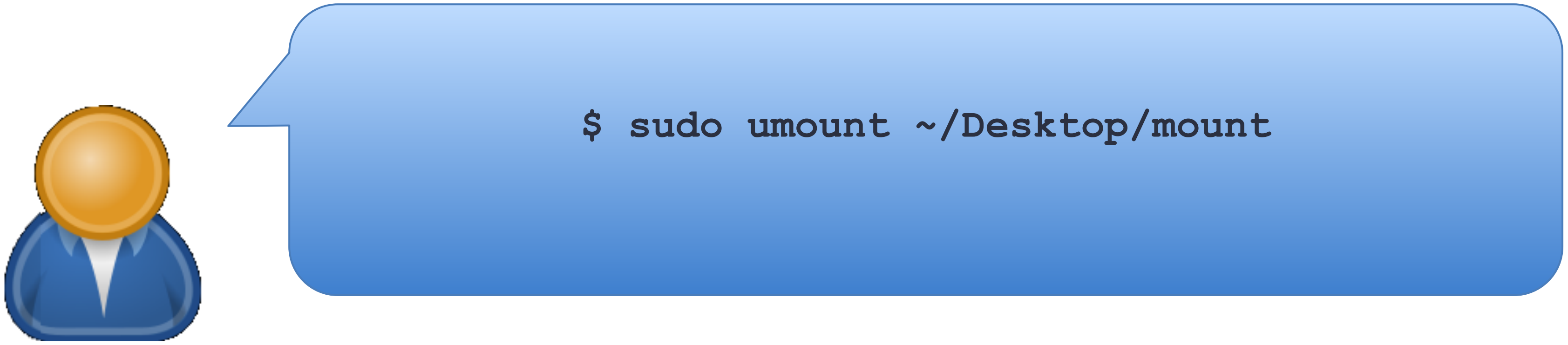
Once the folder is created run the mount command using the offset calculated previously. Also, if file system was FAT16 or FAT32, you can use "vfat" os the file system type.

```
$ sudo mount -t vfat -o ro,loop,offset=51712
~/Desktop/lab1/sdimage.dd ~/Desktop/mount
```

# Lab: Forensic Analysis, Part 1

After we are finished with the image we need to unmount it from Linux.

5. Since we are not going to process any of the logical files in this lab, unmount the image from Linux



```
$ sudo umount ~/Desktop/mount
```

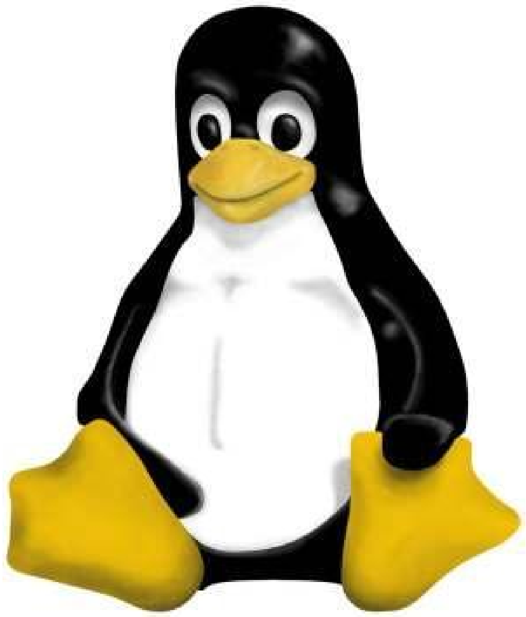
## Lab: Forensic Analysis, Part 1

Congratulations!  You have completed the imaging portion of the Linux lab. If you are not comfortable with imaging and mounting disk images please review the Forensic Analysis Tools section in this training and run through the lab again.

You may not feel like a Linux expert, but you should feel comfortable with the material before moving on.

## Training Outline

1) Linux OS Overview
    a. File system
    b. Basic directory structure
    c. File Permissions
2) Basic Linux Commands
    a. Quiz: Linux Commands
3) Forensic Analysis Tools
    a. Lab: Analysis Part 1
    b. **Lab: Analysis Part 2**
    c. Lab: Analysis Part 3



NowSecure™

## Lab: Forensic Analysis, Part 2

So far you have imaged the SD card and mounted it to view the logical file system. Now we want to carve all the files contained within the image file.

1. Run Scalpel against the image file to carve the files. You can use the default Scalpel configuration file located in /opt/via.

Tip: create a directory at ~/Desktop/lab1/scalpel and use the –o switch to direct your output to that location. Click the play button below to see the command!

```
$ mkdir ~/Desktop/lab1/scalpel
$ scalpel -c /opt/via/scalpel.conf
~/Desktop/lab1/sdimage.dd -o ~/Desktop/lab1/scalpel/
```

NowSecure™

## Lab: Forensic Analysis, Part 2

Your Scalpel output from the previous step should be located in ~/Desktop/lab1/scalpel.  Take a few minutes to look through some of the folders to help gain familiarization with the type of data that is extracted and the way that it is organized.

Remember that if you were to run scalpel again you would need to rename the scalpel-output directory or direct the output to another location since Scalpel is designed to not overwrite the scalpel-output directory.

## Lab: Forensic Analysis, Part 2

To get an idea of what transpired we need to generate a timeline from our image file.

2. Using fls and mactime, create a timeline of the SD card image file.

```
$ mmls ~/Desktop/lab1/sdimage.dd  (To find offset)
$ fls -o 101 -z CST6CDT -s 0 -m ‘\’ -f fat16 -r
~/Desktop/lab1/sdimage.dd > ~/Desktop/lab1/sdcard.body

$ mactime -b ~/Desktop/lab1/sdcard.body -z CST6CDT >
~/Desktop/lab1/timeline.csv
```

# Lab: Forensic Analysis, Part 2

3. Based on the information contained in the timeline, what happened on February 17 around 3:15 PM?

Around this time, a file called "Company-Secrets.txt" was created on the SD Card at 3:15. This file was then deleted at 3: 18.

## Lab: Forensic Analysis, Part 2

On February 14, between 5:00 and 5:05 PM, a file was deleted. Find the name of this file and try to recover it.

The file iphonehand.bmp was deleted exactly at 5:03:10.

## Lab: Forensic Analysis, Part 2

1. What do you think it means when a file ends in (deleted)? Do you think this file is recoverable?

   A. The file is deleted and is not recoverable
   B. The file is deleted but still resides in unallocated space and can possibly be recovered through file carving
   C. The file could be recoverable unless it is appended with “deleted-realloc”
   D. Both B and C

## Lab: Forensic Analysis, Part 2

2. Examine 3 of the files that were accessed on February 17. What do you notice about the individual timestamps? What do you think accounts for this anomaly?

   A. The ACCESS timestamps are zero and were lost during the timeline creation
   B. The ACCESS timestamps are zero because FAT doesn’t track ACCESS events to the hour/minute, only by the day
   C. The MODIFIED timestamps are zero and were lost during the timeline creation
   D. The MODIFIED timestamps are zero because FAT doesn’t track CREATED events to the hour/minute, only by the day

## Lab: Forensic Analysis, Part 2

3. What time was the Projections.txt file created?

   A. 16:48 February 14
   B. 04:48 February 14
   C. 21:48 February 14
   D. 09:48 February 14

## Lab: Forensic Analysis, Part 2

4. What else is peculiar about February 14 at approximately 17:05? (short answer)

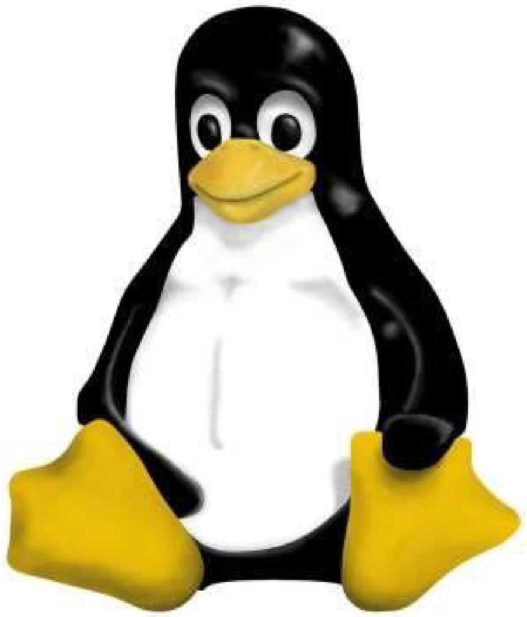
# Lab: Forensic Analysis, Part 2 Answer Key

1. D
2. B
3. A
4. The entire email folder was deleted at 17:05 on February 14. Because the folder was deleted, it received the trashinfo indication. None of the files present in the folder are labeled with the trashinfo tag.

NowSecure™

## Training Outline

1) Linux OS Overview
    a. File system
    b. Basic directory structure
    c. File Permissions
2) Basic Linux Commands
    a. Quiz: Linux Commands
3) Forensic Analysis Tools
    a. Lab: Analysis Part 1
    b. Lab: Analysis Part 2
    c. **Lab: Analysis Part 3**



NowSecure™

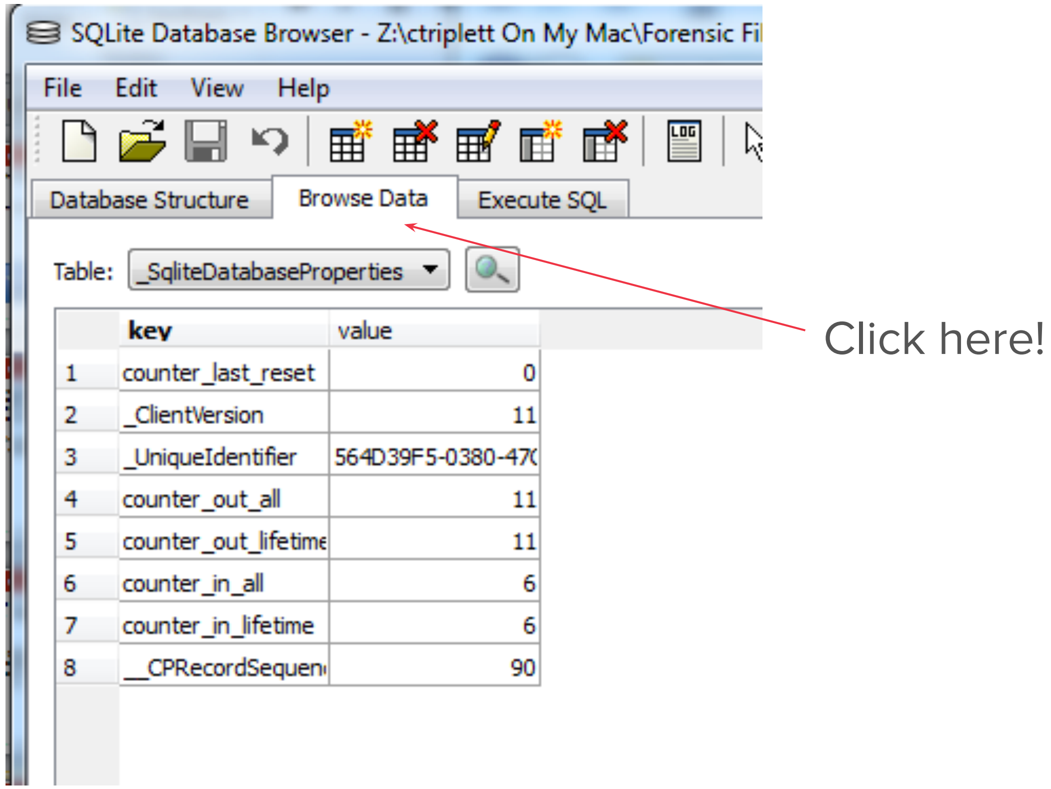
## Lab: Forensic Analysis, Part 3

In this next segment we will focus on locating evidence within an sms database file taken from a mobile phone.

The sms.db file we are going to use is located in /opt/via/sms.db.

# Lab: Forensic Analysis, Part 3

Open the sms.db file in SQLite database viewer, which can be accomplished by simply double clicking on the file.  To view the data contained in the database you have to click on the “Browse Data” tab as shown below:

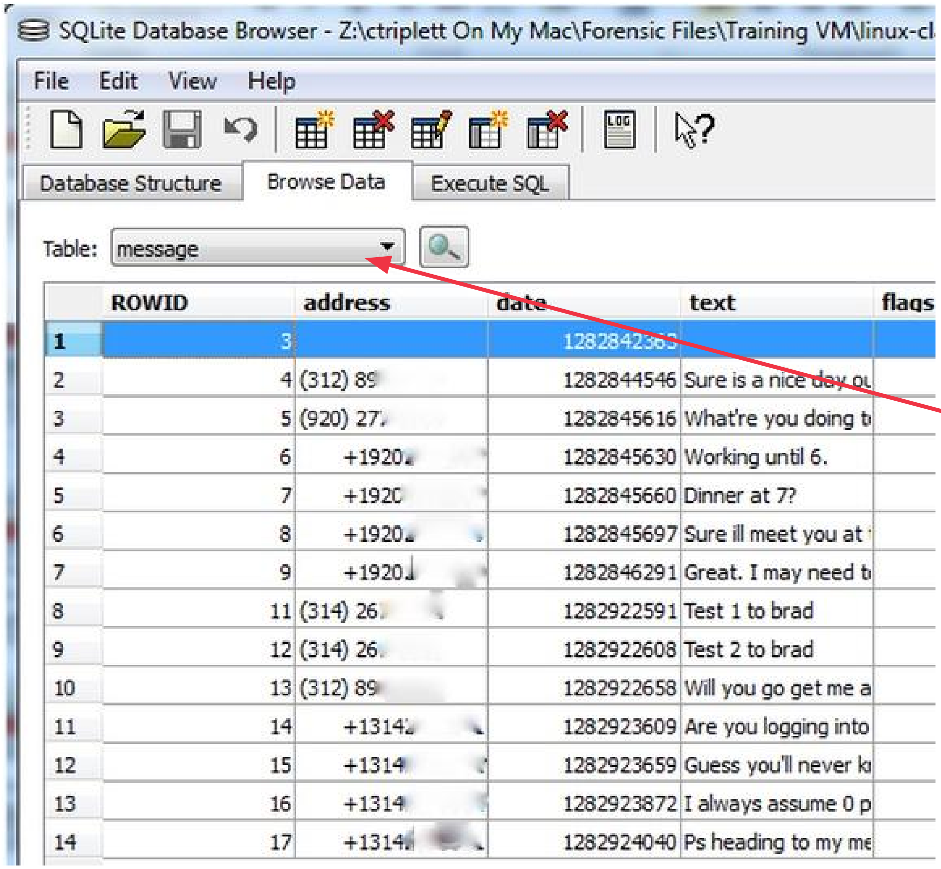

SQLite Database Browser - Z:\ctriplett On My Mac\Forensic Fi
File Edit View Help
Database Structure
Browse Data
Execute SQL
Table: _SqliteDatabaseProperties
Click here!

| | key | value |
|---|---|---|
| 1 | counter_last_reset | 0 |
| 2 | _ClientVersion | 11 |
| 3 | _UniqueIdentifier | 564D39F5-0380-47( |
| 4 | counter_out_all | 11 |
| 5 | counter_out_lifetime | 11 |
| 6 | counter_in_all | 6 |
| 7 | counter_in_lifetime | 6 |
| 8 | __CPRecordSequen | 90 |

NowSecure™

# Lab: Forensic Analysis, Part 3

Select the table where the SMS messages are located to browse logical SMS messages.


SQLite Database Browser - Z:\ctriplett On My Mac\Forensic Files\Training VM\linux-cl
File Edit View Help
Database Structure
Browse Data
Execute SQL
Table: message

| | ROWID | address | date | text | flags |
|---|---|---|---|---|---|
| 1 | 3 | | 1282842363 | | |
| 2 | 4 | (312) 89 | 1282844546 | Sure is a nice day ou | |
| 3 | 5 | (920) 27 | 1282845616 | What're you doing t | |
| 4 | 6 | +1920 | 1282845630 | Working until 6. | |
| 5 | 7 | +1920 | 1282845660 | Dinner at 7? | |
| 6 | 8 | +1920 | 1282845697 | Sure ill meet you at | |
| 7 | 9 | +1920 | 1282846291 | Great. I may need t | |
| 8 | 11 | (314) 26 | 1282922591 | Test 1 to brad | |
| 9 | 12 | (314) 26 | 1282922608 | Test 2 to brad | |
| 10 | 13 | (312) 89 | 1282922658 | Will you go get me a | |
| 11 | 14 | +1314 | 1282923609 | Are you logging into | |
| 12 | 15 | +1314 | 1282923659 | Guess you'll never k | |
| 13 | 16 | +1314 | 1282923872 | I always assume 0 p | |
| 14 | 17 | +1314 | 1282924040 | Ps heading to my me | |

Select the message table from the drop down table menu.

NowSecure™

## Lab: Forensic Analysis, Part 3

Since the database viewer will only display records that haven't been deleted we will need to use a hex editor to look for unallocated messages. SQLite stores deleted messages in freepages until garbage collection removes the record or database from the nand memory.

NowSecure™

## Lab: Forensic Analysis, Part 3

1. Go ahead and open the sms.db database in a hex editor (use hexedit) so you can manually search through the messages.

```
$ hexedit /opt/via/sms.db
```

## Lab: Forensic Analysis, Part 3

2. Search for “Italian” in the ASCII portion of the file. This is just to familiarize you with how the message is displayed in the hex editor since the word “Italian” is part of a logical message and can be viewed in the SQLite file you currently have open.

**You have to first use the TAB key to get the cursor to the ASCII side of the file, then hit the “/” key to initiate the search. Once the search field is displayed simply type Italian and hit enter.**

## Lab: Forensic Analysis, Part 3

Now that you have a feel for what a record looks like in both the logical database file and in a hex editor, go back to the hex and search for some deleted records. Hint: the prosecutor has told you that the suspect was communicating with someone that had a phone number of 312-555-5030. Keep in mind that since the message may have been deleted that part of the phone number might have been reallocated.

NowSecure™

## Lab: Forensic Analysis, Part 3

A good indicator on how many messages have been deleted is to look at the ROWID in the SQLite database viewer. The ROWID is auto incrementing so it can be a helpful tool to reinforce that a message was deleted.

In this case, you can see that rows 1, 2, and 10 are missing! To determine whether any messages at the end are missing, select the "sqlite_sequence" table from sms.db. It will display the most recently assigned ROWID listed next to "message." In this case, 17 was the last assigned ROWID, so we can verify that there were no additional messages deleted after ROWID 17.

NowSecure™

## Lab: Forensic Analysis, Part 3

Another way to view the ASCII portion of the sms.db file is to perform a strings of the file.

3. Perform a strings output of the sms.db file to your desktop.
4. Examine the strings file to see if you can recover any deleted text messages. (Use a text editor to open the file)

```
$ strings /opt/via/sms.db > ~/Desktop/lab1/strings.txt
```

To determine if a record was deleted you will need to compare the SMS records within the strings output file with those found in the hex editor.

# Lab: Forensic Analysis, Part 3

How many deleted messages were you able to recover?

A. 1
B. 2
C. 3
D. 4

**You should have found 2 deleted messages.**
1. “I can give you a key when we”
2. “Piece of cake! Can’t wait to try em…”

## Lab: Forensic Analysis, Part 3

Perform some grep searches against the strings file you just created.

5. Search for: Italian (case sensitive)

How many instances of Italian did you find?

```
$ grep Italian ~/Desktop/strings.txt
```

You should have found one message.

# Lab: Forensic Analysis, Part 3

Perform some grep searches against the strings file you just created.

6. Search for: Italian again but this time case insensitive

Did it return more results?

```
$ grep -i Italian ~/Desktop/strings.txt
```

This command will return one more result for a total of two text messages with the word Italian.

## Lab: Forensic Analysis, Part 3

Perform some grep searches against the strings file you just created.

7. Search for: “you” (a common word) and show the results one screen at a time.

```
$ grep you ~/Desktop/strings.txt | less
```

NowSecure™

## Lab: Forensic Analysis, Part 3 Summary

You can see that Linux tools can be extremely helpful in recovering evidence from all forms of media.  The information presented in this course will especially come in handy for mobile forensics since Android is built on the Linux core.

If you have any questions about the material feel free to review this training again, or contact us with more specific questions.

## Lab: Forensic Analysis, Part 3 Summary

Nicely done!  You are finished!

The goal of this course was to familiarize you with Linux and some powerful tools and commands that can be used as a forensic investigative tool.

## Recommended Downloads

Like this? Share with your friends (they'll thank you). Here's a tweet you can copy / paste:

> Want great #Linux tips and tutorials? Check out "Linux for Mobile Forensics" from @NowSecureMobile: https://www.nowsecure.com/resources/linux-forensics-training/

If you like this training, you will love our other resources:

Reports & Trainings:

- JTAG for Mobile Forensics (3 part training)
- Secure Mobile Development Best Practices

Free Tools & Software:

- Forensics Suite - Community Edition
- App Testing Suite - Community Edition

NowSecure™

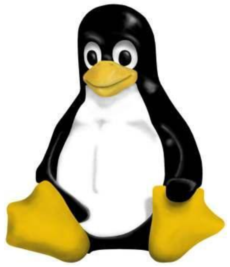
This PDF was developed by NowSecure (@NowSecureMobile) and is not for redistribution. For corrections, comments or additional information please contact:

NowSecure Support

support@nowsecure.com

312-878-1100

Thank you for taking our course!



NowSecure™