Worcester Polytechnic Institute
Department of Computer Science

Activity 3: Linux Data Acquisition

Introduction

Data Acquisition is the process of copying data. For digital forensics, it's the task of collecting digital evidence from electronic media. There are two types of data acquisition: static acquisition and live acquisition. In this practice, you'll perform a static data acquisition using the Linux dd and dcfldd commands. Operations in this guide are performed in CAINE Linux. You may also use Kali Linux for the same practice.

Objectives

- Get familiar with the data acquisition process.
- Prepare a target drive for data acquisition.
- Use Linux data acquisition tool to acquire data from an evidence drive.
- Validate the acquired data.

Tasks

Please Note: Usually you need to have 2 drives for data acquisition: ***1 source (evidence) drive*** and ***1 target drive*** to hold the image. The correct order of a data acquisition is:
Part 1: prepare the ***target*** drive by zeroing-out the ***target*** drive and creating a new file system on it;
Part 2: perform a data acquisition by acquiring data from ***evidence*** drive to ***target*** drive;
Part 3: validate the acquisition.

Special Notes:

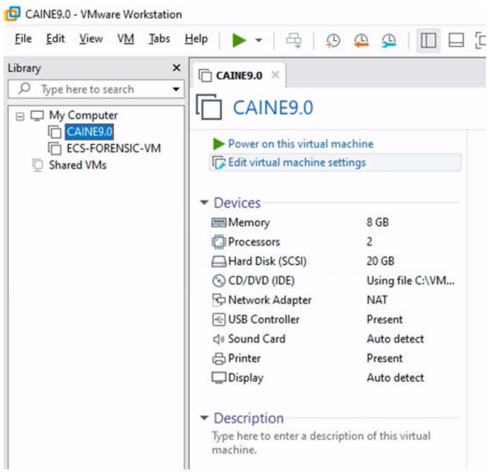
Please **take screenshots for the major steps**: after zero-out; after the partition is created; after the file system is established; after the acquisition is completed; after the validation is completed. You'll need these screenshots to demonstrate that you've completed all the activities required in this hands-on.

Part 1. Add the source drive (evidence drive) and target drive

We'll need to add two drives to the virtual machine to act as the *source (evidence) drive* and *target drive* respectively. In real investigation, you will get the evidence drive from suspect/crime scene directly. Since we do not have an evidence drive from suspect now, we'll add one on our own to pretend it is the evidence drive.

(For VMware Workstation)

1. In VMware Workstation CAINE, click on CAINE9.0, and then click on Edit virtual machine settings.


CAINE9.0 - VMware Workstation

File Edit View VM Tabs Help

Library

My Computer
- CAINE9.0
- ECS-FORENSIC-VM

Shared VMs

CAINE9.0

Power on this virtual machine

Edit virtual machine settings

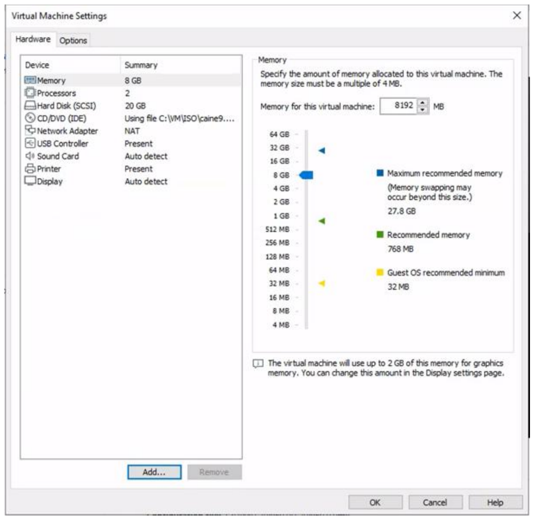
Devices

Memory	8 GB
Processors	2
Hard Disk (SCSI)	20 GB
CD/DVD (IDE)	Using file C:\VM...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Description

Type here to enter a description of this virtual machine.

2. Click on “Add”.


Virtual Machine Settings

Hardware | Options

Device	Summary
Memory	8 GB
Processors	2
Hard Disk (SCSI)	20 GB
CD/DVD (IDE)	Using file C:\VM\ISO\caine9....
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

Memory for this virtual machine: 8192 MB

64 GB, 32 GB, 16 GB, 8 GB, 4 GB, 2 GB, 1 GB, 512 MB, 256 MB, 128 MB, 64 MB, 32 MB, 16 MB, 8 MB, 4 MB

🟦 Maximum recommended memory (Memory swapping may occur beyond this size.) 27.8 GB

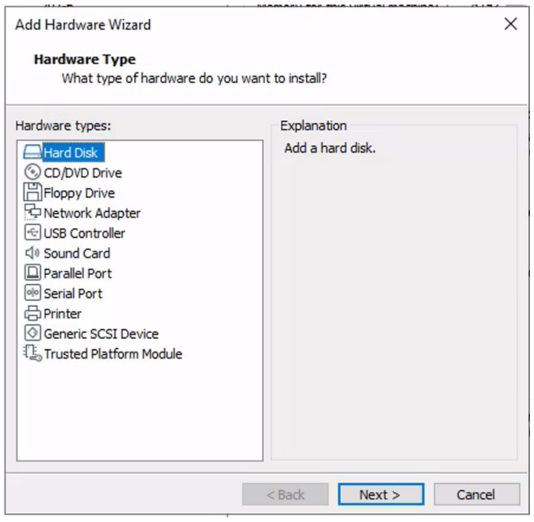
🟩 Recommended memory 768 MB

🟨 Guest OS recommended minimum 32 MB

The virtual machine will use up to 2 GB of this memory for graphics memory. You can change this amount in the Display settings page.

Add... | Remove | OK | Cancel | Help

3. Click on “Hard Disk”, and then “Next”.


Add Hardware Wizard

Hardware Type

What type of hardware do you want to install?

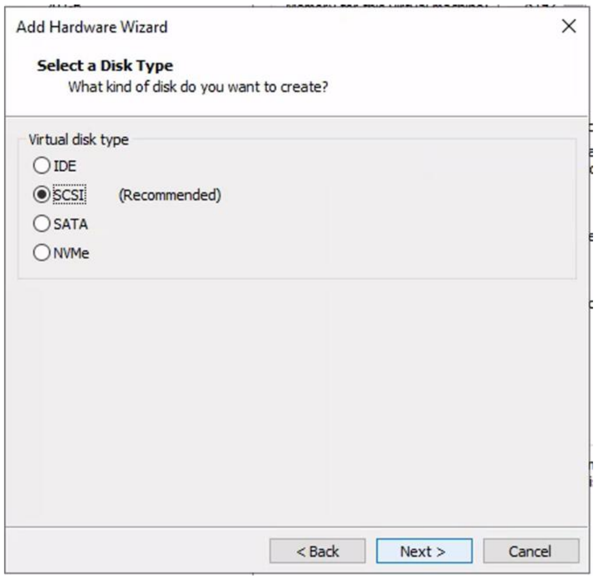
Hardware types:
- Hard Disk
- CD/DVD Drive
- Floppy Drive
- Network Adapter
- USB Controller
- Sound Card
- Parallel Port
- Serial Port
- Printer
- Generic SCSI Device
- Trusted Platform Module

Explanation

Add a hard disk.

< Back | Next > | Cancel

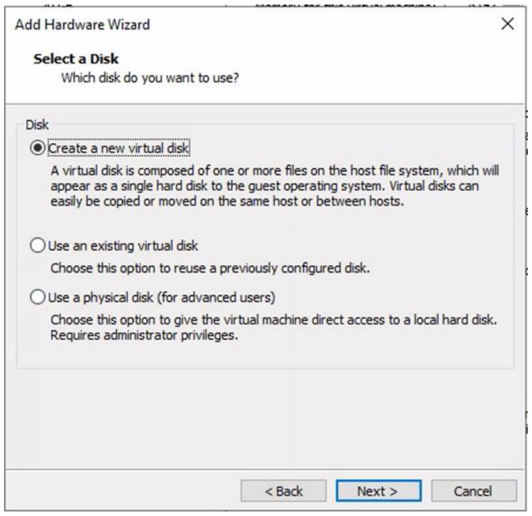
4. Leave “SCSI” as recommended, and click on “Next”.


Add Hardware Wizard

Select a Disk Type
What kind of disk do you want to create?

Virtual disk type
- ( ) IDE
- (●) SCSI (Recommended)
- ( ) SATA
- ( ) NVMe

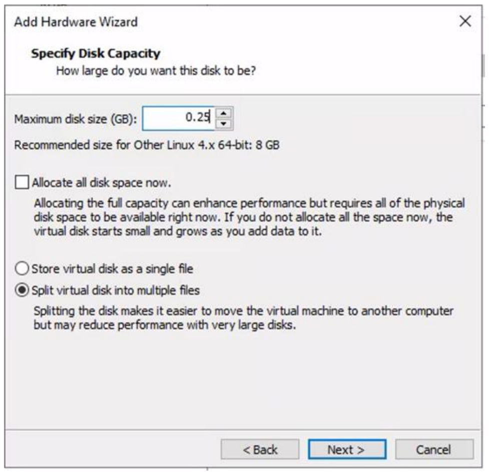
5. Select “Create a new virtual disk” and then “Next”.


Add Hardware Wizard

Select a Disk
Which disk do you want to use?

Disk
- (●) Create a new virtual disk
  A virtual disk is composed of one or more files on the host file system, which will appear as a single hard disk to the guest operating system. Virtual disks can easily be copied or moved on the same host or between hosts.
- ( ) Use an existing virtual disk
  Choose this option to reuse a previously configured disk.
- ( ) Use a physical disk (for advanced users)
  Choose this option to give the virtual machine direct access to a local hard disk. Requires administrator privileges.

6. Change the disk size of “0.25” GB, we want to make it as small as possible so that we can finish the acquisition very soon, and then “Next”.

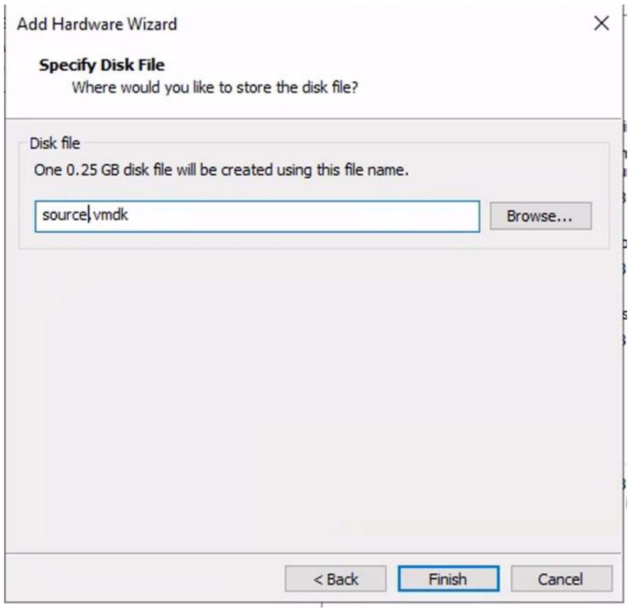

Add Hardware Wizard

Specify Disk Capacity
How large do you want this disk to be?

Maximum disk size (GB): 0.25
Recommended size for Other Linux 4.x 64-bit: 8 GB

- [ ] Allocate all disk space now.
  Allocating the full capacity can enhance performance but requires all of the physical disk space to be available right now. If you do not allocate all the space now, the virtual disk starts small and grows as you add data to it.
- ( ) Store virtual disk as a single file
- (●) Split virtual disk into multiple files
  Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

7. Change the disk name into “source.vmdk”, and then click on “Browse…”.



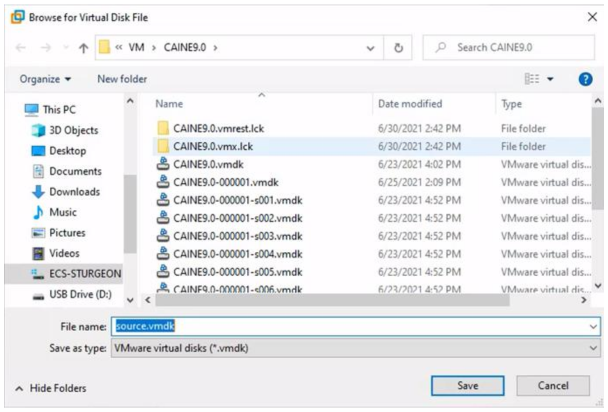
Add Hardware Wizard

Specify Disk File
Where would you like to store the disk file?

Disk file
One 0.25 GB disk file will be created using this file name.

source.vmdk

8. Save the source.vmdk in the CAINE9.0 folder, click on “Save”. And then go back to click on “Finish”.



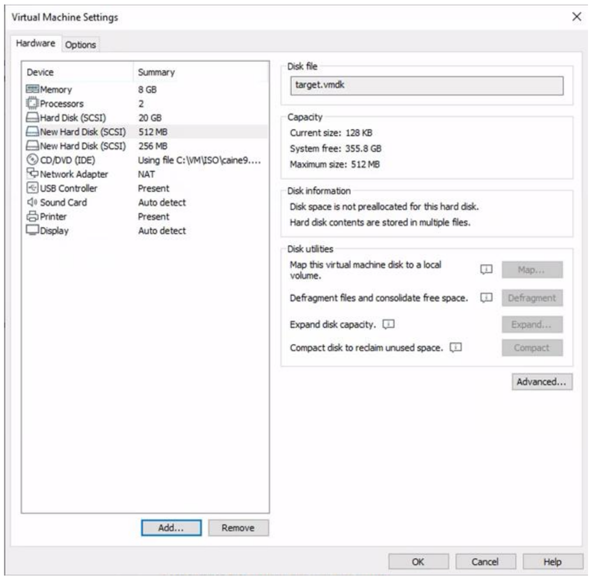
Browse for Virtual Disk File

« VM › CAINE9.0 ›

Name	Date modified	Type
CAINE9.0.vmrest.lck	6/30/2021 2:42 PM	File folder
CAINE9.0.vmx.lck	6/30/2021 2:42 PM	File folder
CAINE9.0.vmdk	6/23/2021 4:02 PM	VMware virtual dis...
CAINE9.0-000001.vmdk	6/25/2021 2:09 PM	VMware virtual dis...
CAINE9.0-000001-s001.vmdk	6/23/2021 4:52 PM	VMware virtual dis...
CAINE9.0-000001-s002.vmdk	6/23/2021 4:52 PM	VMware virtual dis...
CAINE9.0-000001-s003.vmdk	6/23/2021 4:52 PM	VMware virtual dis...
CAINE9.0-000001-s004.vmdk	6/23/2021 4:52 PM	VMware virtual dis...
CAINE9.0-000001-s005.vmdk	6/23/2021 4:52 PM	VMware virtual dis...
CAINE9.0-000001-s006.vmdk	6/23/2021 4:52 PM	VMware virtual dis...

File name: source.vmdk
Save as type: VMware virtual disks (*.vmdk)

9. Repeat Step 1-8 to add a new target drive. Please note: The size of the target drive should be bigger than the source drive. Please use “0.5”GB in this activity. The name of the disk can be “target.vmdk”.
10. Now you can see the VM has three hard drives: the 20GB original hard drive, the 512MB target drive, and the 256MB source (evidence) drive.


Virtual Machine Settings

Device	Summary
Memory	8 GB
Processors	2
Hard Disk (SCSI)	20 GB
New Hard Disk (SCSI)	512 MB
New Hard Disk (SCSI)	256 MB
CD/DVD (IDE)	Using file C:\VM\ISO\caine9....
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

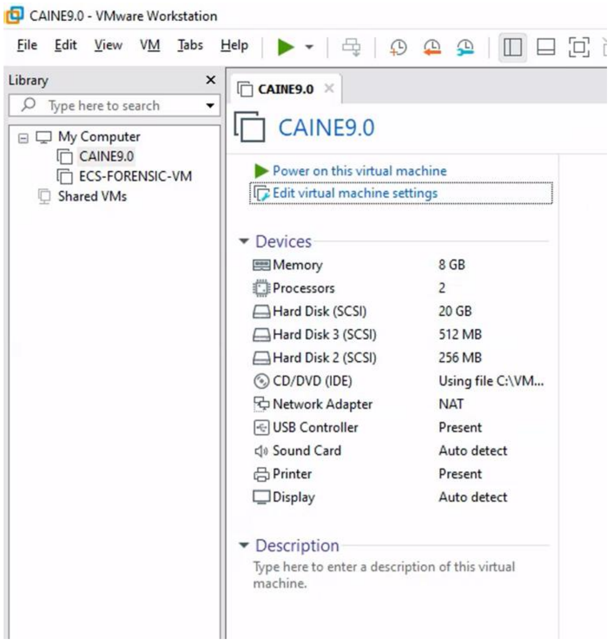
Disk file
target.vmdk

Capacity
Current size: 128 KB
System free: 355.8 GB
Maximum size: 512 MB

Disk information
Disk space is not preallocated for this hard disk.
Hard disk contents are stored in multiple files.

Disk utilities
Map this virtual machine disk to a local volume.
Defragment files and consolidate free space.
Expand disk capacity.
Compact disk to reclaim unused space.

11. Next you can power on the virtual machine.


CAINE9.0 - VMware Workstation

CAINE9.0

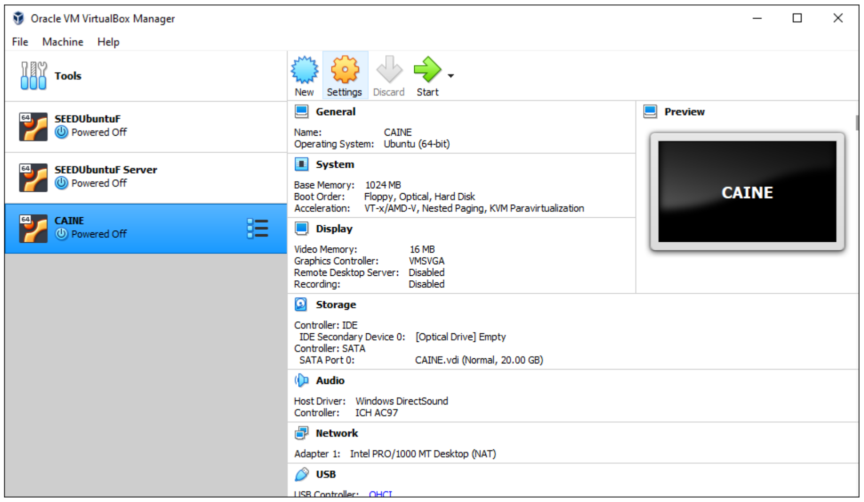
Devices

Memory	8 GB
Processors	2
Hard Disk (SCSI)	20 GB
Hard Disk 3 (SCSI)	512 MB
Hard Disk 2 (SCSI)	256 MB
CD/DVD (IDE)	Using file C:\VM...
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Description
Type here to enter a description of this virtual machine.

(For VirtualBox)

1. In VirtualBox Manager, click on CAINE, and then click on settings.



SEEDUbuntuF
Powered Off
SEEDUbuntuF Server
Powered Off
CAINE
Powered Off

General
Name: CAINE
Operating System: Ubuntu (64-bit)

System
Base Memory: 1024 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Display
Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

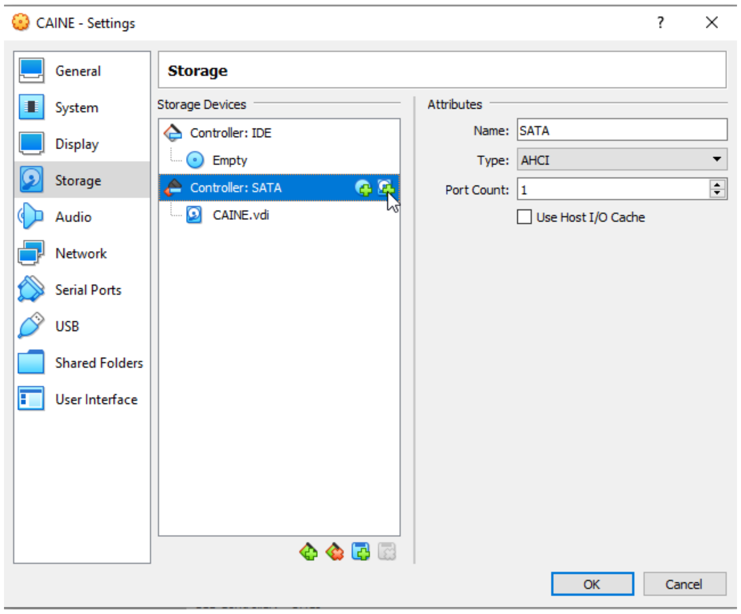
Storage
Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: CAINE.vdi (Normal, 20.00 GB)

Audio
Host Driver: Windows DirectSound
Controller: ICH AC97

Network
Adapter 1: Intel PRO/1000 MT Desktop (NAT)

USB
USB Controller: OHCI

2. In the settings, click on Storage, select Controller: SATA. Then, click on "Adds hard disk" as shown below.

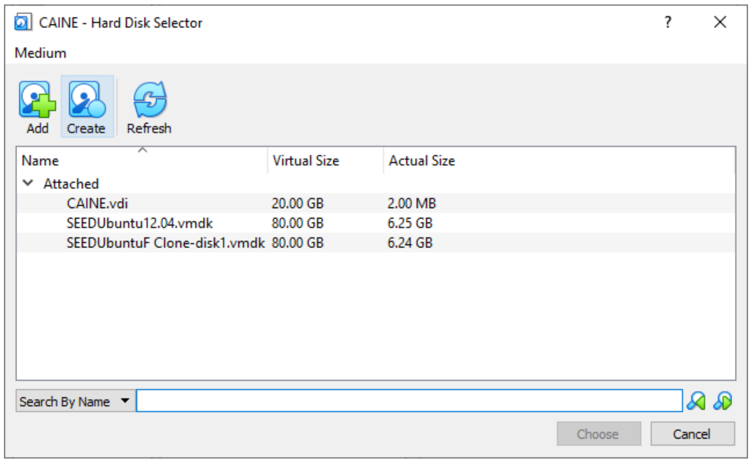


Storage

Storage Devices
Controller: IDE
Empty
Controller: SATA
CAINE.vdi

Attributes
Name: SATA
Type: AHCI
Port Count: 1
Use Host I/O Cache

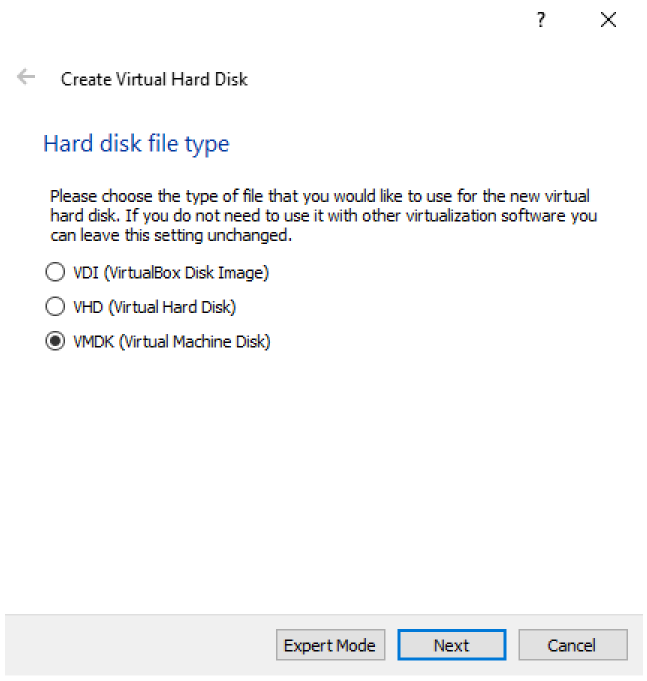
3. Click "Create"


CAINE - Hard Disk Selector

Medium

Name	Virtual Size	Actual Size
Attached		
CAINE.vdi	20.00 GB	2.00 MB
SEEDUbuntu12.04.vmdk	80.00 GB	6.25 GB
SEEDUbuntuF Clone-disk1.vmdk	80.00 GB	6.24 GB

4. Select VMDK (Virtual Machine Disk), then “Next”

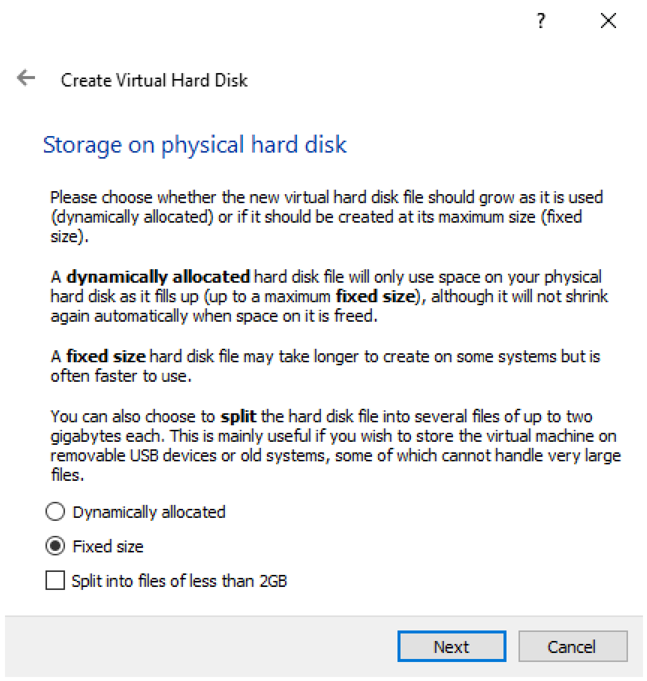

Create Virtual Hard Disk

Hard disk file type

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

- ○ VDI (VirtualBox Disk Image)
- ○ VHD (Virtual Hard Disk)
- ● VMDK (Virtual Machine Disk)

5. Select “Fixed size”, then “Next”


Create Virtual Hard Disk

Storage on physical hard disk

Please choose whether the new virtual hard disk file should grow as it is used (dynamically allocated) or if it should be created at its maximum size (fixed size).

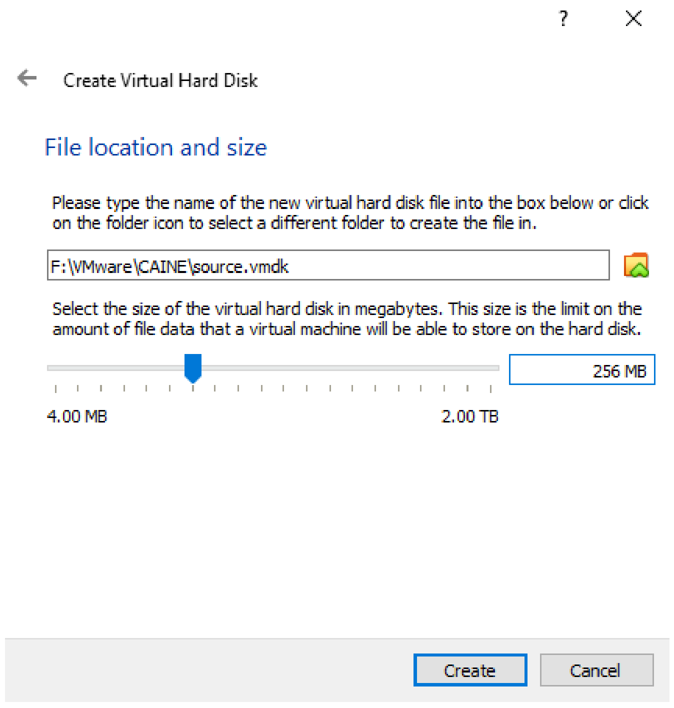
A **dynamically allocated** hard disk file will only use space on your physical hard disk as it fills up (up to a maximum **fixed size**), although it will not shrink again automatically when space on it is freed.

A **fixed size** hard disk file may take longer to create on some systems but is often faster to use.

You can also choose to **split** the hard disk file into several files of up to two gigabytes each. This is mainly useful if you wish to store the virtual machine on removable USB devices or old systems, some of which cannot handle very large files.

○ Dynamically allocated
● Fixed size
☐ Split into files of less than 2GB

6. Rename the disk name to “source.vmdk”, and assign 256 MB to the disk size. Then “Create”


Create Virtual Hard Disk

File location and size

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

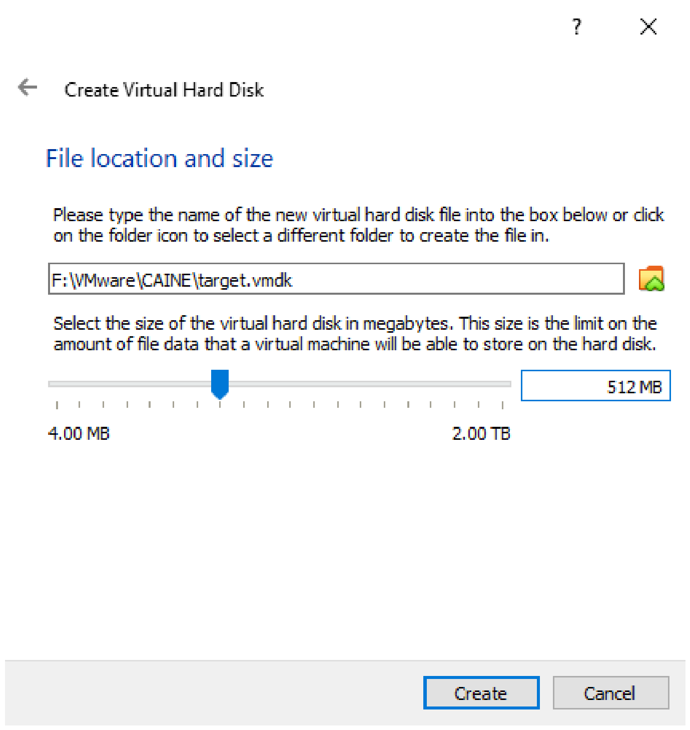
F:\VMware\CAINE\source.vmdk

Select the size of the virtual hard disk 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 disk.

256 MB

Repeat step 3-5 to create another drive

7. Rename the disk name to “target.vmdk”, and assign 512 MB to the disk size. Then “Create”



File location and size

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

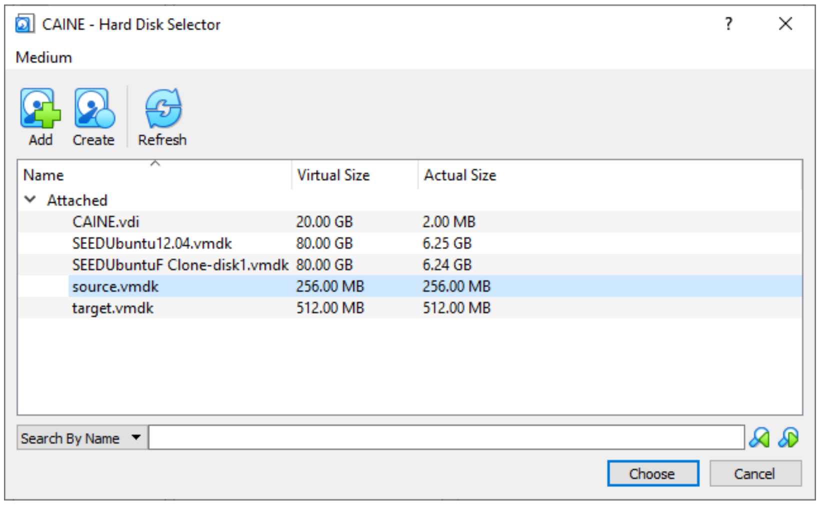
F:\VMware\CAINE\target.vmdk

Select the size of the virtual hard disk 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 disk.

512 MB

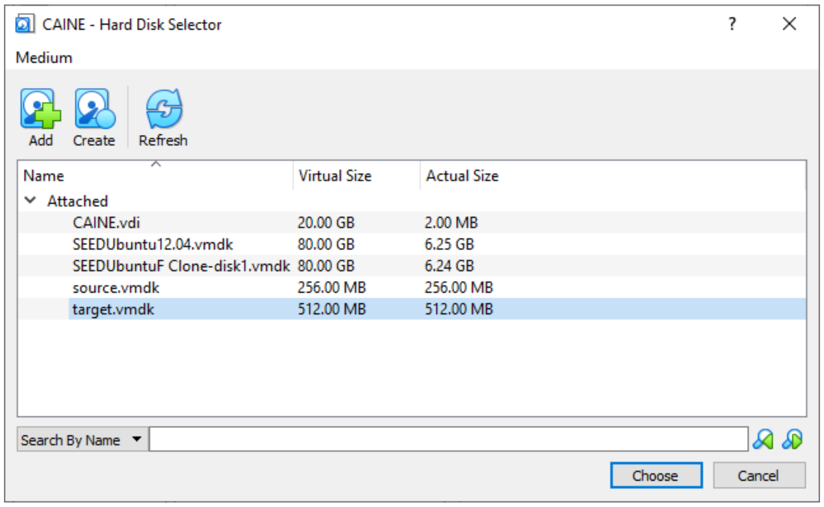
4.00 MB — 2.00 TB

8. Back to the hard disk selector, select “source.vmdk” and click “Choose”.



Name	Virtual Size	Actual Size
Attached		
CAINE.vdi	20.00 GB	2.00 MB
SEEDUbuntu12.04.vmdk	80.00 GB	6.25 GB
SEEDUbuntuF Clone-disk1.vmdk	80.00 GB	6.24 GB
source.vmdk	256.00 MB	256.00 MB
target.vmdk	512.00 MB	512.00 MB

9. Click “Adds hard disk” again, select “target.vmdk” and click “Choose”.

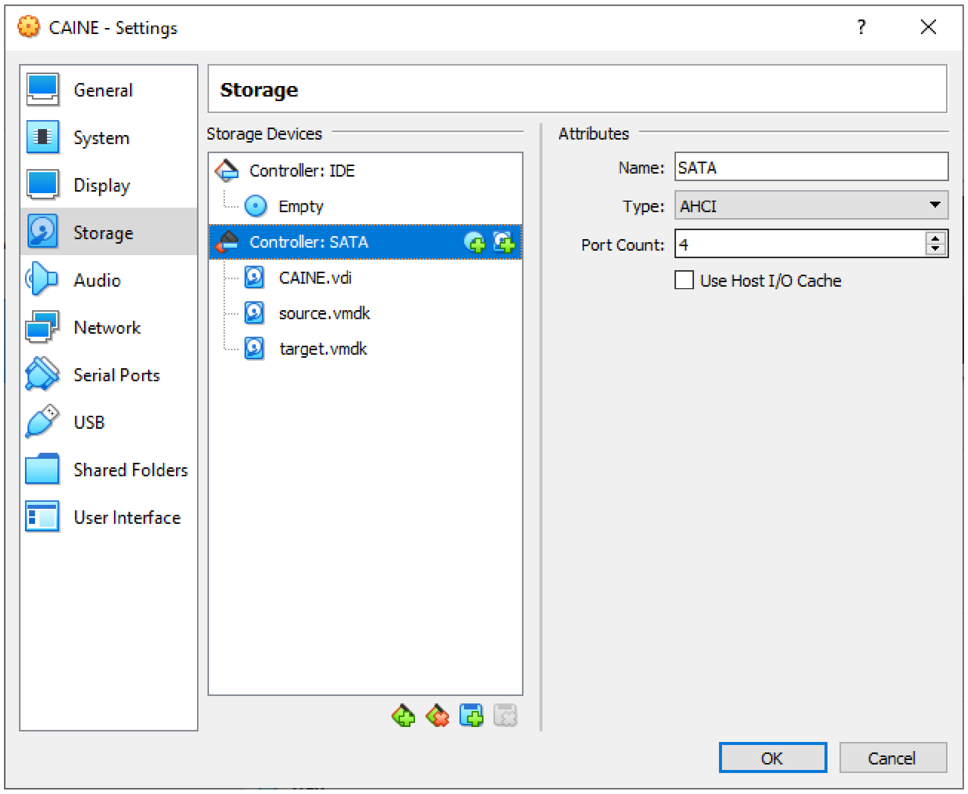


CAINE - Hard Disk Selector

Medium

Name	Virtual Size	Actual Size
Attached		
CAINE.vdi	20.00 GB	2.00 MB
SEEDUbuntu12.04.vmdk	80.00 GB	6.25 GB
SEEDUbuntuF Clone-disk1.vmdk	80.00 GB	6.24 GB
source.vmdk	256.00 MB	256.00 MB
target.vmdk	512.00 MB	512.00 MB

10. Now you can see the VM has three hard drives: the 20GB original hard drive, the 512MB target drive, and the 256MB source (evidence) drive. Then, click "OK"



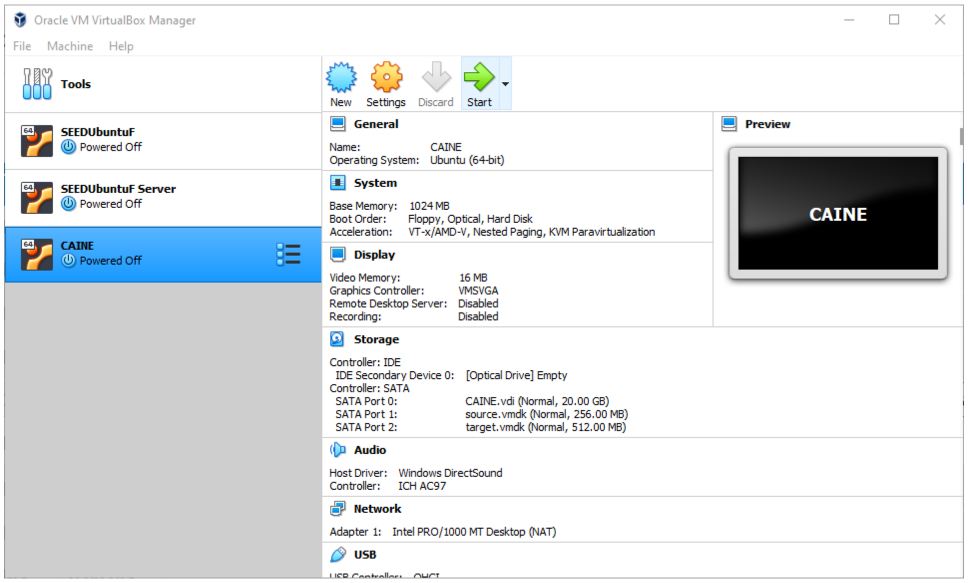
CAINE - Settings

Storage

Storage Devices
- Controller: IDE
  - Empty
- Controller: SATA
  - CAINE.vdi
  - source.vmdk
  - target.vmdk

Attributes
Name: SATA
Type: AHCI
Port Count: 4
Use Host I/O Cache

11. Next you can start the virtual machine.


Oracle VM VirtualBox Manager

File Machine Help

Tools

SEEDUbuntuF
Powered Off

SEEDUbuntuF Server
Powered Off

CAINE
Powered Off

New Settings Discard Start

General

Name: CAINE
Operating System: Ubuntu (64-bit)

System

Base Memory: 1024 MB
Boot Order: Floppy, Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

Display

Video Memory: 16 MB
Graphics Controller: VMSVGA
Remote Desktop Server: Disabled
Recording: Disabled

Storage

Controller: IDE
IDE Secondary Device 0: [Optical Drive] Empty
Controller: SATA
SATA Port 0: CAINE.vdi (Normal, 20.00 GB)
SATA Port 1: source.vmdk (Normal, 256.00 MB)
SATA Port 2: target.vmdk (Normal, 512.00 MB)

Audio

Host Driver: Windows DirectSound
Controller: ICH AC97

Network

Adapter 1: Intel PRO/1000 MT Desktop (NAT)

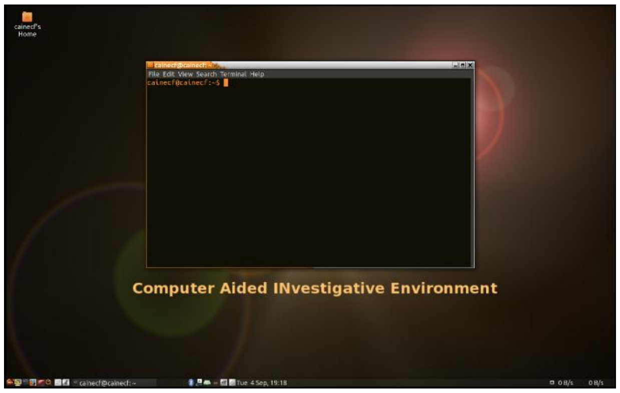
USB

Preview

CAINE

Part1. Prepare the target drive

12. In CAINE, Find terminal in Main->system tools->MATE Terminal. In Kali Linux, Find the terminal on the panel to the left of the screen. Open the terminal.

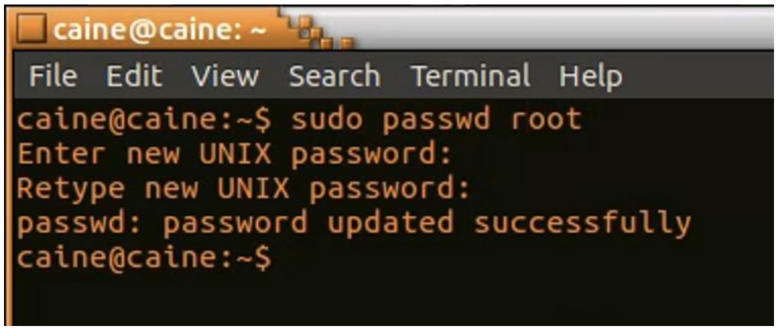


13. To change to root account, type command

    su

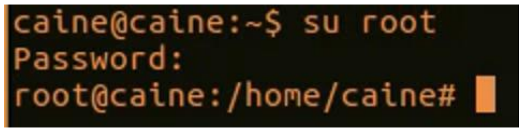
    And provide the password.

    You may see authentication failure in this step. That's because you didn't set up the password for **root** account yet. Please set up the password for root use the following way:



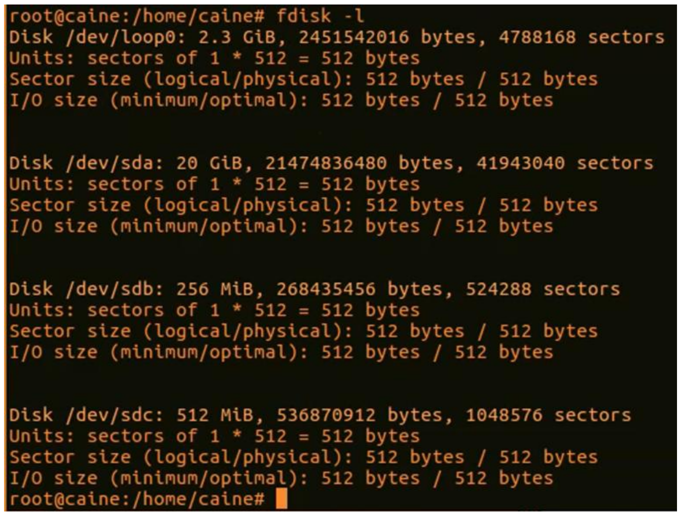
```
caine@caine:~$ sudo passwd root
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
caine@caine:~$
```

Then you can change to root account using the password you just set up:



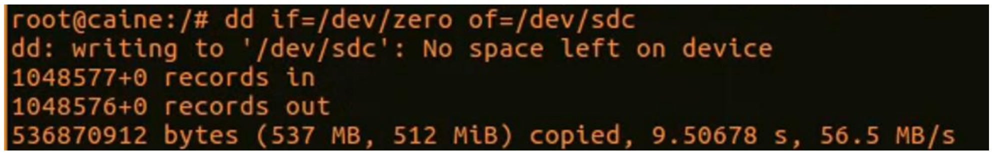
```
caine@caine:~$ su root
Password:
root@caine:/home/caine# 
```

14. Type ***fdisk –l*** to show the current disks. You can see there are three disks showing in the screenshot.
    a. /dev/sda is the original hard drive assigned to the CAINE vm;
    b. /dev/sdb (256MB) is the source (evidence) drive we added;
    c. /dev/sdc (512MB) is the target drive we'll use to store the image.
    d. **!!!! Please always make sure you are using the smaller driver as the source/evidence drive, and the bigger drive as the target drive.**



```
root@caine:/home/caine# fdisk -l
Disk /dev/loop0: 2.3 GiB, 2451542016 bytes, 4788168 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sdb: 256 MiB, 268435456 bytes, 524288 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sdc: 512 MiB, 536870912 bytes, 1048576 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@caine:/home/caine# 
```

15. Zero out the target drive:
    dd if=/dev/zero of=/dev/sdc

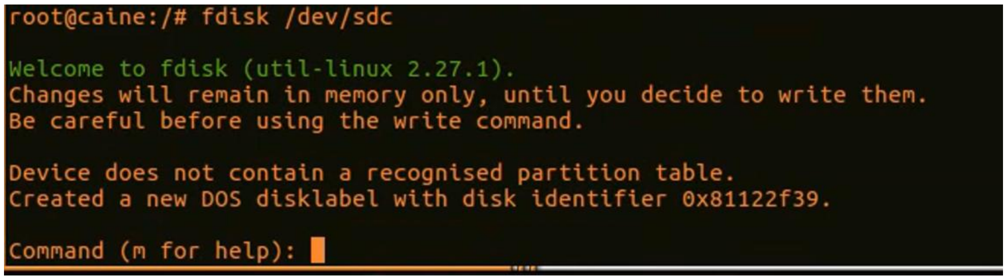


```
root@caine:/# dd if=/dev/zero of=/dev/sdc
dd: writing to '/dev/sdc': No space left on device
1048577+0 records in
1048576+0 records out
536870912 bytes (537 MB, 512 MiB) copied, 9.50678 s, 56.5 MB/s
```

This will take some time depending on the virtual machine configuration. In this activity, it is very fast because we have a very small hard drive: 512MB. *If you are zero-out a big drive, please be patient as it is writing bit by bit. The time can vary from seconds to days.*
Note:
1) Make sure you are zeroing out the target drive, NOT the original drive or other drives!
2) Make sure you are zeroing out the entire drive, not just a partition under it. For example, /dev/sdb is the entire drive, but /dev/sdb1 is just a partition.

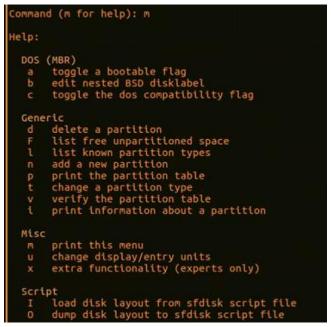
Currently the target drive is completely empty with all 0s. Next, you need to create a partition on the target drive and then create a file system on the partition, so that (image) files can be stored on the drive.

16. Type ***fdisk /dev/sdc*** to begin creating a partition on the target drive.



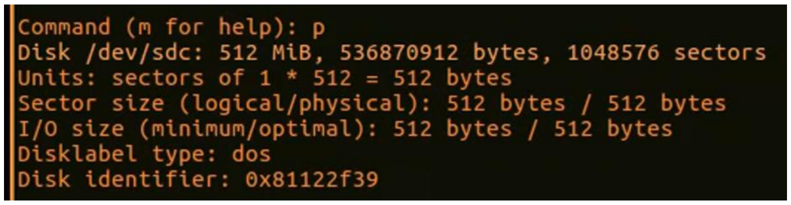
```
root@caine:/# fdisk /dev/sdc

Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognised partition table.
Created a new DOS disklabel with disk identifier 0x81122f39.

Command (m for help): 
```

   a. Then type ***m*** to show the menu



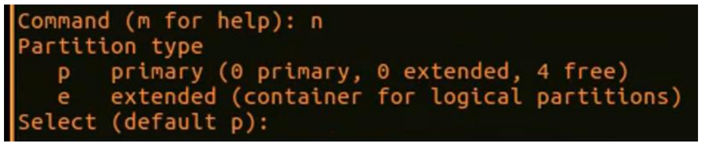
```
Command (m for help): m

Help:

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

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

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

  Script
   I   load disk layout from sfdisk script file
   O   dump disk layout to sfdisk script file
```

   b. Type ***p*** to print the partition table and see if there are any partitions on /dev/sdb
   If there are no partitions on target drive, the output will sometimes be similar to:
   Disk /dev/sdb doesn't contain a valid partition table
   Or the output simply doesn't show any partition information.
   Then you'll need to create a new partition following the steps below.



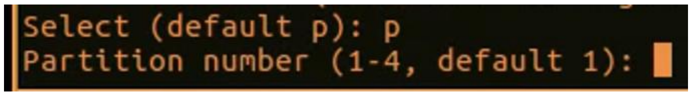
```
Command (m for help): p
Disk /dev/sdc: 512 MiB, 536870912 bytes, 1048576 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x81122f39
```

   c. Type ***n*** and hit enter, to create a new partition. It lists two partition types: primary and extended.



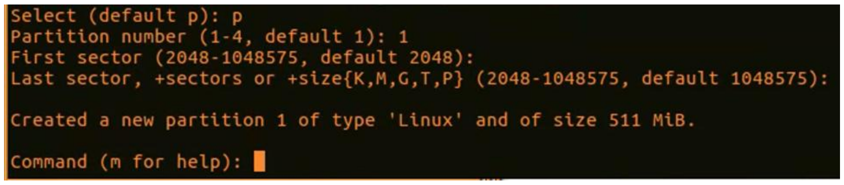
```
Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p):
```

   d. Type ***p*** and hit enter, to choose a primary partition table.



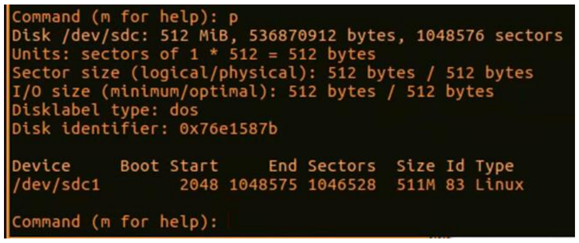
```
Select (default p): p
Partition number (1-4, default 1): 
```

   e. Type ***1*** and hit enter, to select the first partition; then hit enter for first section and last sector to use the default value



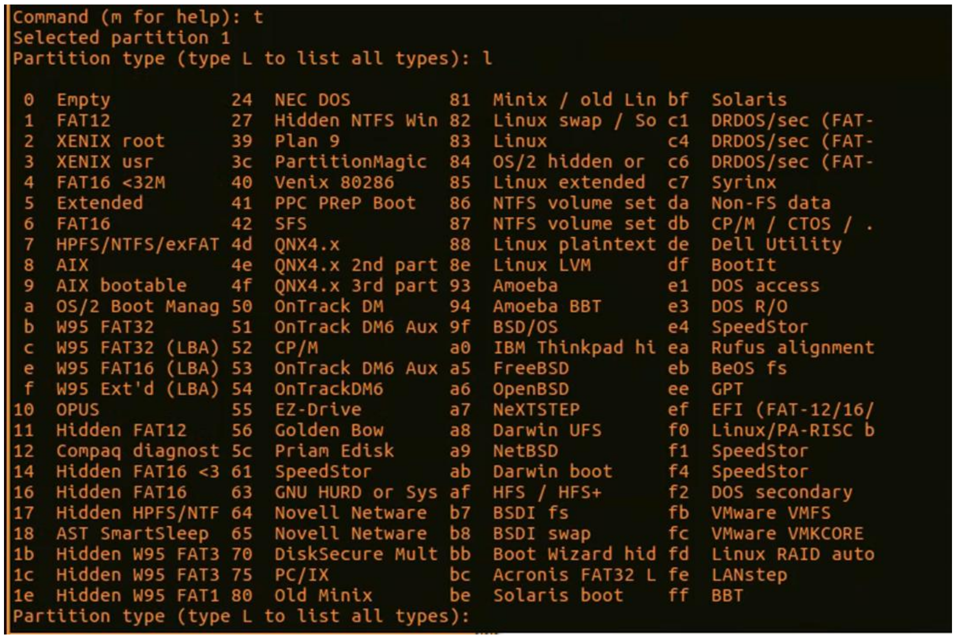
```
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-1048575, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-1048575, default 1048575):

Created a new partition 1 of type 'Linux' and of size 511 MiB.

Command (m for help):
```

f. After the new partition is created, type ***p*** again to show current partitions on /dev/sdb and you'll see the newly created partition. In this case /dev/sdc1 is the new partition created.



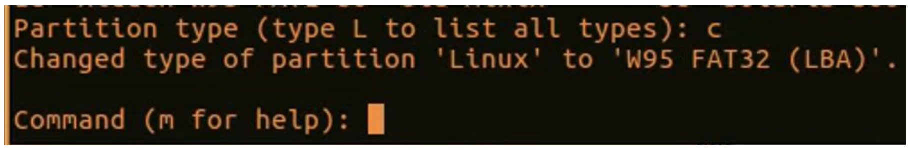
```
Command (m for help): p
Disk /dev/sdc: 512 MiB, 536870912 bytes, 1048576 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x76e1587b

Device     Boot Start     End Sectors  Size Id Type
/dev/sdc1        2048 1048575 1046528  511M 83 Linux

Command (m for help):
```

17. In last step we created a new linux partition /dev/sdc1. In this step, we'll change the newly created partition to windows 95 FAT32 file system.
    a. Type ***m*** to show the menu
    b. Type ***t*** to change the partition type
    c. Type ***l*** (lowercase L) to show available file systems and their code values



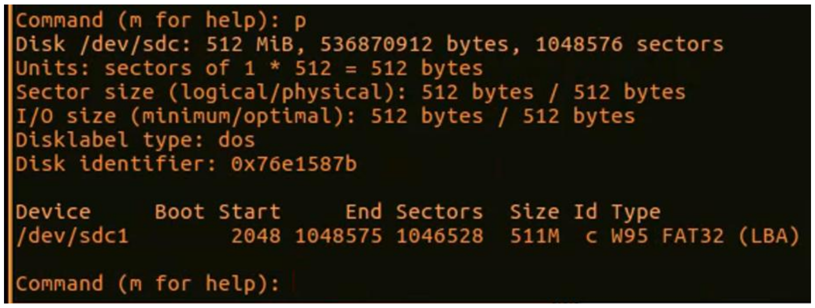
```
Command (m for help): t
Selected partition 1
Partition type (type L to list all types): l

 0  Empty           24  NEC DOS         81  Minix / old Lin bf  Solaris
 1  FAT12           27  Hidden NTFS Win 82  Linux swap / So c1  DRDOS/sec (FAT-
 2  XENIX root      39  Plan 9          83  Linux           c4  DRDOS/sec (FAT-
 3  XENIX usr       3c  PartitionMagic  84  OS/2 hidden or  c6  DRDOS/sec (FAT-
 4  FAT16 <32M      40  Venix 80286     85  Linux extended  c7  Syrinx
 5  Extended        41  PPC PReP Boot   86  NTFS volume set da  Non-FS data
 6  FAT16           42  SFS             87  NTFS volume set db  CP/M / CTOS / .
 7  HPFS/NTFS/exFAT 4d  QNX4.x          88  Linux plaintext de  Dell Utility
 8  AIX             4e  QNX4.x 2nd part 8e  Linux LVM       df  BootIt
 9  AIX bootable    4f  QNX4.x 3rd part 93  Amoeba          e1  DOS access
 a  OS/2 Boot Manag 50  OnTrack DM      94  Amoeba BBT      e3  DOS R/O
 b  W95 FAT32       51  OnTrack DM6 Aux 9f  BSD/OS          e4  SpeedStor
 c  W95 FAT32 (LBA) 52  CP/M            a0  IBM Thinkpad hi ea  Rufus alignment
 e  W95 FAT16 (LBA) 53  OnTrack DM6 Aux a5  FreeBSD         eb  BeOS fs
 f  W95 Ext'd (LBA) 54  OnTrackDM6      a6  OpenBSD         ee  GPT
10  OPUS            55  EZ-Drive        a7  NeXTSTEP        ef  EFI (FAT-12/16/
11  Hidden FAT12    56  Golden Bow      a8  Darwin UFS      f0  Linux/PA-RISC b
12  Compaq diagnost 5c  Priam Edisk     a9  NetBSD          f1  SpeedStor
14  Hidden FAT16 <3 61  SpeedStor       ab  Darwin boot     f4  SpeedStor
16  Hidden FAT16    63  GNU HURD or Sys af  HFS / HFS+      f2  DOS secondary
17  Hidden HPFS/NTF 64  Novell Netware  b7  BSDI fs         fb  VMware VMFS
18  AST SmartSleep  65  Novell Netware  b8  BSDI swap       fc  VMware VMKCORE
1b  Hidden W95 FAT3 70  DiskSecure Mult bb  Boot Wizard hid fd  Linux RAID auto
1c  Hidden W95 FAT3 75  PC/IX           bc  Acronis FAT32 L fe  LANstep
1e  Hidden W95 FAT1 80  Old Minix       be  Solaris boot    ff  BBT
Partition type (type L to list all types):
```

d. Type ***c*** to change the partition to Windows 95 FAT32(LBA)



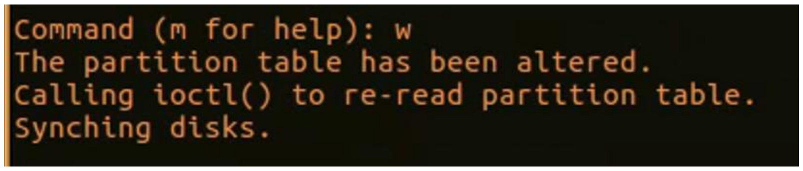
```
Partition type (type L to list all types): c
Changed type of partition 'Linux' to 'W95 FAT32 (LBA)'.

Command (m for help):
```

e. Type ***p*** to display the newly changed drive



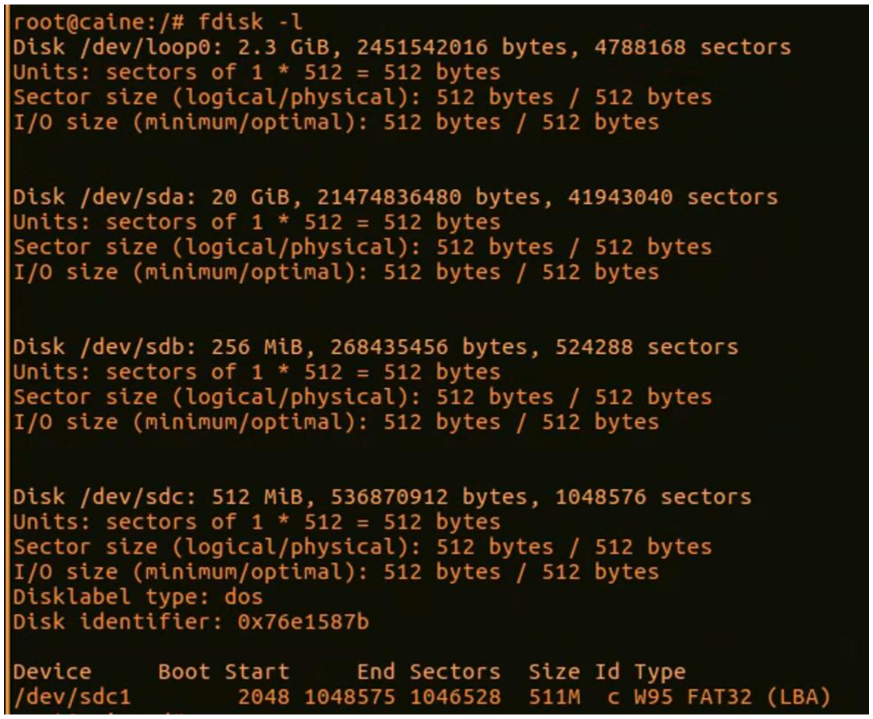
```
Command (m for help): p
Disk /dev/sdc: 512 MiB, 536870912 bytes, 1048576 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x76e1587b

Device     Boot Start     End Sectors  Size Id Type
/dev/sdc1        2048 1048575 1046528  511M  c W95 FAT32 (LBA)

Command (m for help):
```

f. Type ***w*** to save/write the newly created partition to the /dev/sdb drive.



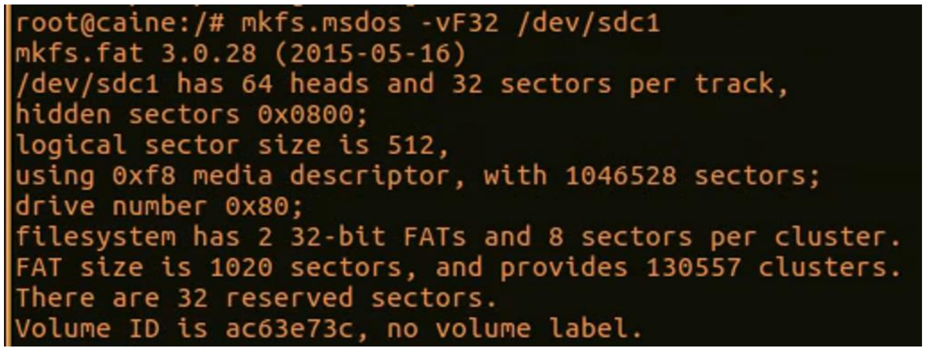
```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Synching disks.
```

18. Type ***fdisk –l*** to see the drives again. Now you can see the target drive /dev/sdc has a new partition /dev/sdc1 with a new file system FAT 32.



```
root@caine:/# fdisk -l
Disk /dev/loop0: 2.3 GiB, 2451542016 bytes, 4788168 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sdb: 256 MiB, 268435456 bytes, 524288 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sdc: 512 MiB, 536870912 bytes, 1048576 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x76e1587b

Device     Boot Start     End Sectors  Size Id Type
/dev/sdc1        2048 1048575 1046528  511M  c W95 FAT32 (LBA)
```

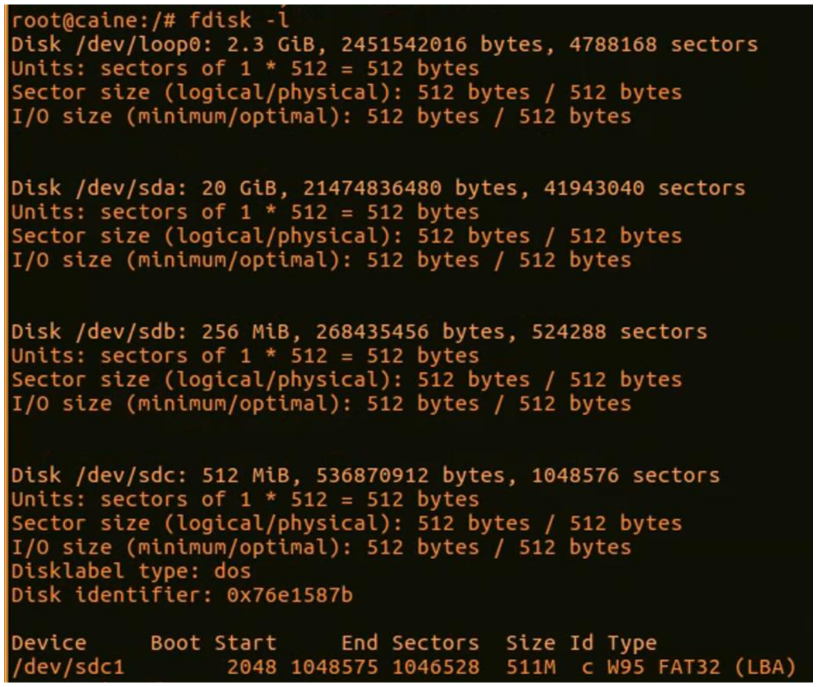
19. To format a FAT file system from Linux, type ***mkfs.msdos –vF32 /dev/sdc1***
    a. –v means verbose execution
    b. F32 means the type of file allocation table used is 32 bit (it can be 12, 16 or 32 bit).



```
root@caine:/# mkfs.msdos -vF32 /dev/sdc1
mkfs.fat 3.0.28 (2015-05-16)
/dev/sdc1 has 64 heads and 32 sectors per track,
hidden sectors 0x0800;
logical sector size is 512,
using 0xf8 media descriptor, with 1046528 sectors;
drive number 0x80;
filesystem has 2 32-bit FATs and 8 sectors per cluster.
FAT size is 1020 sectors, and provides 130557 clusters.
There are 32 reserved sectors.
Volume ID is ac63e73c, no volume label.
```

Part 2. Perform Data Acquisition

20. Type ***fdisk –l*** to show all disks. The next step is to perform data acquisition towards the evidence drive. In this case, /dev/sdb is the evidence drive.



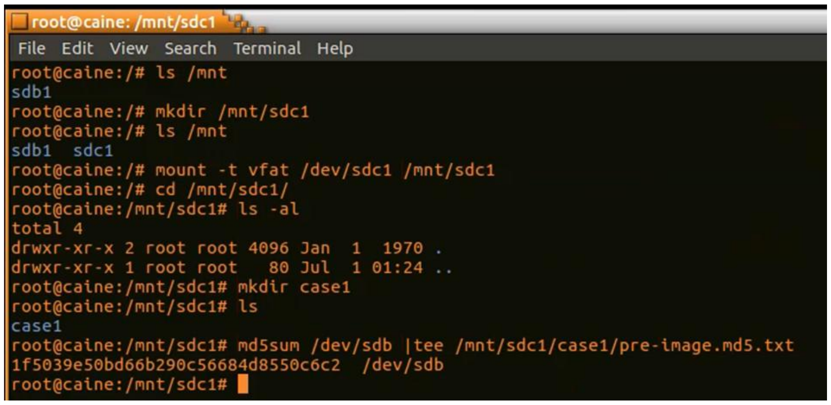
```
root@caine:/# fdisk -l
Disk /dev/loop0: 2.3 GiB, 2451542016 bytes, 4788168 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sdb: 256 MiB, 268435456 bytes, 524288 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sdc: 512 MiB, 536870912 bytes, 1048576 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x76e1587b

Device     Boot Start     End Sectors  Size Id Type
/dev/sdc1        2048 1048575 1046528  511M  c W95 FAT32 (LBA)
```

In order to store files to the target drive, we need to first mount the target drive to the operating system. The following steps will mount the target drive to /mnt folder.

21. Type ***ls /mnt*** to show files under /mnt.
22. Type ***mkdir /mnt/sdc1*** to create a folder under /mnt.
23. Type ***ls /mnt*** again to show files under /mnt.
24. Type ***mount –t vfat /dev/sdc1 /mnt/sdc1*** to mount the target drive partition.
    a. –t means the file system type is vfat.
25. Type cd ***/mnt/sdc1*** to change to default directory to target drive
26. Type ***ls –al*** to show the contents of the target drive's root level.
27. Type ***mkdir case1*** to create a target directory.
28. Type ***ls*** to confirm that the new directory has been created.

Next, we'll need to calculate the hash of the evidence drive /dev/sdb, so that later we can compare this hash with the hash of image after acquisition to see if they are the same. If they are the same, that means the acquisition is successful.

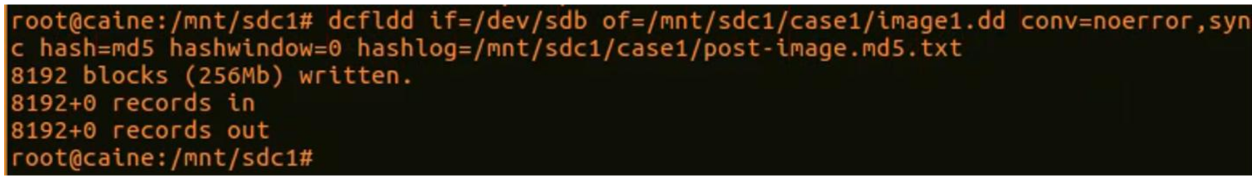
29. Type ***md5sum /dev/sdb |tee /mnt/sdc1/case1/pre-image.md5.txt***
    a. Md5sum calculate the hash of the evidence drive using MD5 algorithm
    b. |tee means the output is added to the txt file and also displayed in terminal



```
root@caine:/# ls /mnt
sdb1
root@caine:/# mkdir /mnt/sdc1
root@caine:/# ls /mnt
sdb1  sdc1
root@caine:/# mount -t vfat /dev/sdc1 /mnt/sdc1
root@caine:/# cd /mnt/sdc1/
root@caine:/mnt/sdc1# ls -al
total 4
drwxr-xr-x 2 root root 4096 Jan  1  1970 .
drwxr-xr-x 1 root root   80 Jul  1 01:24 ..
root@caine:/mnt/sdc1# mkdir case1
root@caine:/mnt/sdc1# ls
case1
root@caine:/mnt/sdc1# md5sum /dev/sdb |tee /mnt/sdc1/case1/pre-image.md5.txt
1f5039e50bd66b290c56684d8550c6c2  /dev/sdb
root@caine:/mnt/sdc1# 
```

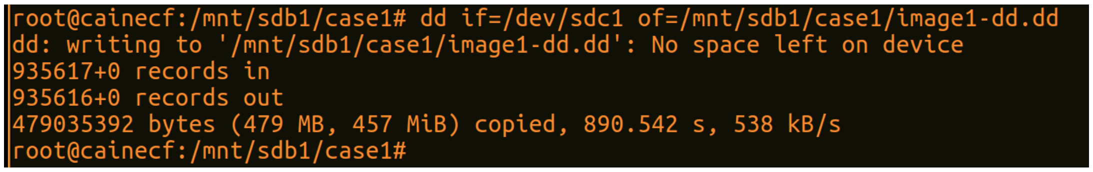
Now we can perform data acquisition.

30. To acquire data from the evidence drive /dev/sdb, type
    dcfldd if=/dev/sdb of=/mnt/sdc1/case1/image1.dd conv=noerror,sync hash=md5 hashwindow=0 hashlog=/mnt/sdc1/case1/post-image.md5.txt
    Special note here: to acquire the entire drive, if=/dev/sdb is correct. If you do if=/dev/sdb1, that means only acquire the sdb1 partition. This is not recommended in real investigation.



```
root@caine:/mnt/sdc1# dcfldd if=/dev/sdb of=/mnt/sdc1/case1/image1.dd conv=noerror,syn
c hash=md5 hashwindow=0 hashlog=/mnt/sdc1/case1/post-image.md5.txt
8192 blocks (256Mb) written.
8192+0 records in
8192+0 records out
root@caine:/mnt/sdc1#
```

    a. If you want to create segmented volumes of 2MB each, you may use (Please note: **This step is optional**. If you have already done acquisition using dcfldd, you cannot do this step again, otherwise you'll do acquisition twice.)
       dcfldd if=/dev/sdb split=2M of=/mnt/sdc1/case1/image1.dd conv=noerror,sync hash=md5 hashwindow=0 hashlog=/mnt/sdc1/case1/post-image.md5.txt
    b. If use dd command for data acquisition, the command should be (Please note: **This step is optional**. If you have already done acquisition using dcfldd, you **can NOT** do this step again, otherwise you'll do acquisition twice.)
       dd if=/dev/sdb |split -b 2m – image_sdc
       if you want to create multiple segments.
       Otherwise you can simply use
       dd if=/dev/sdb of=/mnt/sdc1/case1/image1-dd.dd
       In my demo, my target drive does not have enough disk storage to store the target image. You can read the number of blocks and bytes copied.

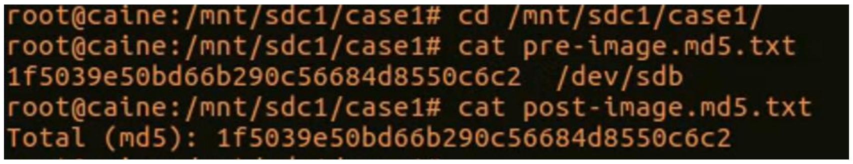


```
root@cainecf:/mnt/sdb1/case1# dd if=/dev/sdc1 of=/mnt/sdb1/case1/image1-dd.dd
dd: writing to '/mnt/sdb1/case1/image1-dd.dd': No space left on device
935617+0 records in
935616+0 records out
479035392 bytes (479 MB, 457 MiB) copied, 890.542 s, 538 kB/s
root@cainecf:/mnt/sdb1/case1#
```

Part 3. Validate the acquired data

31. To validate the acquired data, there are two ways if using *dcfldd* for data acquisition (Please note: only choose one way below. **You don't need to do both**. First way is recommended):
    a. Go to the /case1 folder by
    cd case1
    Type
    cat pre-image.md5.txt
    to show the hash value of original evidence drive;
    The hash of the acquired image is already computed and put into post-image.md5.txt.
    Type
    cat post-image.md5.txt
    to show the hash value and then compare the value in pre-image.md5.txt

    If pre-image.md5.txt and post-image.md5.txt have the same value, that means the acquisition is successful.

    ```
    root@caine:/mnt/sdc1/case1# cd /mnt/sdc1/case1/
    root@caine:/mnt/sdc1/case1# cat pre-image.md5.txt
    1f5039e50bd66b290c56684d8550c6c2  /dev/sdb
    root@caine:/mnt/sdc1/case1# cat post-image.md5.txt
    Total (md5): 1f5039e50bd66b290c56684d8550c6c2
    ```

    b. Type ***dcfldd if=/dev/sdb vf=/mnt/sdc1/case1/image1.dd*** (Note: this only applies to the nonsegmented image file).

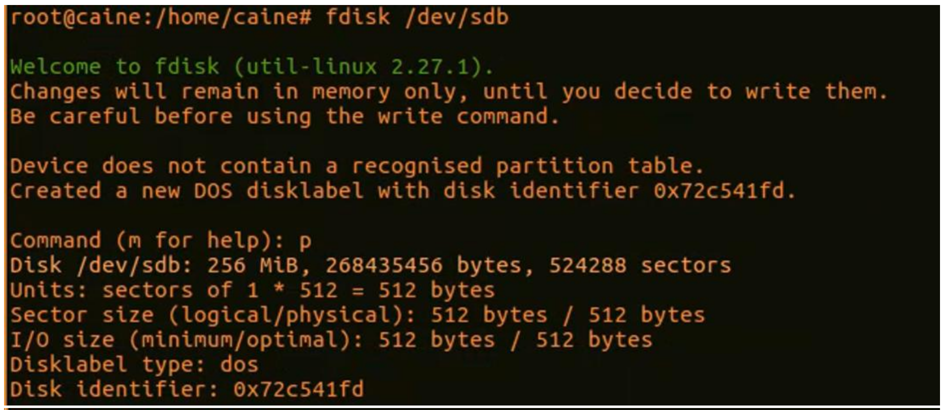
32. To validate the acquired data if using *dd* for data acquisition (**Please note: only use this if you have used dd to do data acquisition**):
    a. Type ***md5sum /dev/sdb |tee /mnt/sdc1/case1/pre-image.md5.txt***
    b. Type ***cat image_sdb.*|md5sum > post-image.md5.txt***
    And then compare the values in both files to see if they are the same. This applies to the segmented image file too.

The below steps are optional. If you are interested, you can continue working on it.

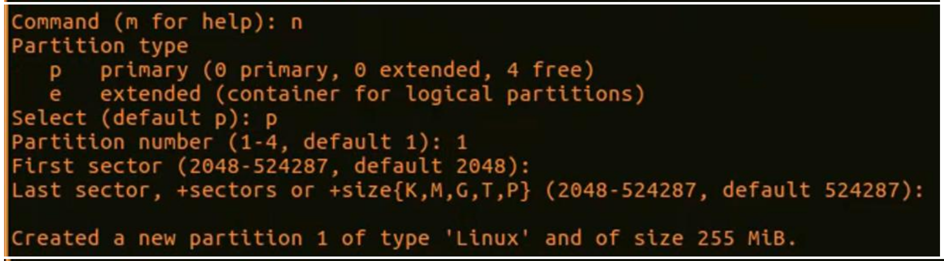
Currently the evidence drive is completely empty. If you want to add some data to your evidence drive (just for the purpose of this activity, you don't want to change your evidence drive in real investigation), you need to create a new partition on the evidence drive, create the file system, mount the drive, and then add some data onto it.

Please follow the steps below:

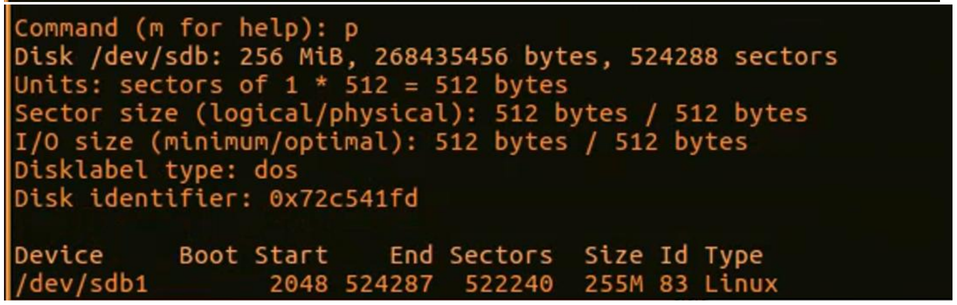
1. Create a new partition:
    a. Type fdisk /dev/sdb
    b. Type p
    c. Type n
    d. Type p
    e. Type 1
    f. Hit enter
    g. Hit enter
    h. Type p



```
root@caine:/home/caine# fdisk /dev/sdb

Welcome to fdisk (util-linux 2.27.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognised partition table.
Created a new DOS disklabel with disk identifier 0x72c541fd.

Command (m for help): p
Disk /dev/sdb: 256 MiB, 268435456 bytes, 524288 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x72c541fd
```

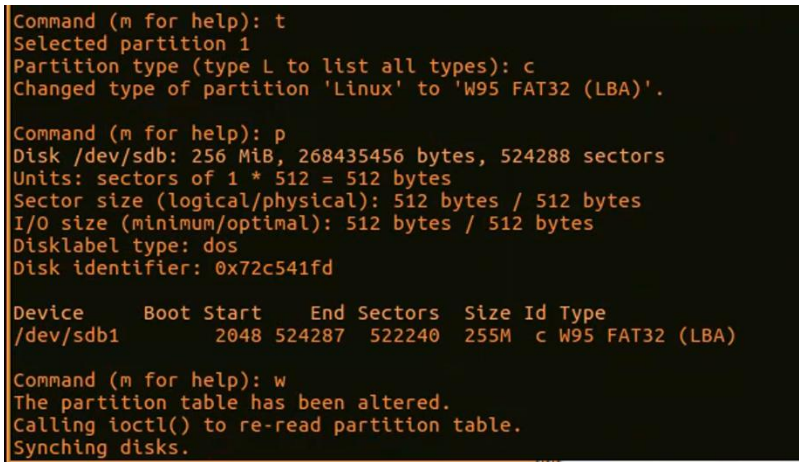


```
Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-524287, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-524287, default 524287):

Created a new partition 1 of type 'Linux' and of size 255 MiB.
```

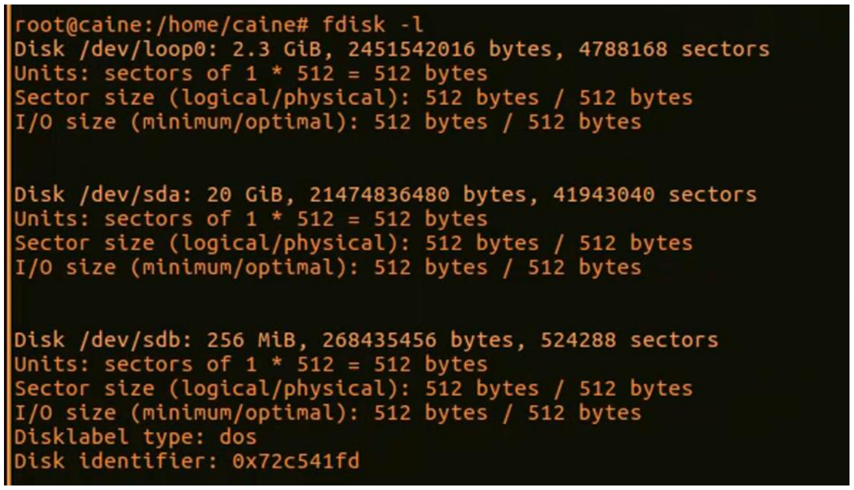


```
Command (m for help): p
Disk /dev/sdb: 256 MiB, 268435456 bytes, 524288 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x72c541fd

Device     Boot Start    End Sectors  Size Id Type
/dev/sdb1        2048 524287  522240  255M 83 Linux
```

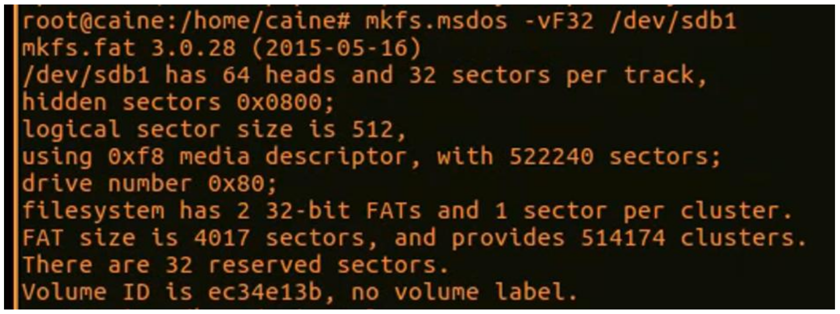
2. Create the file system
    a. Type t
    b. Type c
    c. Type p
    d. Type w
    e. Type fisk -l
    f. Type mkfs.msdos -vF32 /dev/sdb1



```
Command (m for help): t
Selected partition 1
Partition type (type L to list all types): c
Changed type of partition 'Linux' to 'W95 FAT32 (LBA)'.

Command (m for help): p
Disk /dev/sdb: 256 MiB, 268435456 bytes, 524288 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x72c541fd

Device     Boot Start    End Sectors  Size Id Type
/dev/sdb1        2048 524287  522240  255M  c W95 FAT32 (LBA)

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Synching disks.
```



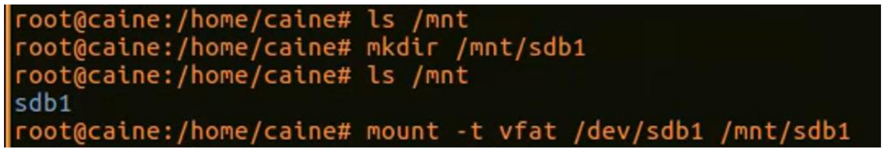
```
root@caine:/home/caine# fdisk -l
Disk /dev/loop0: 2.3 GiB, 2451542016 bytes, 4788168 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sda: 20 GiB, 21474836480 bytes, 41943040 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sdb: 256 MiB, 268435456 bytes, 524288 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x72c541fd
```



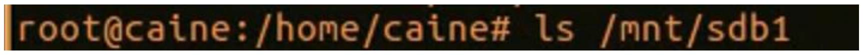
```
root@caine:/home/caine# mkfs.msdos -vF32 /dev/sdb1
mkfs.fat 3.0.28 (2015-05-16)
/dev/sdb1 has 64 heads and 32 sectors per track,
hidden sectors 0x0800;
logical sector size is 512,
using 0xf8 media descriptor, with 522240 sectors;
drive number 0x80;
filesystem has 2 32-bit FATs and 1 sector per cluster.
FAT size is 4017 sectors, and provides 514174 clusters.
There are 32 reserved sectors.
Volume ID is ec34e13b, no volume label.
```

3. Mount the drive
   a. Type ls /mnt
   b. Type mkdir /mnt/sdb1
   c. Type ls /mnt
   d. Type **mount -t vfat /dev/sdb1 /mnt/sdb1** to mount the /dev/sdb1 to /mnt/sdb1
   e. Type ls /mnt/sdb1
   f. Type cd /mnt/sdb1
   g. Type ls
   h. Type touch test.txt

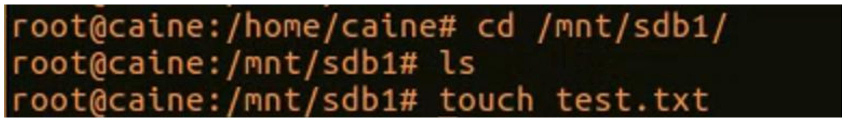
i. The above commands mounted the /dev/sdb1 to /mnt/sdb1, and then created a new file inside the /mnt/sdb1 folder. You can check the /mnt/sdb1 folder to validate if the new test.txt file is indeed created.



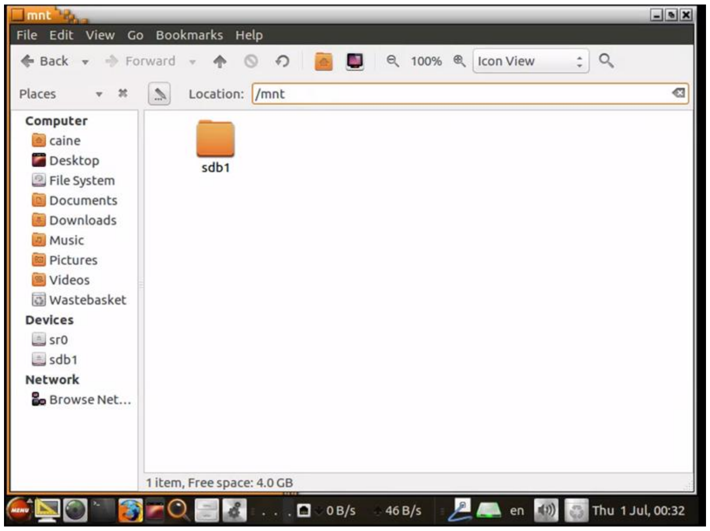
```
root@caine:/home/caine# ls /mnt
root@caine:/home/caine# mkdir /mnt/sdb1
root@caine:/home/caine# ls /mnt
sdb1
root@caine:/home/caine# mount -t vfat /dev/sdb1 /mnt/sdb1
```



```
root@caine:/home/caine# ls /mnt/sdb1
```



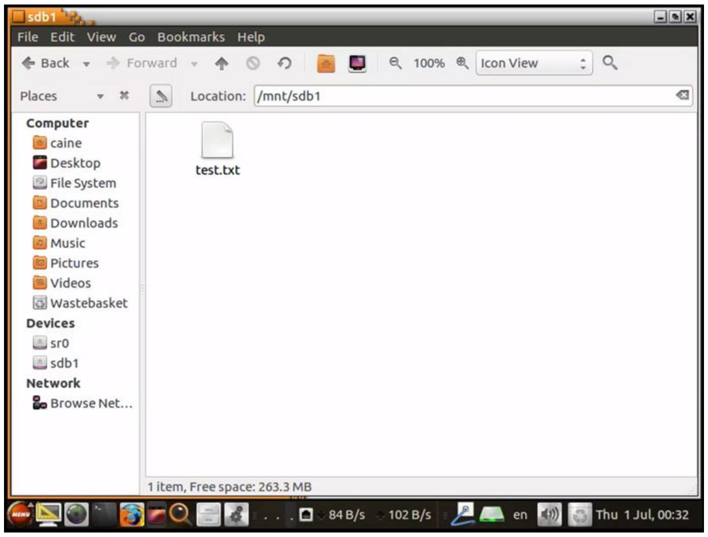
```
root@caine:/home/caine# cd /mnt/sdb1/
root@caine:/mnt/sdb1# ls
root@caine:/mnt/sdb1# touch test.txt
```



Location: /mnt

sdb1

1 item, Free space: 4.0 GB



Location: /mnt/sdb1

test.txt

1 item, Free space: 263.3 MB

Post-Activity questions.

Please complete this exercise and submit the PDF to Canvas.

1. What are the two broad categories of acquisition?
2. What is a live storage acquisition and when is it used?
3. Which command should be used to check the disks available on the current system? You only need to state the command name, not the entire command string.
4. The mkfs command does what?
5. Which drive should be 'zeroed out', the source evidence drive or the target drive?
6. What is the purpose of 'zeroing out' before a storage acquisition is performed?
7. When you issue the command the command
   dd if=/dev/zero of=/dev/sdc
   What does the string "/dev/sdc" represent?
8. The md5sum /dev/sdb command does what? Why is it used?
9. How many times should the ***hash*** be calculated at least in one acquisition? When should they be calculated?
10. Instead of using “dd”, what other commands can you use to perform data acquisition in Linux?