Computer Forensics

Chapter 3 - Data Acquisition

Introduction

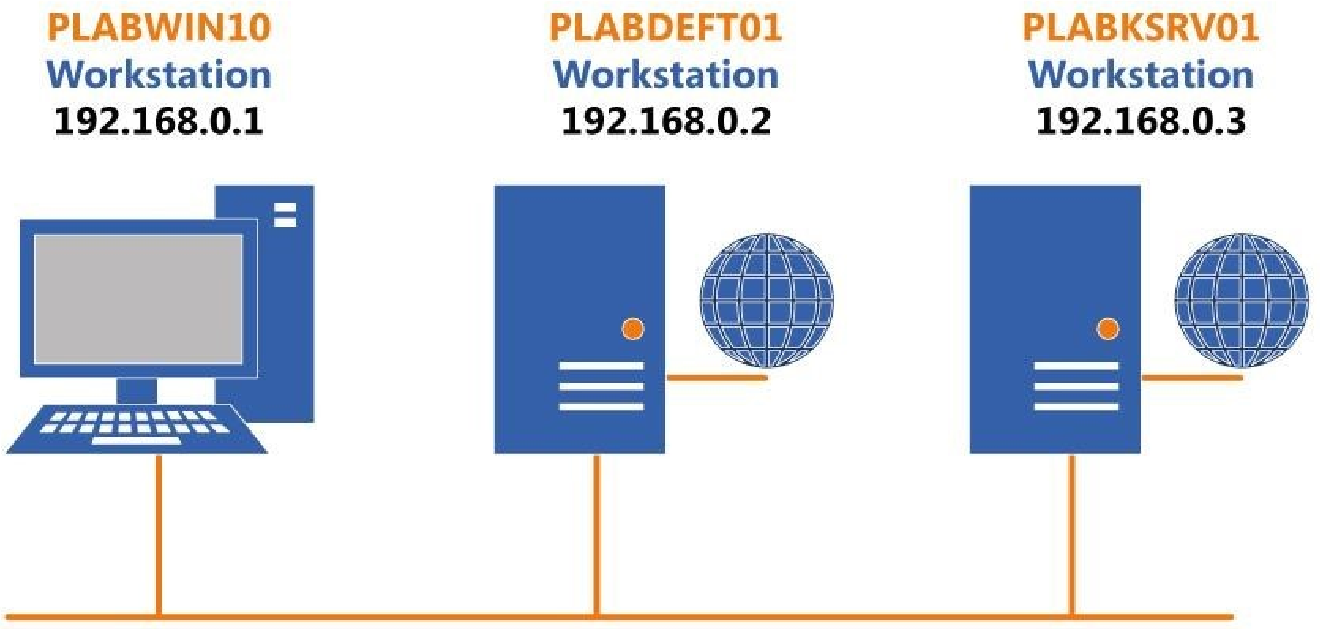
The **Data Acquisition** module provides you with the instructions and devices to develop your hands-on skills in the following topics:

Exam Objectives

- Exercise 3-1 Preparing a Target Drive for Acquisition in Linux
- Exercise 3-2 Acquiring Data with dd in Linux
- Hands-On Project 3-3
- Hands-On Project 3-4

Lab Diagram

During your session, you will have access to the following lab configuration. Depending on the exercises, you may or may not use all of the devices, but they are shown here in the layout to get an overall understanding of the topology of the lab.



Connecting to your Lab

In this module, you will be working on the following equipment to carry out the steps defined in each exercise.

- **PLABWIN10** (Windows 10 - Standalone Workstation)
- **PLABKSRV01** (Kali Linux Workstation)

Help and Support

For more information on using Practice Labs, please see our **Help and Support** page. You can also raise a technical support ticket from this page.

Click Next to proceed to the first exercise.

Exercise 3-1 - Preparing a Target Drive for Acquisition in Linux

The Linux OS has many tools you can use to modify non-Linux file systems. Current Linux distributions can create Microsoft File Allocation Table (FAT) and New Technology File System (NTFS) partition tables. Linux kernel version 2.6.17.7 and earlier can format and read only the FAT file system, although an NTFS driver, NTFS-3G, is available that allows Linux to mount and write data only to NTFS partitions. You can download this driver from http://sourceforge.net/projects/ntfs-3g, where you can also find information about NTFS and instructions for installing the driver. For information on Mac OS X file systems and acquisitions, see Chapter 7.

Please refer to your course material or use your preferred search engine to research this topic in more detail.

Task 1 - Prepare Linux File System

In this task, you learn how to partition and format a Microsoft FAT drive from Linux so that you don't have to switch OSs or computers to prepare a FAT target disk. If you have a previously used target drive, you can use the following procedure to format it as a FAT32 drive. After you make the acquisition, you can then transfer the FAT disk to a Windows system to use a Windows analysis tool.

When preparing a drive to be used on a Linux system for forensics acquisition or analysis, do it in a separate boot session with no suspect drive attached.

Assuming you have a functioning Linux computer or one running with a Linux Live CD, perform the following steps from a shell prompt or GUI-based application like GParted.

To prepare a Windows FAT32 file system, perform the following steps:

Step 1

Ensure that you have powered on the required devices indicated in the Introduction.

Connect to **PLABKSRV01** device.

Step 2

Log into KALI with the following credentials:

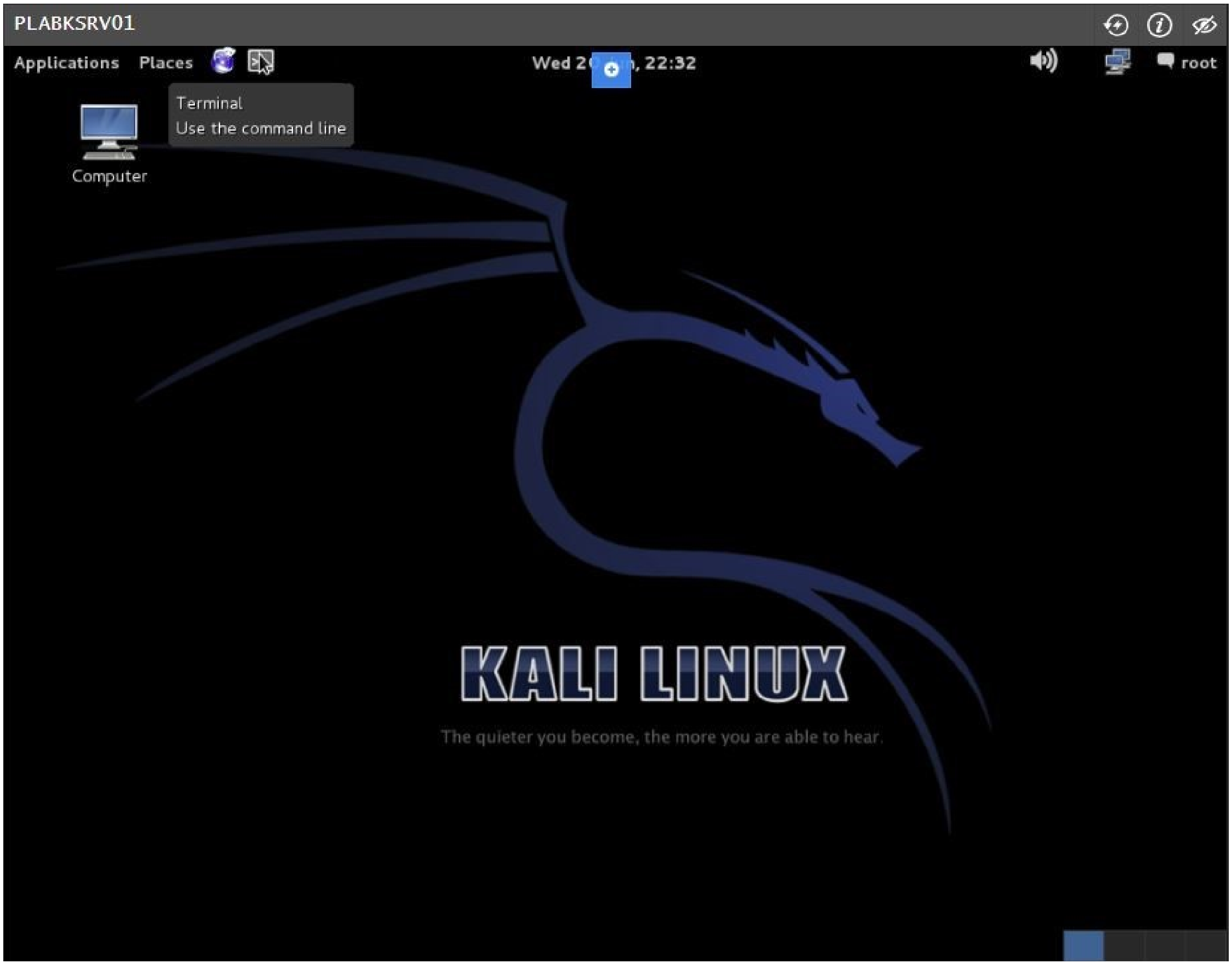
Username:

```
root
```

Password:

```
Passw0rd
```

When logged on to KALI Linux, locate the top menu, click **Terminal** icon.



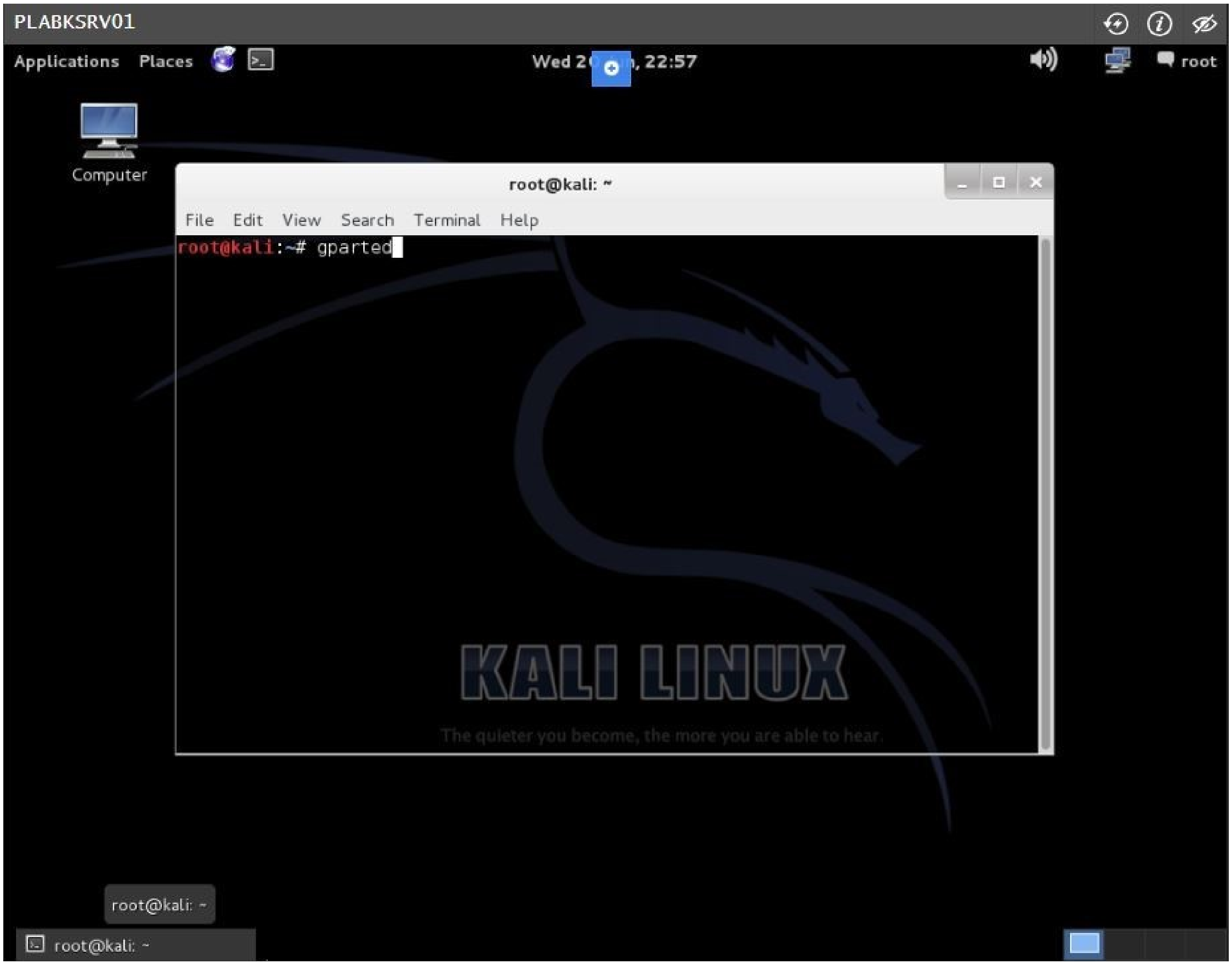
Step 3

The prompt changes to **root@kali**:~#.

To use a graphical user interface application for preparing disk storage, type:

```
gparted
```

Press **Enter**.

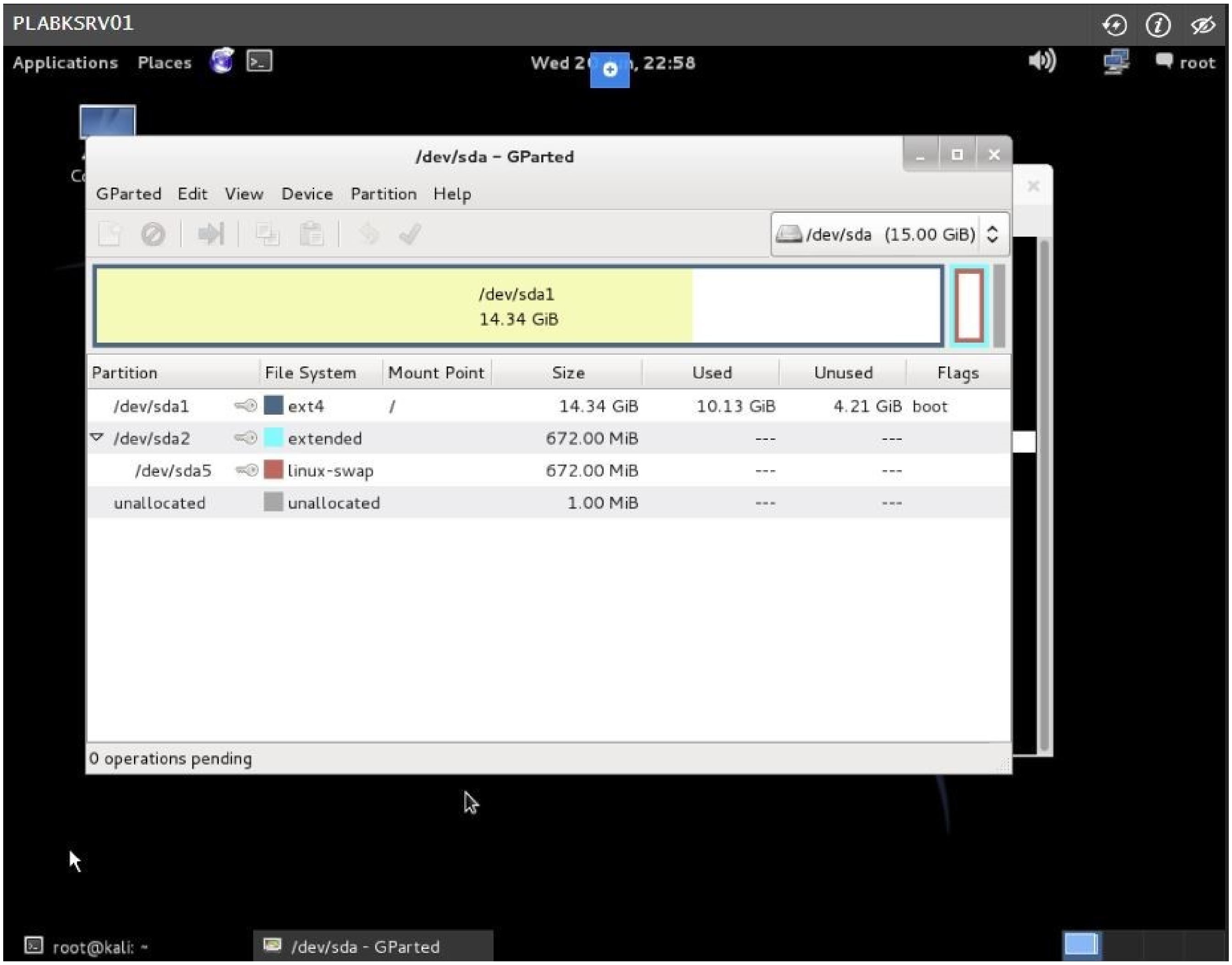


```
root@kali:~# gparted
```

Step 4

The /**dev/sda - GParted** window is open.

Information about the partitioning details of selected device **/dev/sda** is displayed in the middle pane.

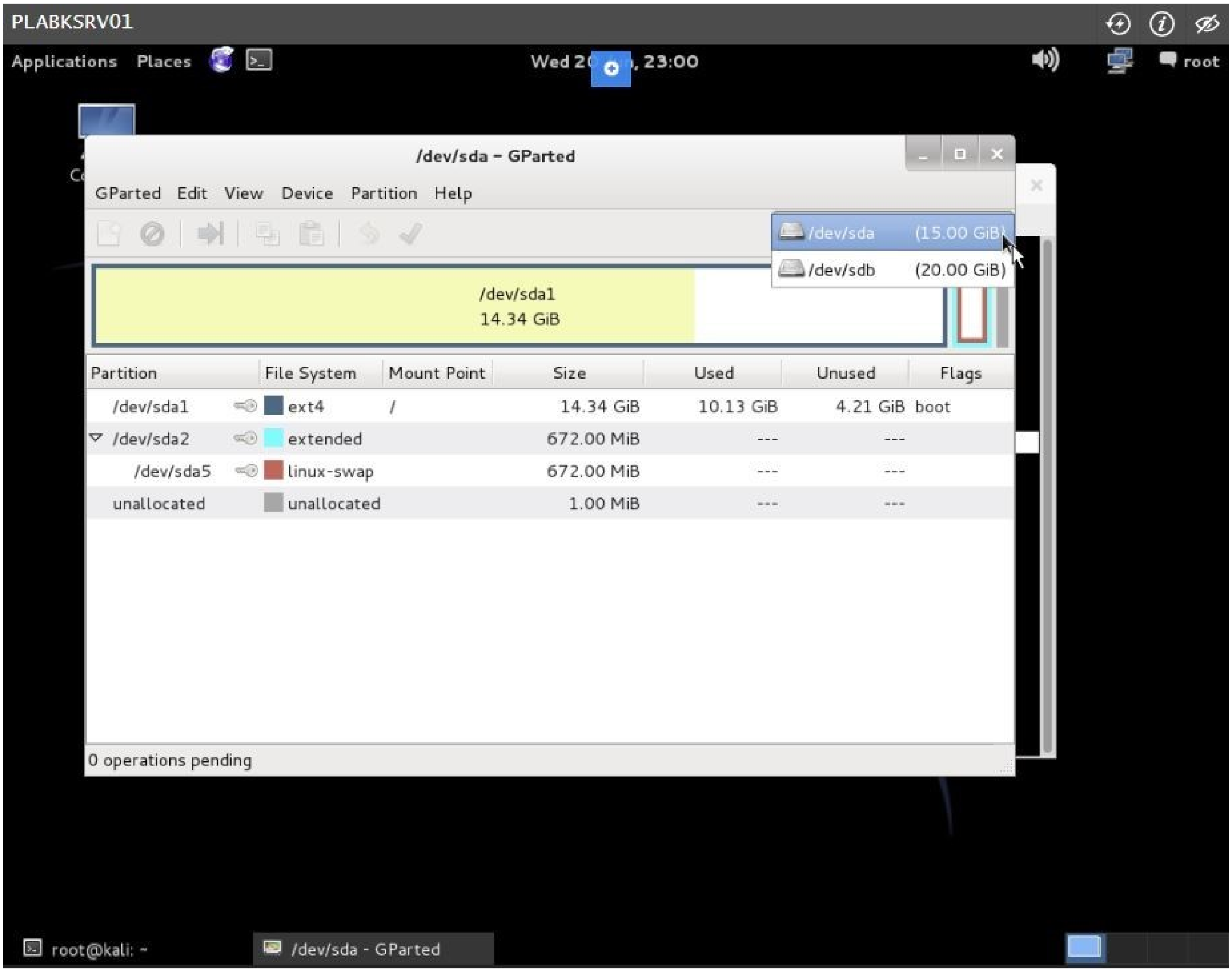


Partition	File System	Mount Point	Size	Used	Unused	Flags
/dev/sda1	ext4	/	14.34 GiB	10.13 GiB	4.21 GiB	boot
/dev/sda2	extended		672.00 MiB	---	---	
/dev/sda5	linux-swap		672.00 MiB	---	---	
unallocated	unallocated		1.00 MiB	---	---	

0 operations pending

Step 5

On the far right corner, access the drop-down list and change it to **/dev/sdb**.



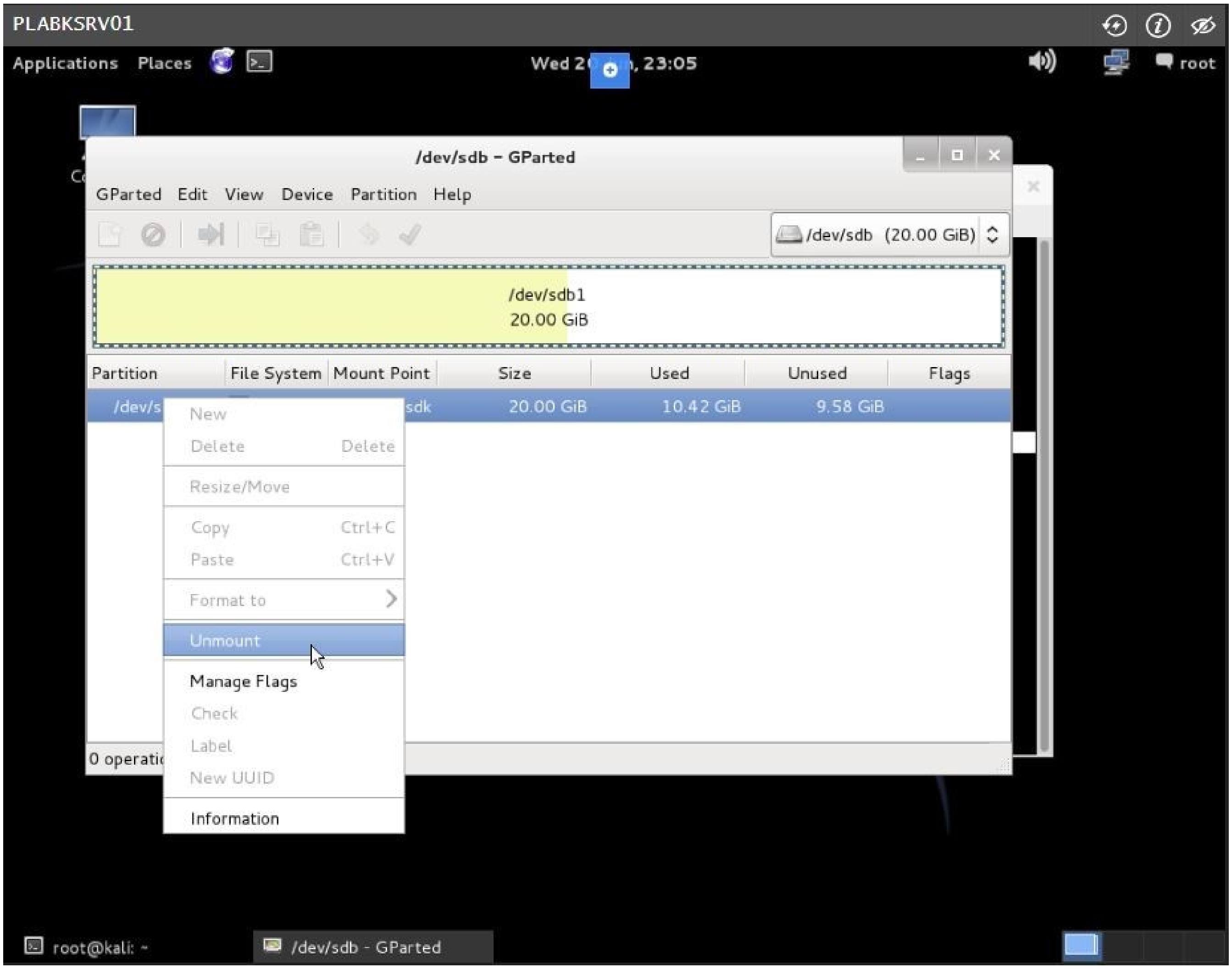
Partition	File System	Mount Point	Size	Used	Unused	Flags
/dev/sda1	ext4	/	14.34 GiB	10.13 GiB	4.21 GiB	boot
/dev/sda2	extended		672.00 MiB	---	---	
/dev/sda5	linux-swap		672.00 MiB	---	---	
unallocated	unallocated		1.00 MiB	---	---	

0 operations pending

Step 6

The selection changes to **/dev/sdb**. You will be using this disk to prepare the volume to use another system like **FAT32**.

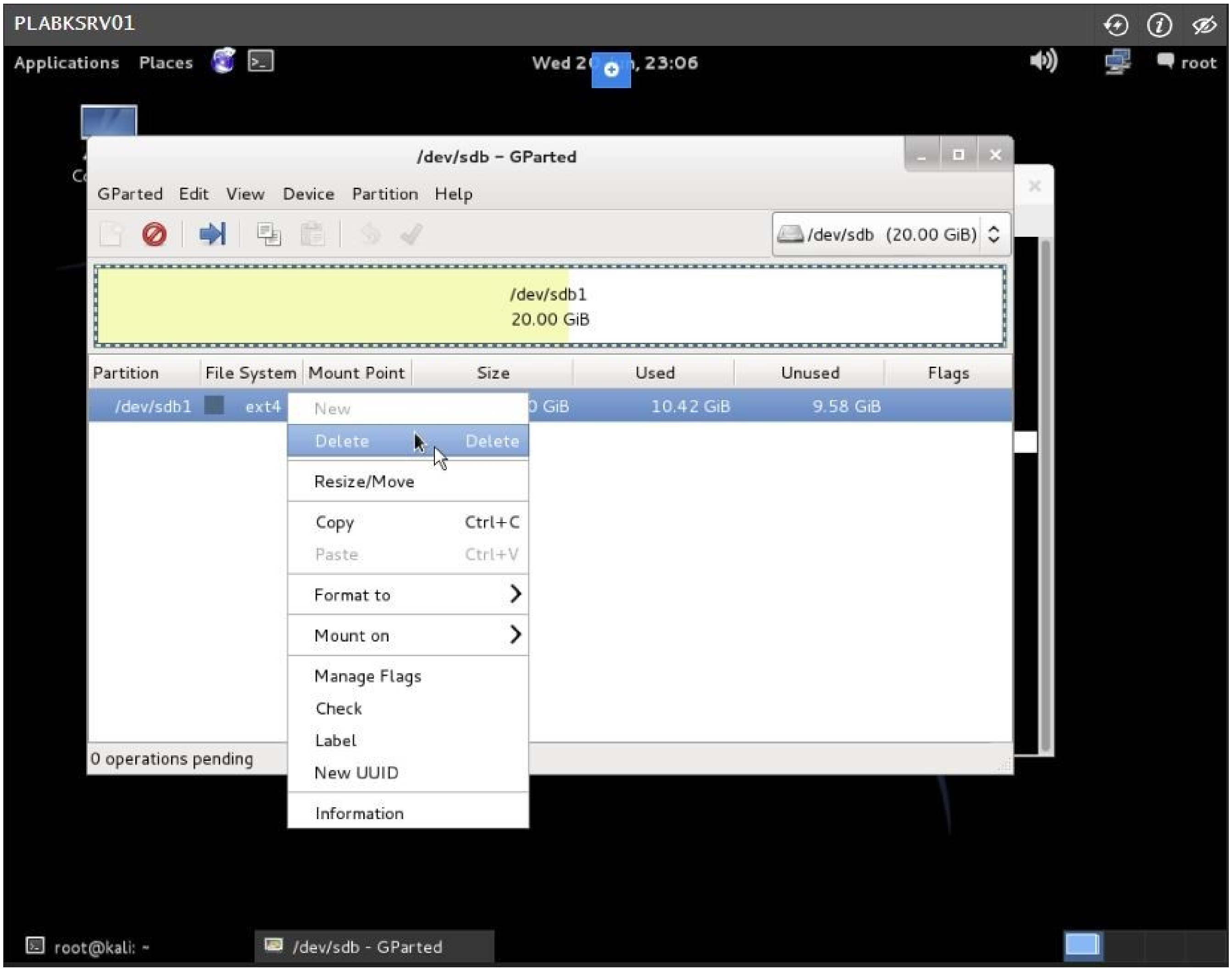
Right-click **/dev/sdb1** and select **Unmount**.



Partition	File System	Mount Point	Size	Used	Unused	Flags
/dev/s		sdk	20.00 GiB	10.42 GiB	9.58 GiB	

Step 7

Right-click **/dev/sdb1** and select **Delete**.



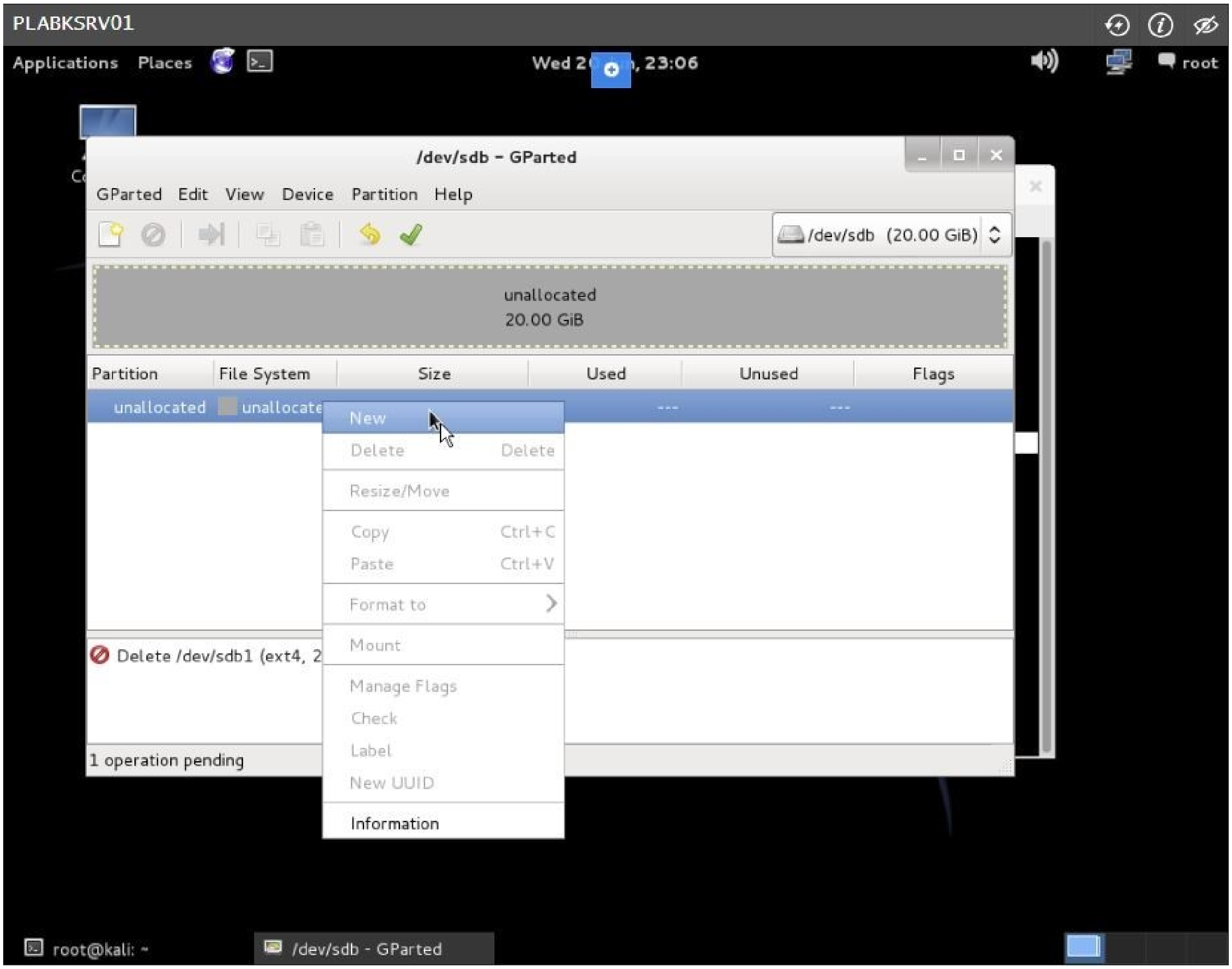
Partition	File System	Mount Point	Size	Used	Unused	Flags
/dev/sdb1	ext4		0 GiB	10.42 GiB	9.58 GiB	

0 operations pending

Step 8

After deleting the partition it becomes unallocated.

Right-click on unallocated and select **New**.



Delete /dev/sdb1 (ext4, 2

1 operation pending

Step 9

On the **Create new Partition** dialog box, access the **New size (MiB)** spin box and type:

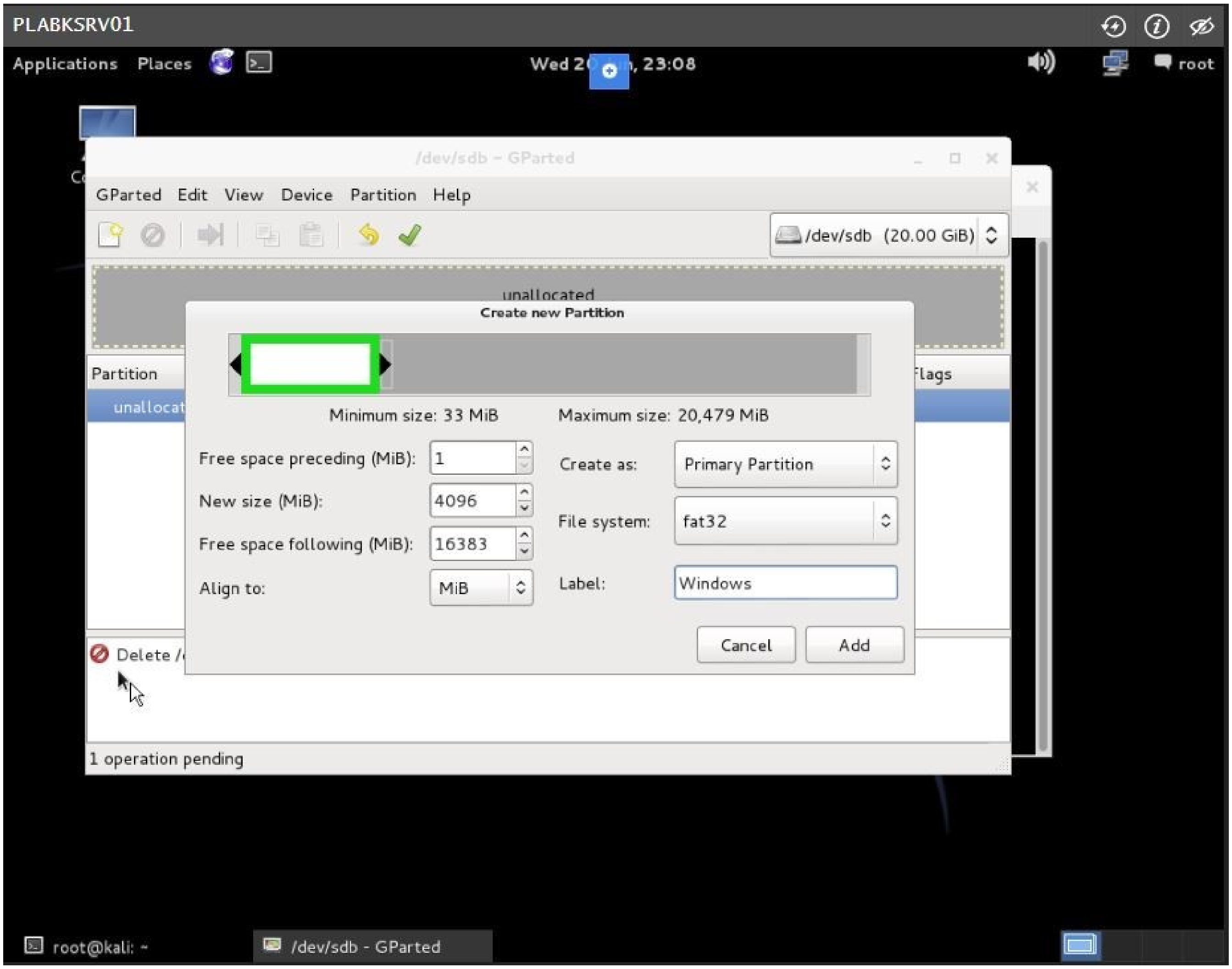
```
4096
```

On the **File system** spin box, select **fat32**.

On the **Label** text box, type:

```
Windows
```

Click **Add**.



Create new Partition

Minimum size: 33 MiB
Maximum size: 20,479 MiB

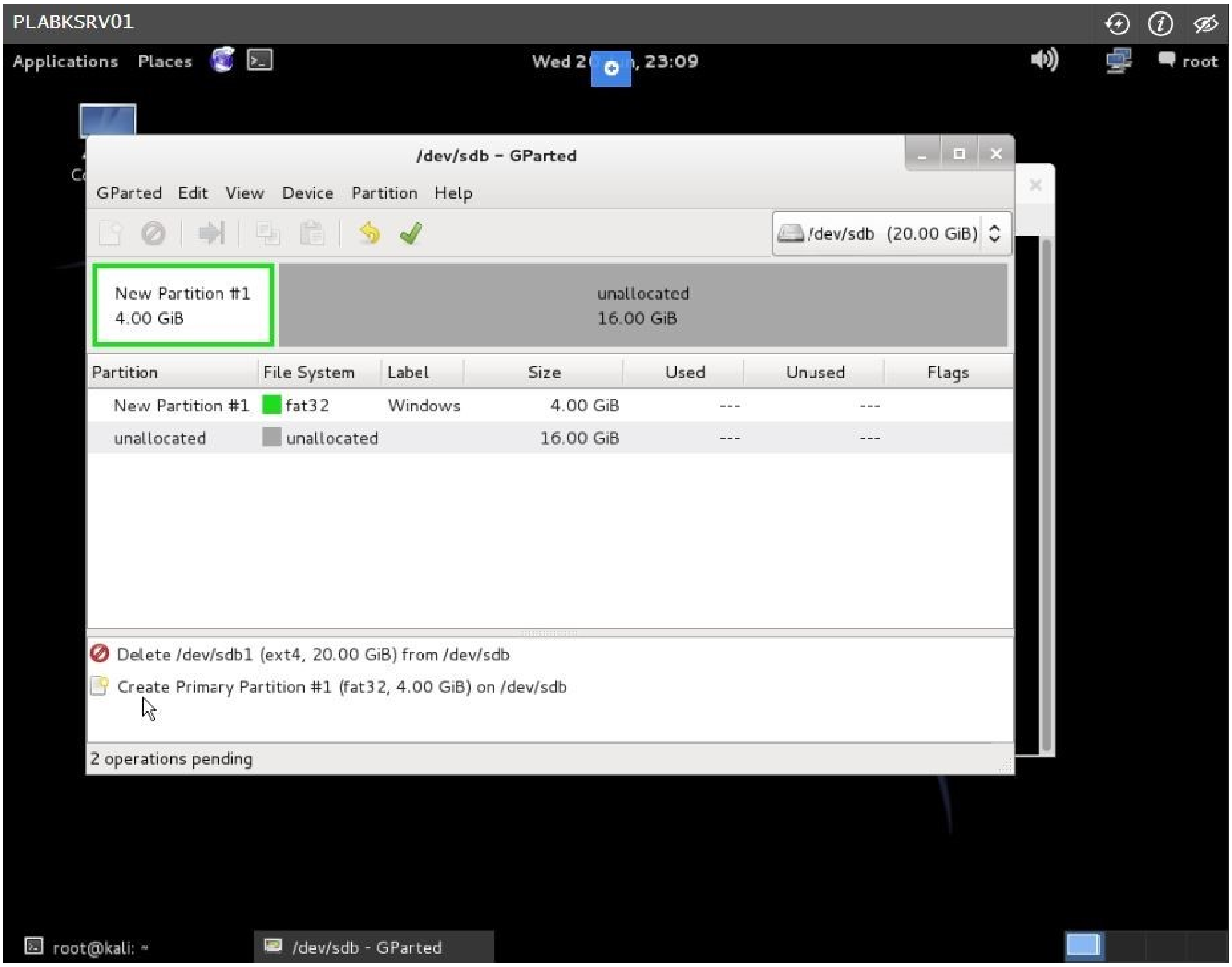
Free space preceding (MiB): 1
New size (MiB): 4096
Free space following (MiB): 16383
Align to: MiB

Create as: Primary Partition
File system: fat32
Label: Windows

1 operation pending

Step 10

The **New Partition #1** is now added. In order to save the changes made on the file system, click the green check or tick mark.



Partition	File System	Label	Size	Used	Unused	Flags
New Partition #1	🟩 fat32	Windows	4.00 GiB	---	---	
unallocated	▨ unallocated		16.00 GiB	---	---	

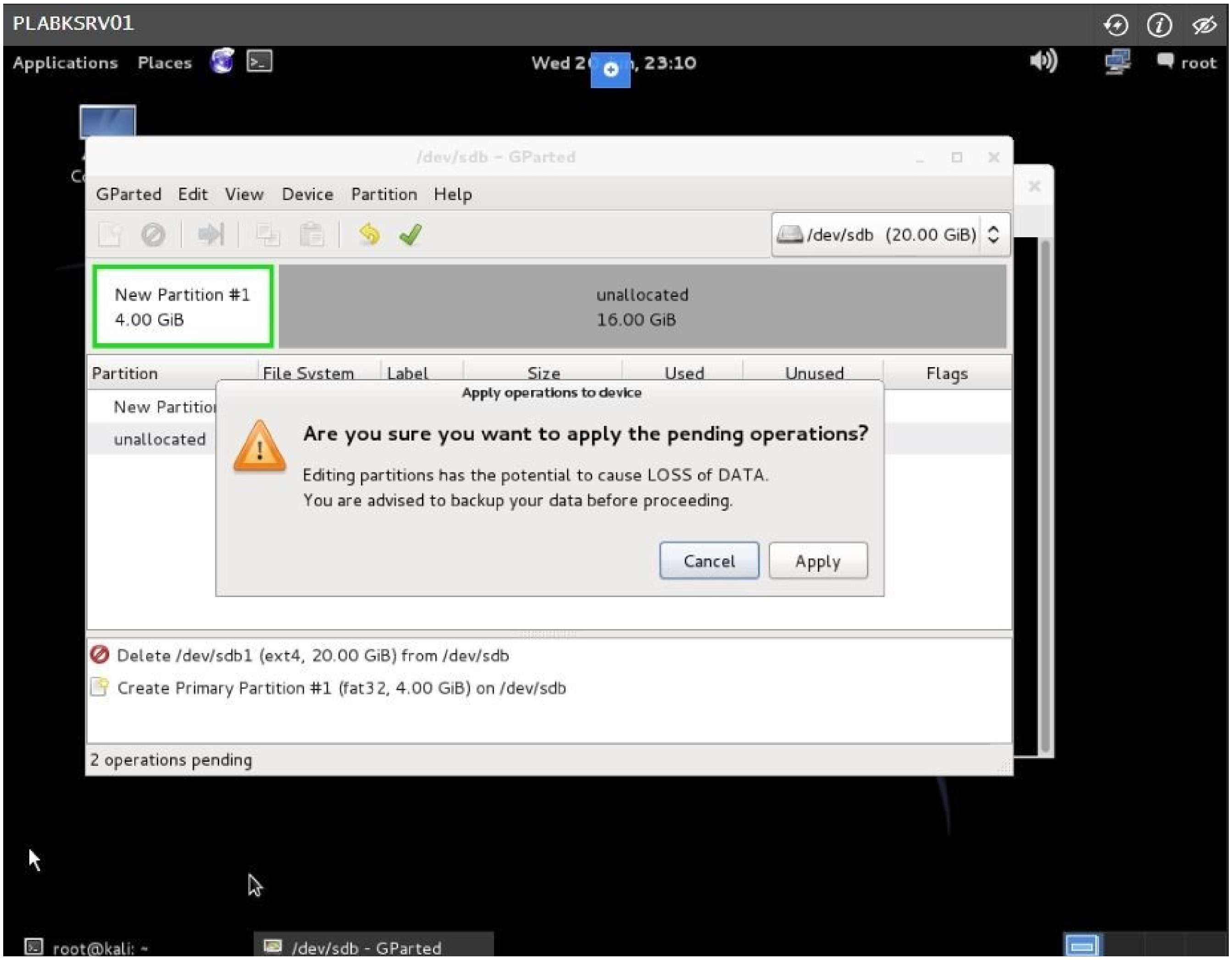
Delete /dev/sdb1 (ext4, 20.00 GiB) from /dev/sdb

Create Primary Partition #1 (fat32, 4.00 GiB) on /dev/sdb

2 operations pending

Step 11

Click **Apply** when asked if you want to apply pending operations.



Apply operations to device

Are you sure you want to apply the pending operations?

Editing partitions has the potential to cause LOSS of DATA.
You are advised to backup your data before proceeding.

Delete /dev/sdb1 (ext4, 20.00 GiB) from /dev/sdb

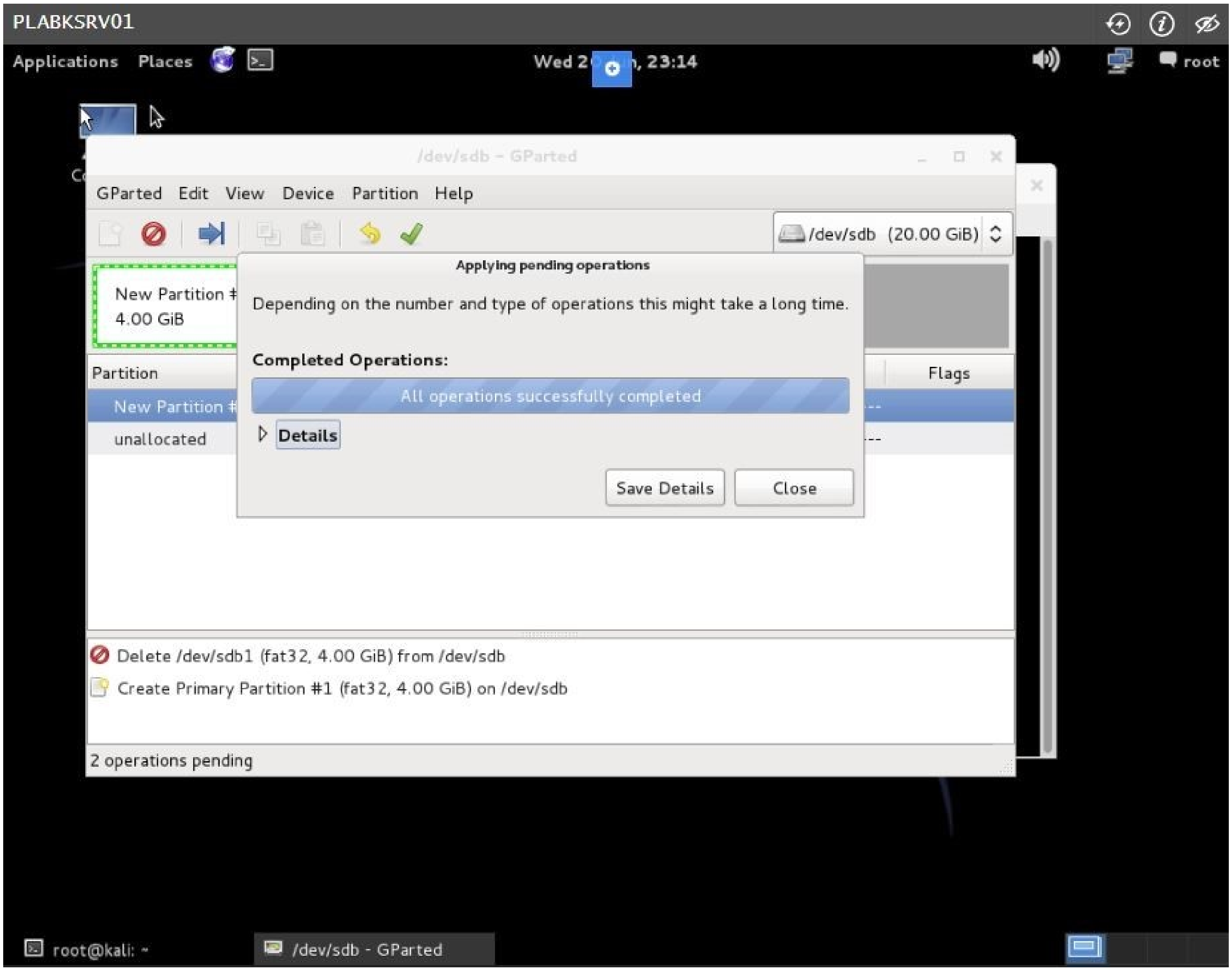
Create Primary Partition #1 (fat32, 4.00 GiB) on /dev/sdb

2 operations pending

Step 12

Please wait while changes are being committed.

Click **Close** when notified that the operations were successfully completed.



Applying pending operations

Depending on the number and type of operations this might take a long time.

Completed Operations:

All operations successfully completed

Delete /dev/sdb1 (fat32, 4.00 GiB) from /dev/sdb

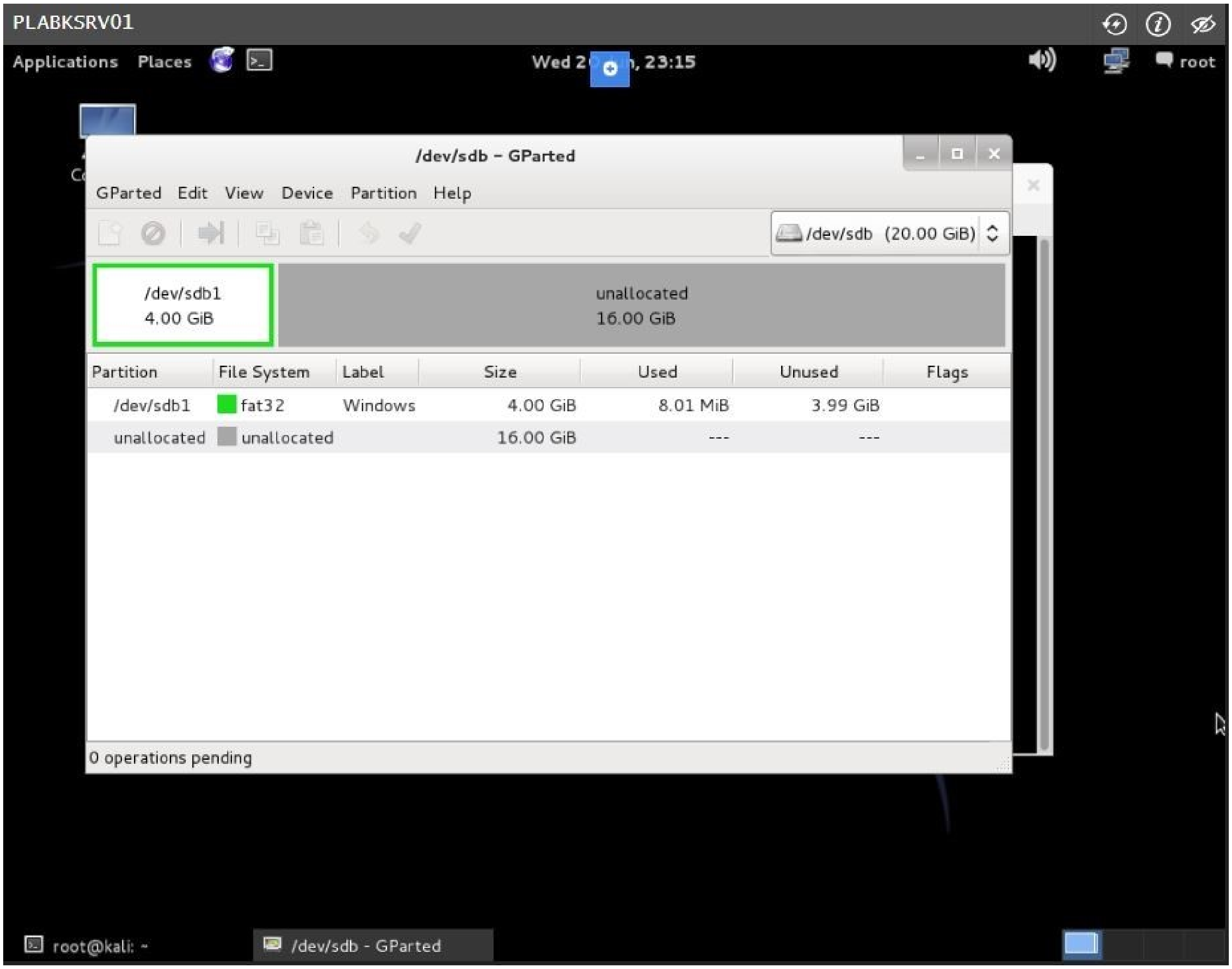
Create Primary Partition #1 (fat32, 4.00 GiB) on /dev/sdb

2 operations pending

Step 13

The new primary partition in **/dev/sdb1** is now added.

Close **GParted** window.



Partition	File System	Label	Size	Used	Unused	Flags
/dev/sdb1	🟩 fat32	Windows	4.00 GiB	8.01 MiB	3.99 GiB	
unallocated	▨ unallocated		16.00 GiB	---	---	

0 operations pending

Step 14

You are redirected to the terminal window.

Keep this terminal window open.

Keep all devices powered on in their current state and proceed to the next task.

Task 2 - Mounting a disk volume

To mount the partition that you created earlier, perform the following steps:

Step 1

You are connected to **PLABKSRV01**.

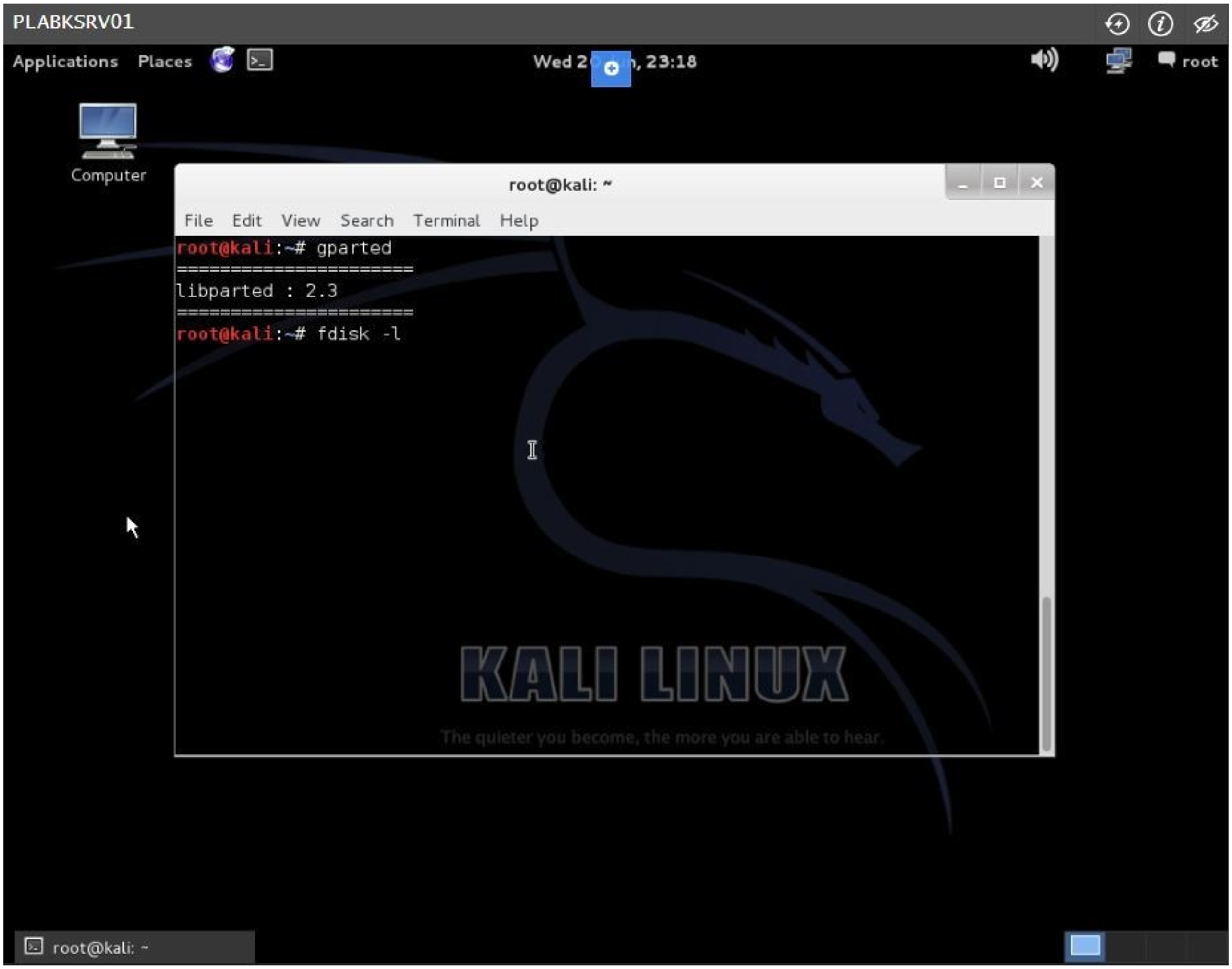
The Terminal window is open.

To find out what name your device file have, run the following command:

```
fdisk -l
```

Where “l” is lower case L.

Press **Enter**.

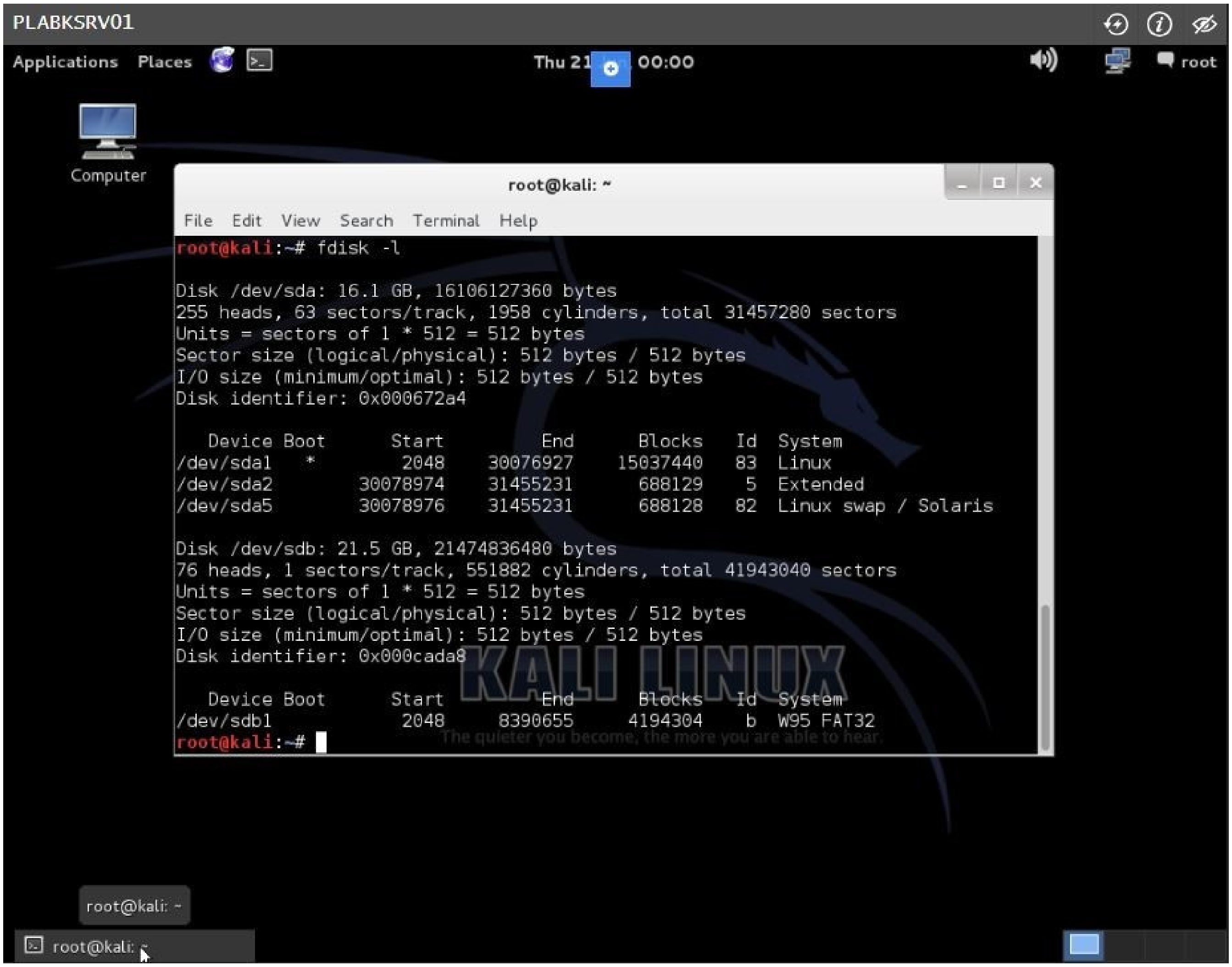


```
root@kali:~# gparted
======================
libparted : 2.3
======================
root@kali:~# fdisk -l
```

Step 2

On the results of the “fdisk -l” command, locate the section, **/dev/sdb**.

Notice the **Start**, **End**, **Blocks**, **Id** and **System** about this partition.



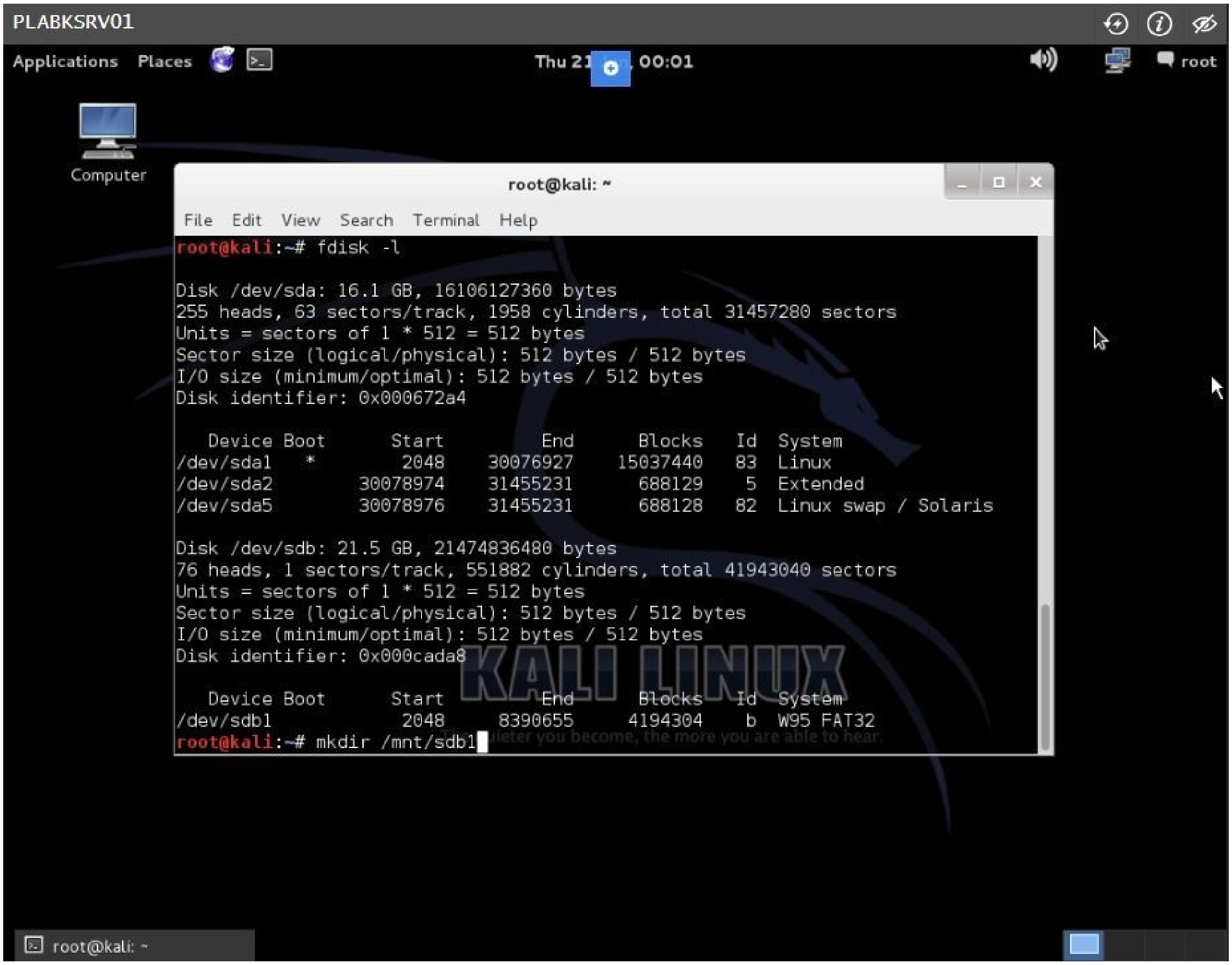
```
root@kali:~# fdisk -l

Disk /dev/sda: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders, total 31457280 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000672a4

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048    30076927    15037440   83  Linux
/dev/sda2        30078974    31455231      688129    5  Extended
/dev/sda5        30078976    31455231      688128   82  Linux swap / Solaris

Disk /dev/sdb: 21.5 GB, 21474836480 bytes
76 heads, 1 sectors/track, 551882 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000cada8

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048     8390655     4194304    b  W95 FAT32
root@kali:~# 
```

Step 3

To create a directory where you will mount the device, type:

```
mkdir /mnt/sdb1
```

Press **Enter**.



```
root@kali:~# fdisk -l

Disk /dev/sda: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders, total 31457280 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000672a4

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048    30076927    15037440   83  Linux
/dev/sda2        30078974    31455231      688129    5  Extended
/dev/sda5        30078976    31455231      688128   82  Linux swap / Solaris

Disk /dev/sdb: 21.5 GB, 21474836480 bytes
76 heads, 1 sectors/track, 551882 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000cada8

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048     8390655     4194304    b  W95 FAT32
root@kali:~# mkdir /mnt/sdb1
```

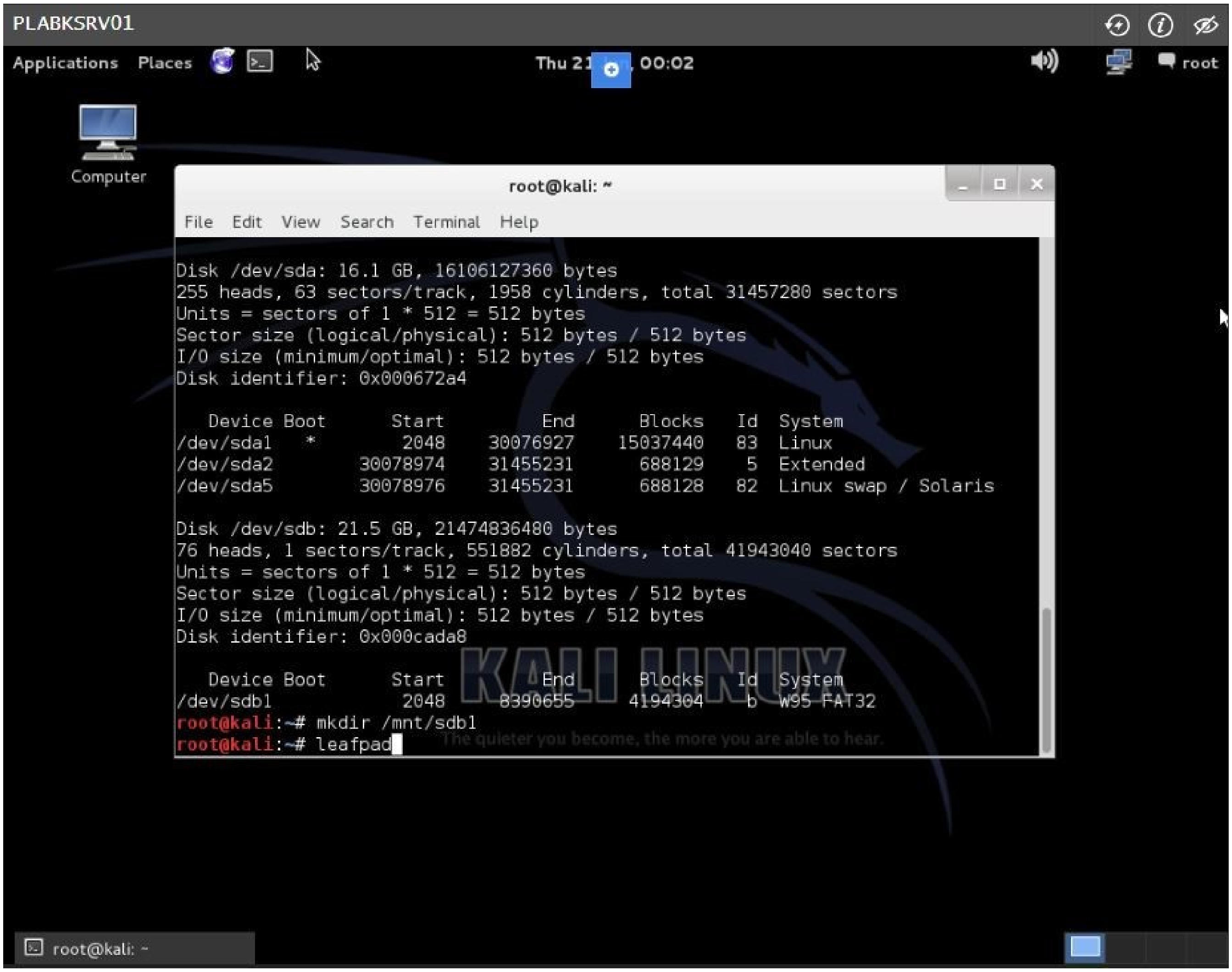
Step 4

After creating “**mnt**” directory, on the next prompt, you will edit the “**/etc/fstab**” file.

On the next prompt, type:

```
leafpad
```

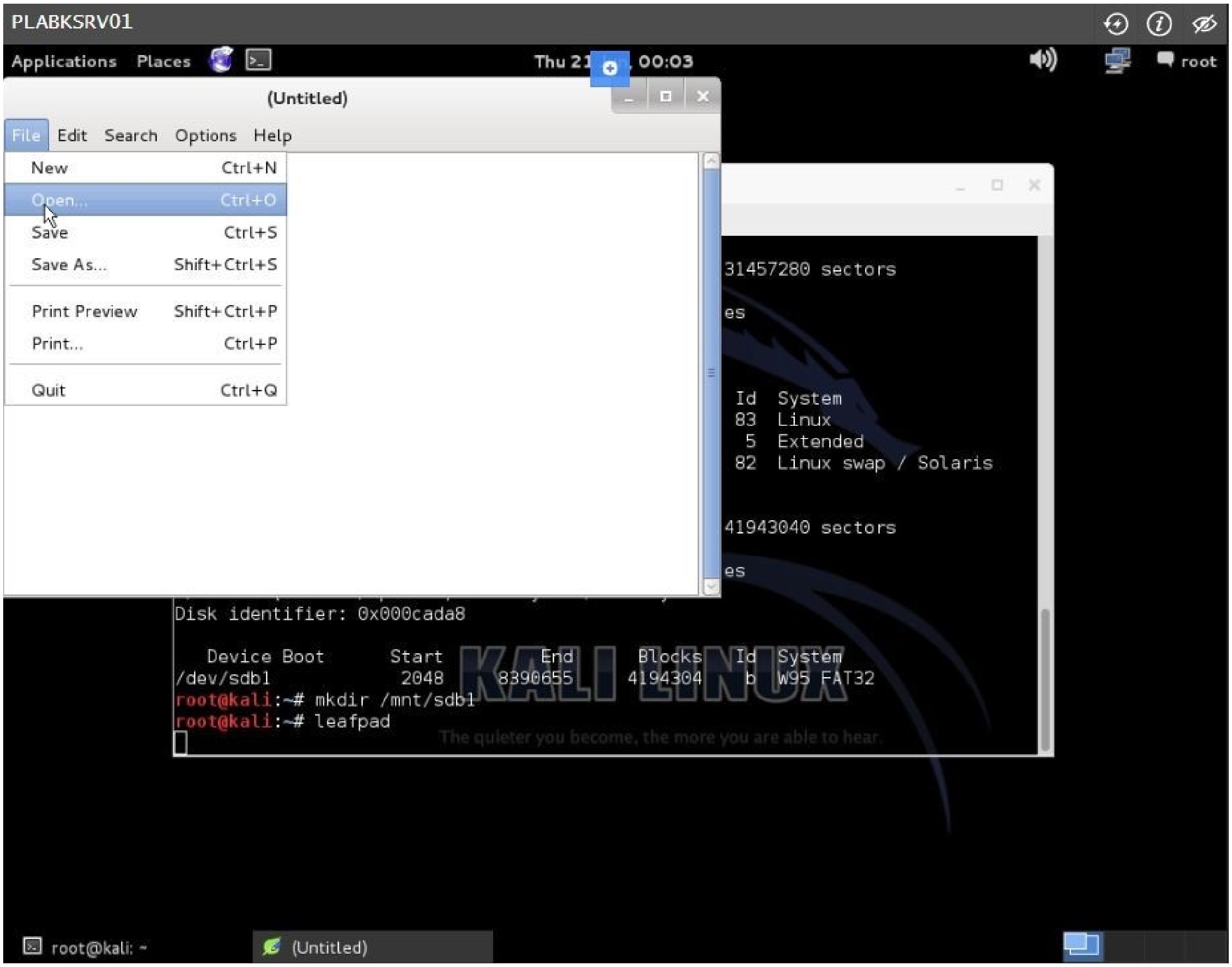
Press **Enter**.



```
Disk /dev/sda: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders, total 31457280 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000672a4

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048    30076927    15037440   83  Linux
/dev/sda2        30078974    31455231      688129    5  Extended
/dev/sda5        30078976    31455231      688128   82  Linux swap / Solaris

Disk /dev/sdb: 21.5 GB, 21474836480 bytes
76 heads, 1 sectors/track, 551882 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000cada8

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048     8390655     4194304    b  W95 FAT32
root@kali:~# mkdir /mnt/sdb1
root@kali:~# leafpad
```

Step 5

When **Leafpad** application opens, click **File** then select **Open**.



```
31457280 sectors
es

 Id  System
 83  Linux
  5  Extended
 82  Linux swap / Solaris

41943040 sectors
es
Disk identifier: 0x000cada8

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048     8390655     4194304    b  W95 FAT32
root@kali:~# mkdir /mnt/sdb1
root@kali:~# leafpad
```

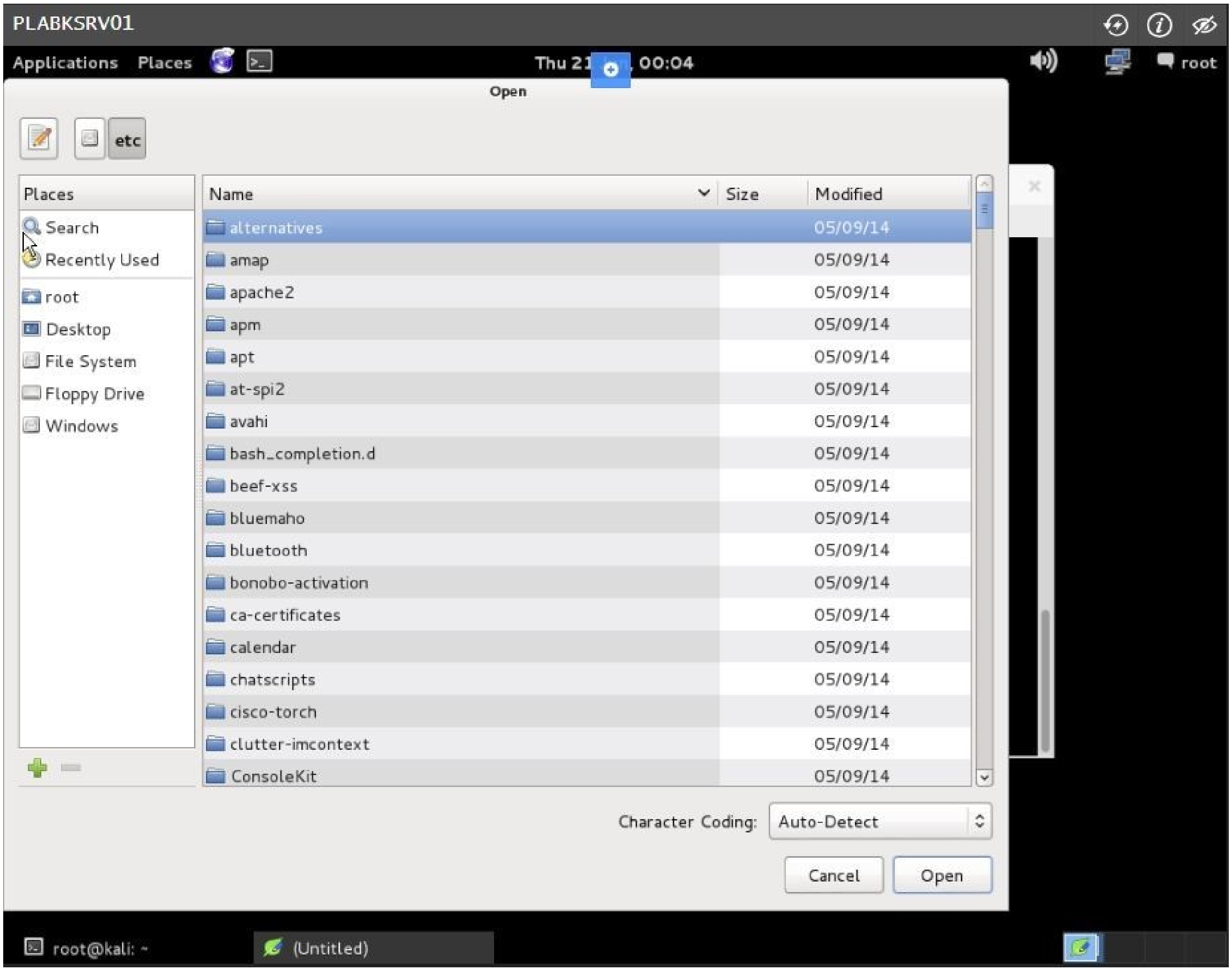
Step 6

On the **Open** dialog box, click **File System** on the left pane.

Then double-click **etc** folder on the right pane.

When **etc** folder opens, scroll down and locate the file called **fstab**.

Please be aware that the fstab file is not located in the folder fstab.d and instead found at the bottom of the etc folder.



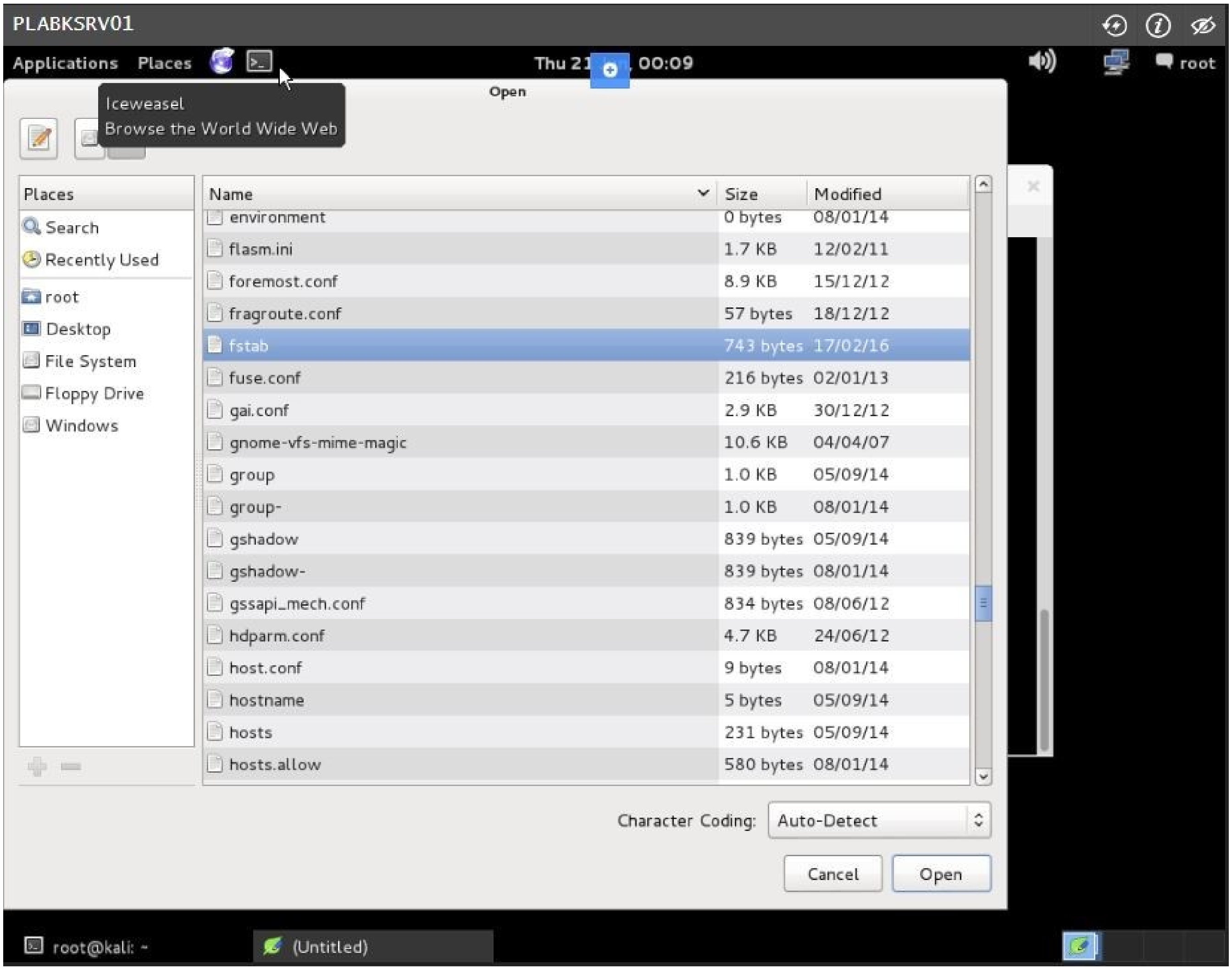
Name	Size	Modified
alternatives		05/09/14
amap		05/09/14
apache2		05/09/14
apm		05/09/14
apt		05/09/14
at-spi2		05/09/14
avahi		05/09/14
bash_completion.d		05/09/14
beef-xss		05/09/14
bluemaho		05/09/14
bluetooth		05/09/14
bonobo-activation		05/09/14
ca-certificates		05/09/14
calendar		05/09/14
chatscripts		05/09/14
cisco-torch		05/09/14
clutter-imcontext		05/09/14
ConsoleKit		05/09/14

Character Coding: Auto-Detect

Step 7

Click **fstab** and select **Open**.

> *Note: Depending on your screen resolution, you may not see the Open button properly. Please see the red arrow in the provided screenshot.*



Name	Size	Modified
environment	0 bytes	08/01/14
flasm.ini	1.7 KB	12/02/11
foremost.conf	8.9 KB	15/12/12
fragroute.conf	57 bytes	18/12/12
fstab	743 bytes	17/02/16
fuse.conf	216 bytes	02/01/13
gai.conf	2.9 KB	30/12/12
gnome-vfs-mime-magic	10.6 KB	04/04/07
group	1.0 KB	05/09/14
group-	1.0 KB	08/01/14
gshadow	839 bytes	05/09/14
gshadow-	839 bytes	08/01/14
gssapi_mech.conf	834 bytes	08/06/12
hdparm.conf	4.7 KB	24/06/12
host.conf	9 bytes	08/01/14
hostname	5 bytes	05/09/14
hosts	231 bytes	05/09/14
hosts.allow	580 bytes	08/01/14

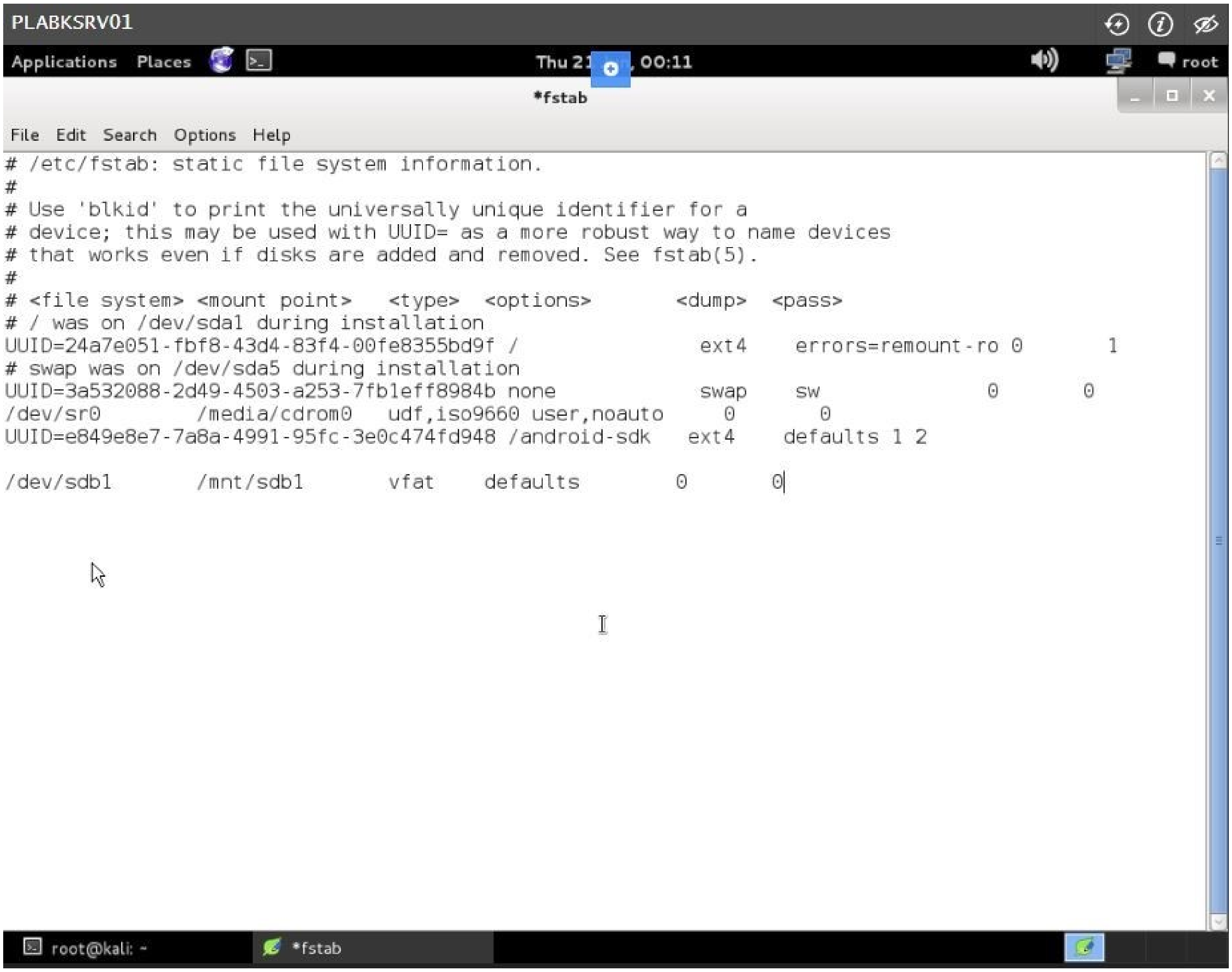
Character Coding: Auto-Detect

Step 8

When **fstab** file opens, go the end of the file and type the following entries:

```
/dev/sdb1  /mnt/sdb1  vfat  defaults  0  0
```

Press the **TAB** key to provide a space between the entries.

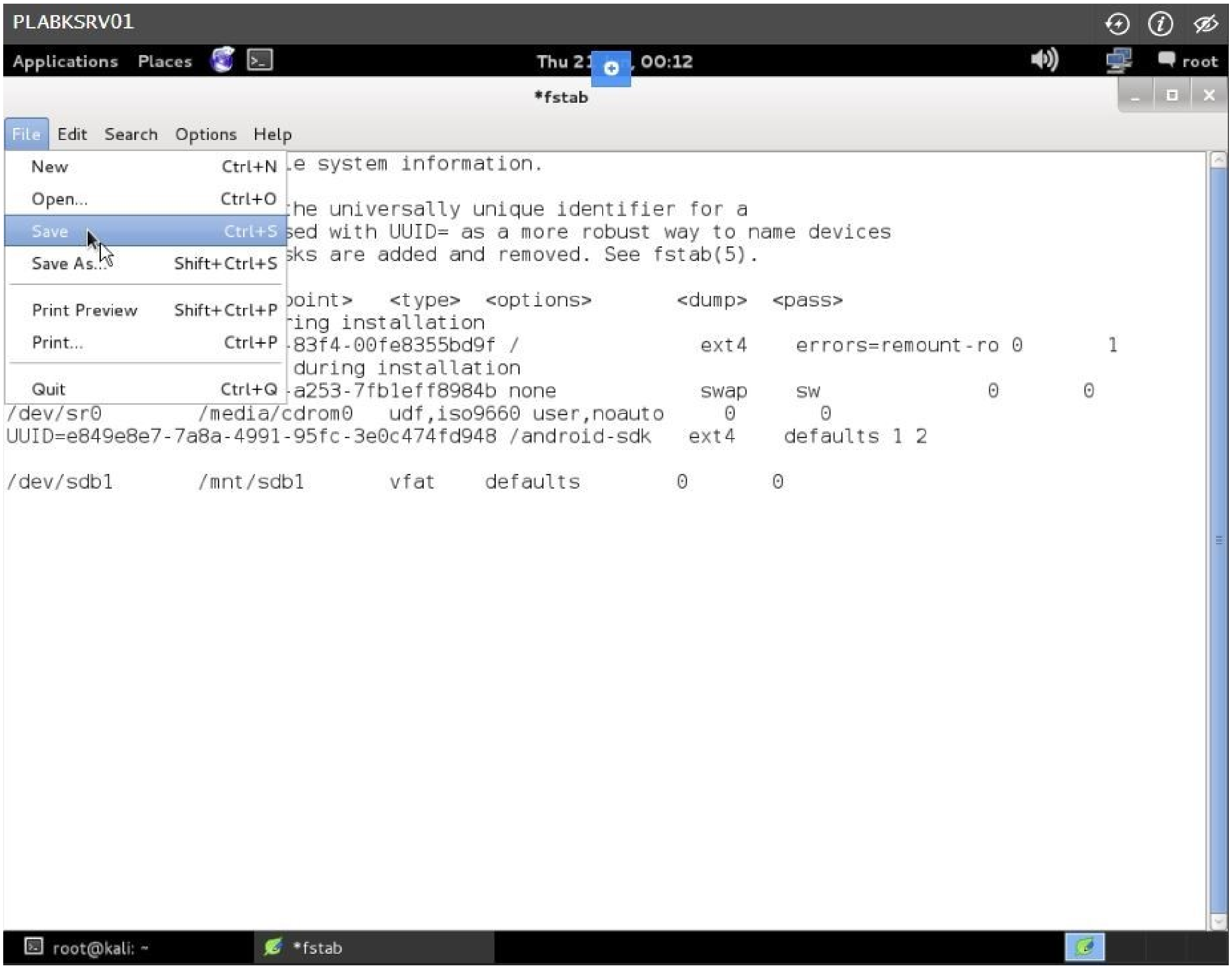


```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda1 during installation
UUID=24a7e051-fbf8-43d4-83f4-00fe8355bd9f /               ext4    errors=remount-ro 0       1
# swap was on /dev/sda5 during installation
UUID=3a532088-2d49-4503-a253-7fb1eff8984b none            swap    sw              0       0
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0
UUID=e849e8e7-7a8a-4991-95fc-3e0c474fd948 /android-sdk   ext4    defaults 1 2

/dev/sdb1       /mnt/sdb1       vfat    defaults        0       0
```

Step 9

Click **File** and select **Save**.

Close **Leafpad** text editor window.



```
.e system information.

the universally unique identifier for a
sed with UUID= as a more robust way to name devices
sks are added and removed. See fstab(5).

point>   <type>  <options>       <dump>  <pass>
ring installation
-83f4-00fe8355bd9f /               ext4    errors=remount-ro 0       1
 during installation
-a253-7fb1eff8984b none            swap    sw              0       0
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0
UUID=e849e8e7-7a8a-4991-95fc-3e0c474fd948 /android-sdk   ext4    defaults 1 2

/dev/sdb1       /mnt/sdb1       vfat    defaults        0       0
```

Step 10

Back on the terminal window, to clear the screen, type:

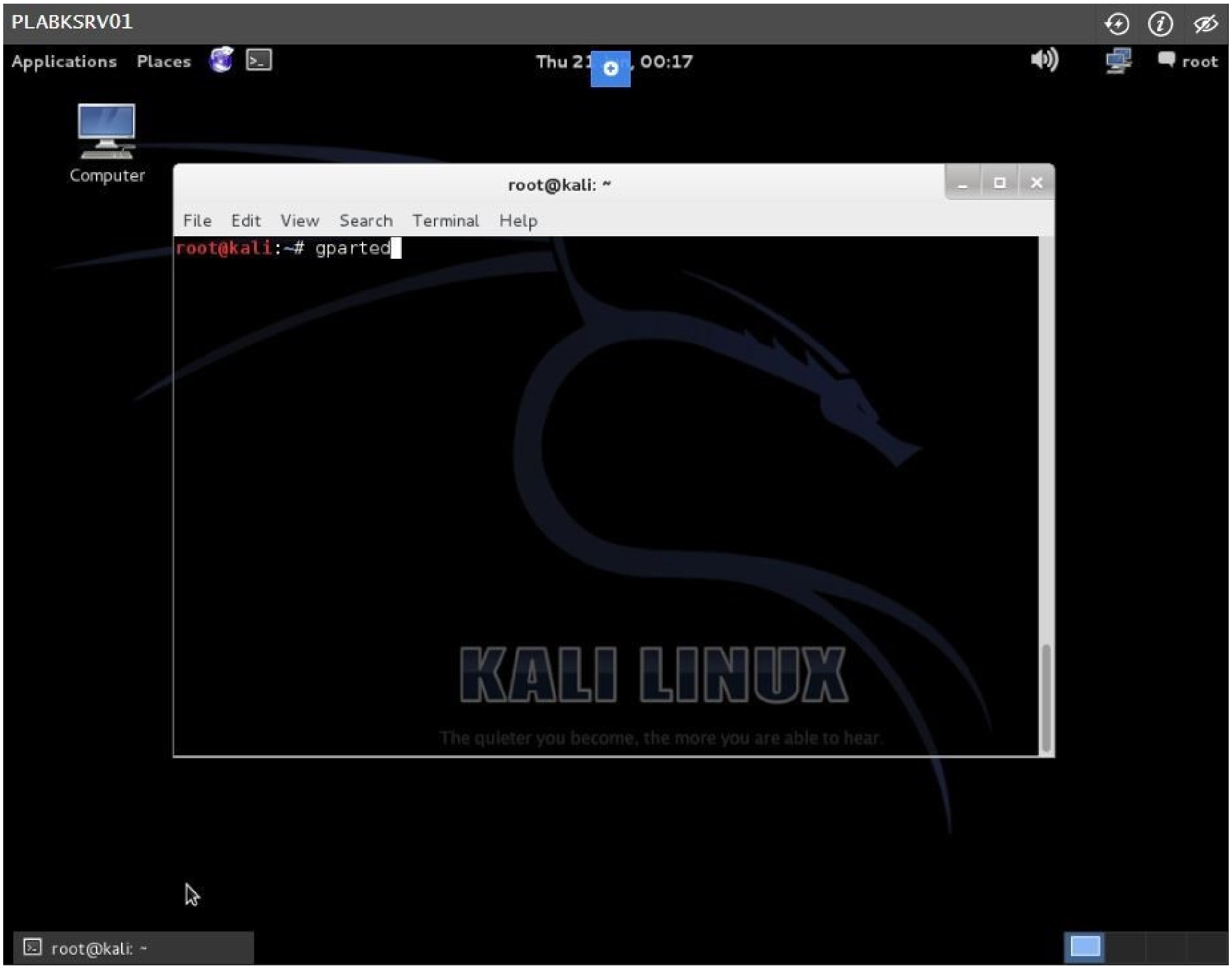
```
clear
```

Press **Enter**.

On the next prompt, type:

```
gparted
```

Press **Enter**.

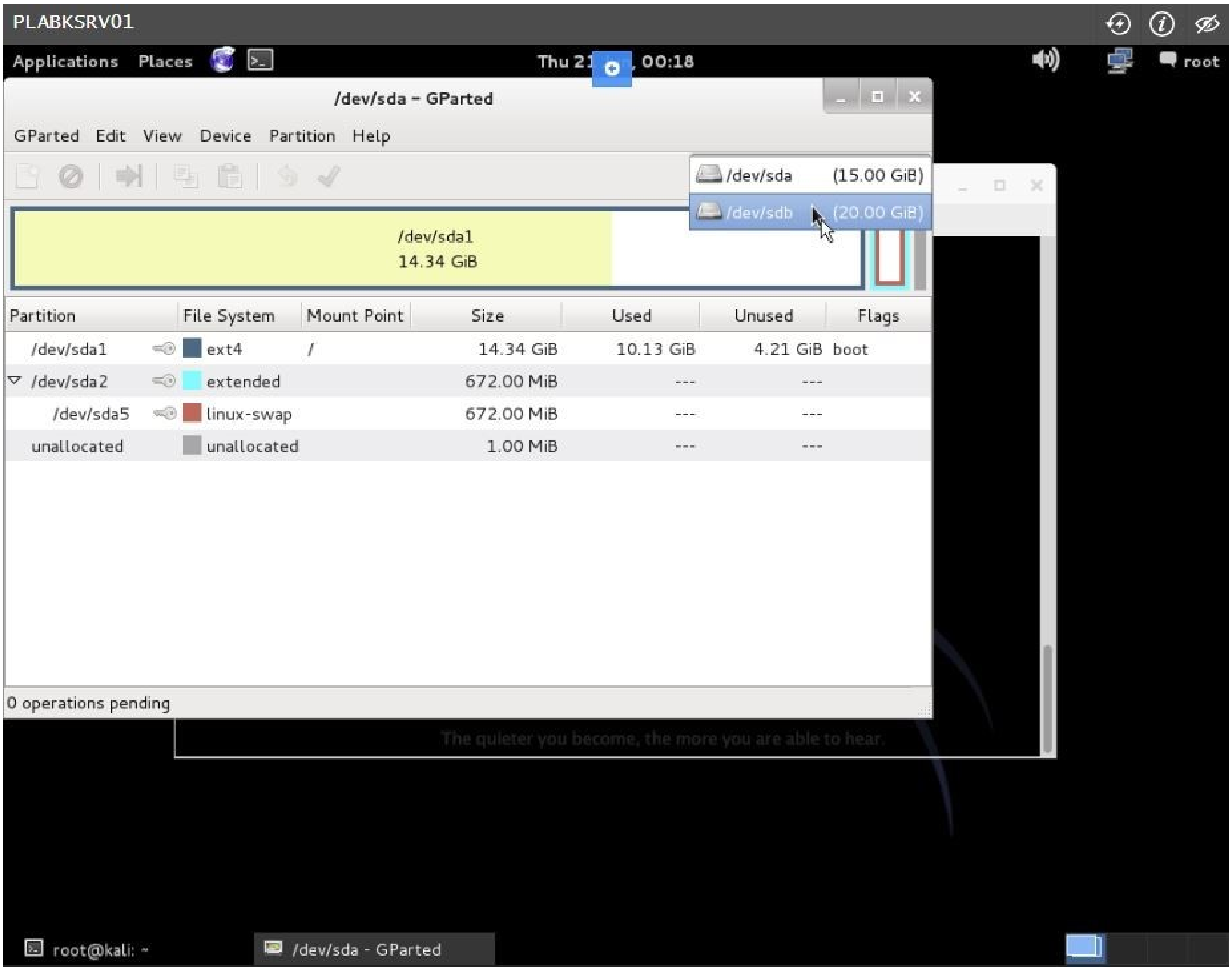


```
root@kali:~# gparted
```

Step 11

When **GParted** opens, locate the drop-down list on the far-right of the window.

Change it to **/dev/sdb**.



Partition	File System	Mount Point	Size	Used	Unused	Flags
/dev/sda1	ext4	/	14.34 GiB	10.13 GiB	4.21 GiB	boot
/dev/sda2	extended		672.00 MiB	---	---	
/dev/sda5	linux-swap		672.00 MiB	---	---	
unallocated	unallocated		1.00 MiB	---	---	

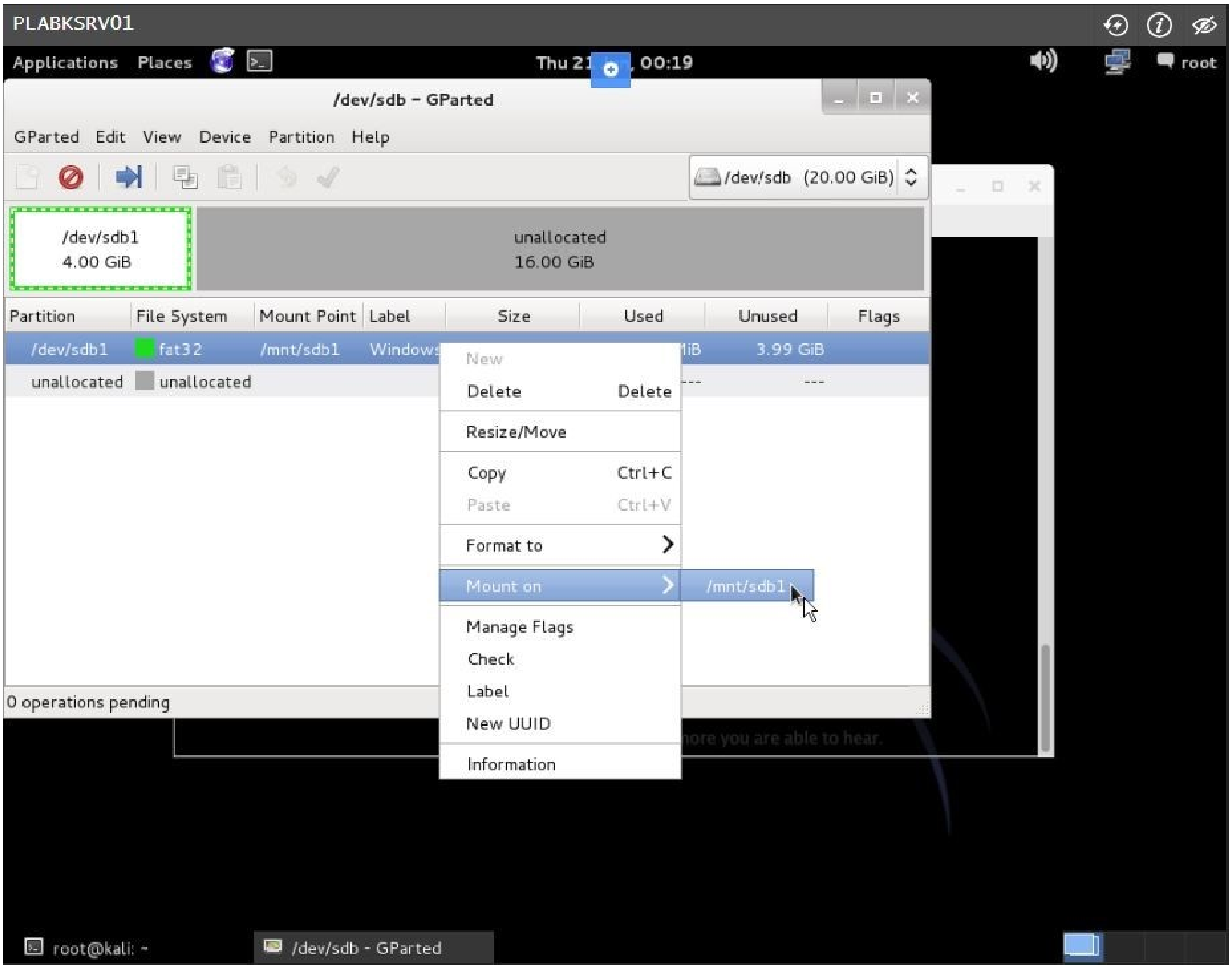
0 operations pending

Step 12

Notice that on the middle details pane **/dev/sdb1** partition is placed on mount point called /**mnt/sdb1**.

This drive can now be mounted and used to receive an image of a suspect drive. Later in this section, you learn how to mount and write to this Microsoft FAT32 target drive.

Right-click **/dev/sdb1** and select **Mount on** > **/mnt/sdb1**/.



Partition	File System	Mount Point	Label	Size	Used	Unused	Flags
/dev/sdb1	🟩 fat32	/mnt/sdb1	Windows		iB	3.99 GiB	
unallocated	▨ unallocated				---	---	

0 operations pending

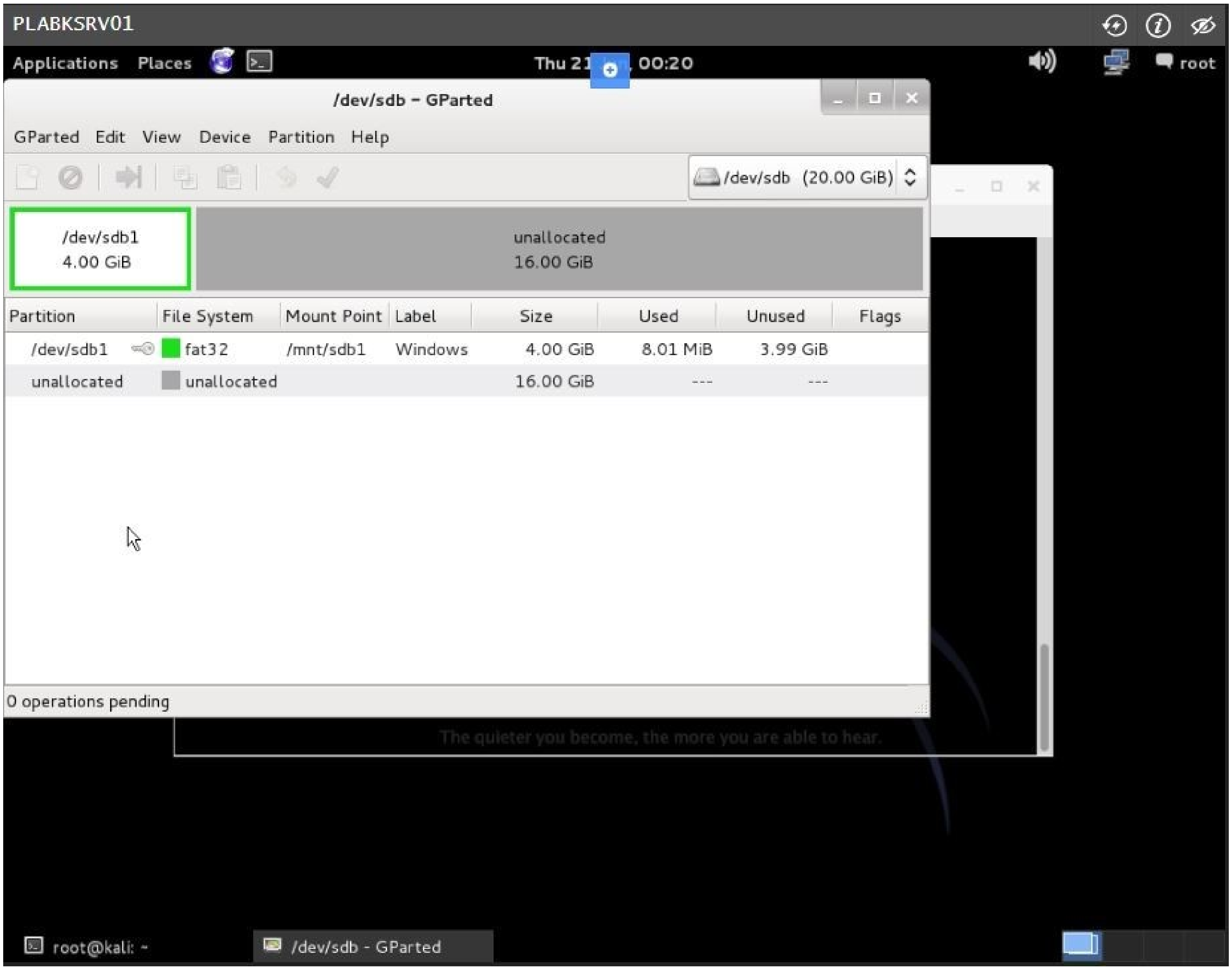
Step 13

Please wait while the partition is being mounted.

After a few seconds notice that there is a “key” icon next to fat32.

Close **GParted** window to return to Terminal window.

Keep this window open for the next activity.



Partition	File System	Mount Point	Label	Size	Used	Unused	Flags
/dev/sdb1	🟩 fat32	/mnt/sdb1	Windows	4.00 GiB	8.01 MiB	3.99 GiB	
unallocated	▨ unallocated			16.00 GiB	---	---	

0 operations pending

Keep all devices powered on in their current state and proceed to the next task.

Exercise 3-2 - Acquiring Data with dd in Linux

Follow these steps to make an image of an NTFS disk on a FAT32 disk by using the dd command.

Linux/UNIX commands are case sensitive, so make sure you type commands exactly as shown in this section's steps.

Task 1 - Using dd command

To use the dd command to create an image file of a storage device, perform the following steps:

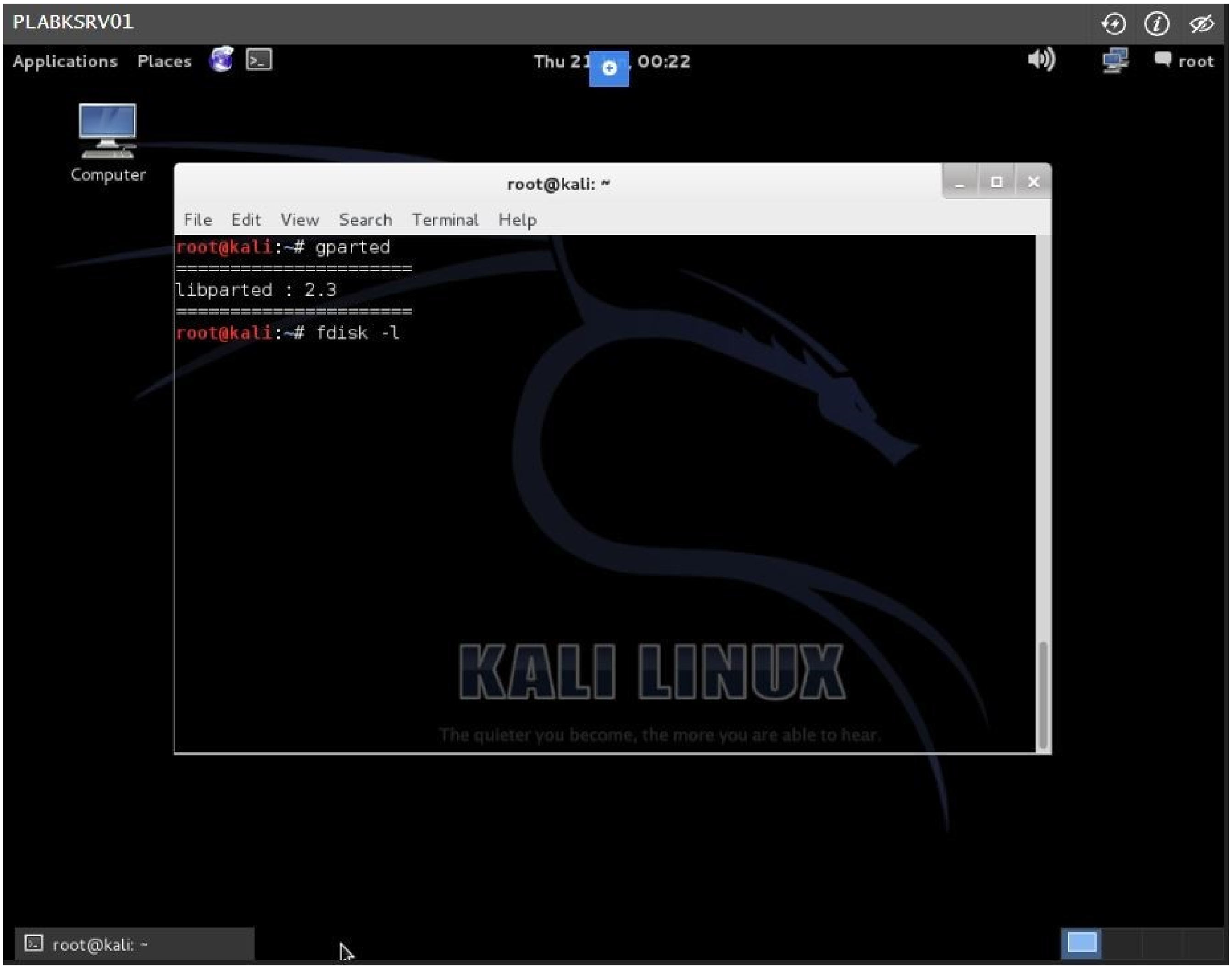
Step 1

On **PLABKSRV01** device, **KALI** Linux's Terminal window is open.

At the shell prompt, list all drives connected to the computer type:

```
fdisk -l
```

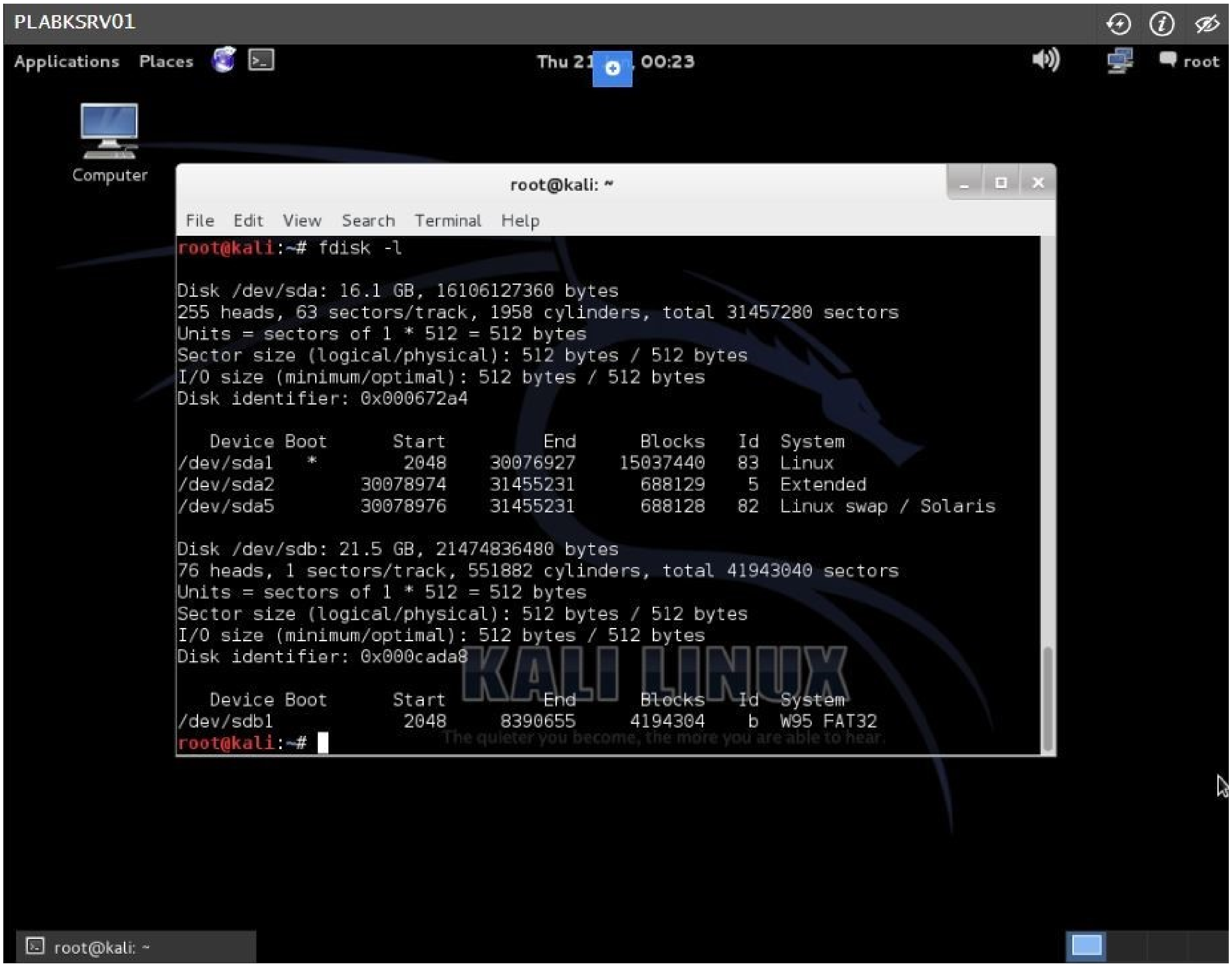
Press **Enter**.



```
root@kali:~# gparted
======================
libparted : 2.3
======================
root@kali:~# fdisk -l
```

Step 2

The following is the output of the fdisk command.



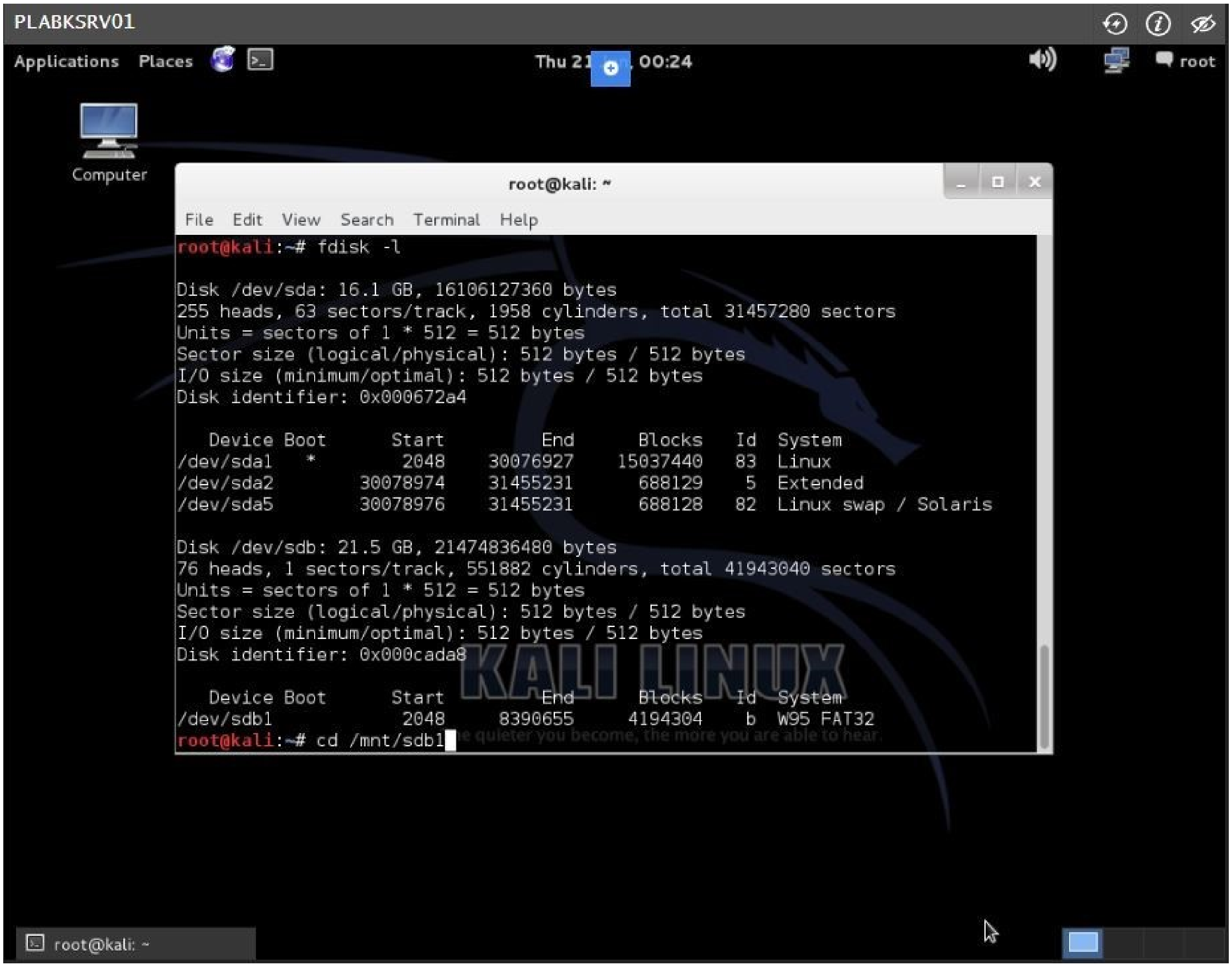
```
root@kali:~# fdisk -l

Disk /dev/sda: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders, total 31457280 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000672a4

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048    30076927    15037440   83  Linux
/dev/sda2        30078974    31455231      688129    5  Extended
/dev/sda5        30078976    31455231      688128   82  Linux swap / Solaris

Disk /dev/sdb: 21.5 GB, 21474836480 bytes
76 heads, 1 sectors/track, 551882 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000cada8

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048     8390655     4194304    b  W95 FAT32
root@kali:~# 
```

Step 3

To change your default directory to the target drive, type:

```
cd /mnt/sdb1
```

Press **Enter**.



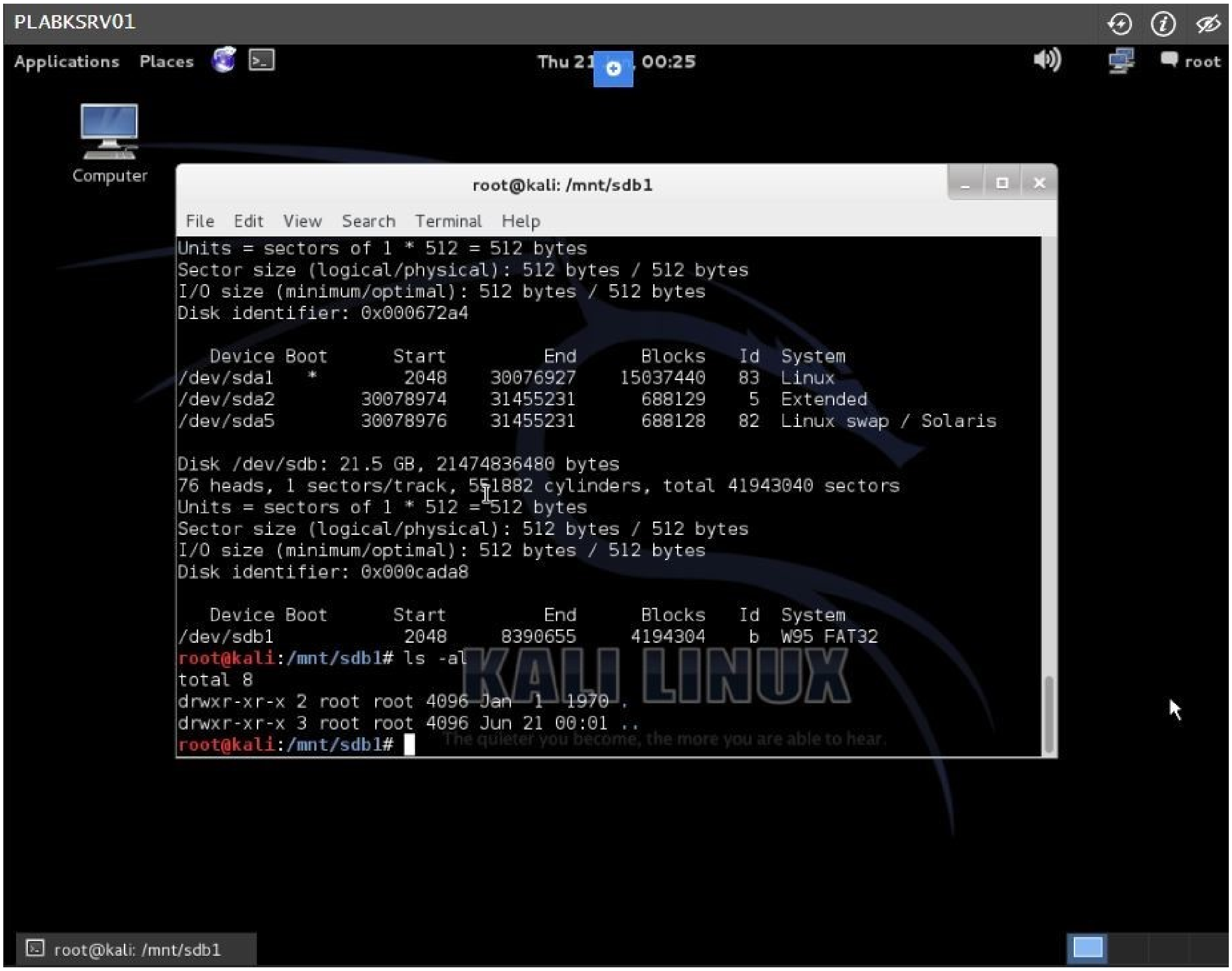
```
root@kali:~# fdisk -l

Disk /dev/sda: 16.1 GB, 16106127360 bytes
255 heads, 63 sectors/track, 1958 cylinders, total 31457280 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000672a4

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048    30076927    15037440   83  Linux
/dev/sda2        30078974    31455231      688129    5  Extended
/dev/sda5        30078976    31455231      688128   82  Linux swap / Solaris

Disk /dev/sdb: 21.5 GB, 21474836480 bytes
76 heads, 1 sectors/track, 551882 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000cada8

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048     8390655     4194304    b  W95 FAT32
root@kali:~# cd /mnt/sdb1
```

Step 4

List the contents of the target drive's root level by typing **ls -al** and pressing Enter.

Your output should be similar to the following:

```
total 8
drwxr-xr-x 2 root root 4096 Jun 1 03:35 .
drwxr-xr-x 3 root root 4096 Jun 1 03:35 ..
```



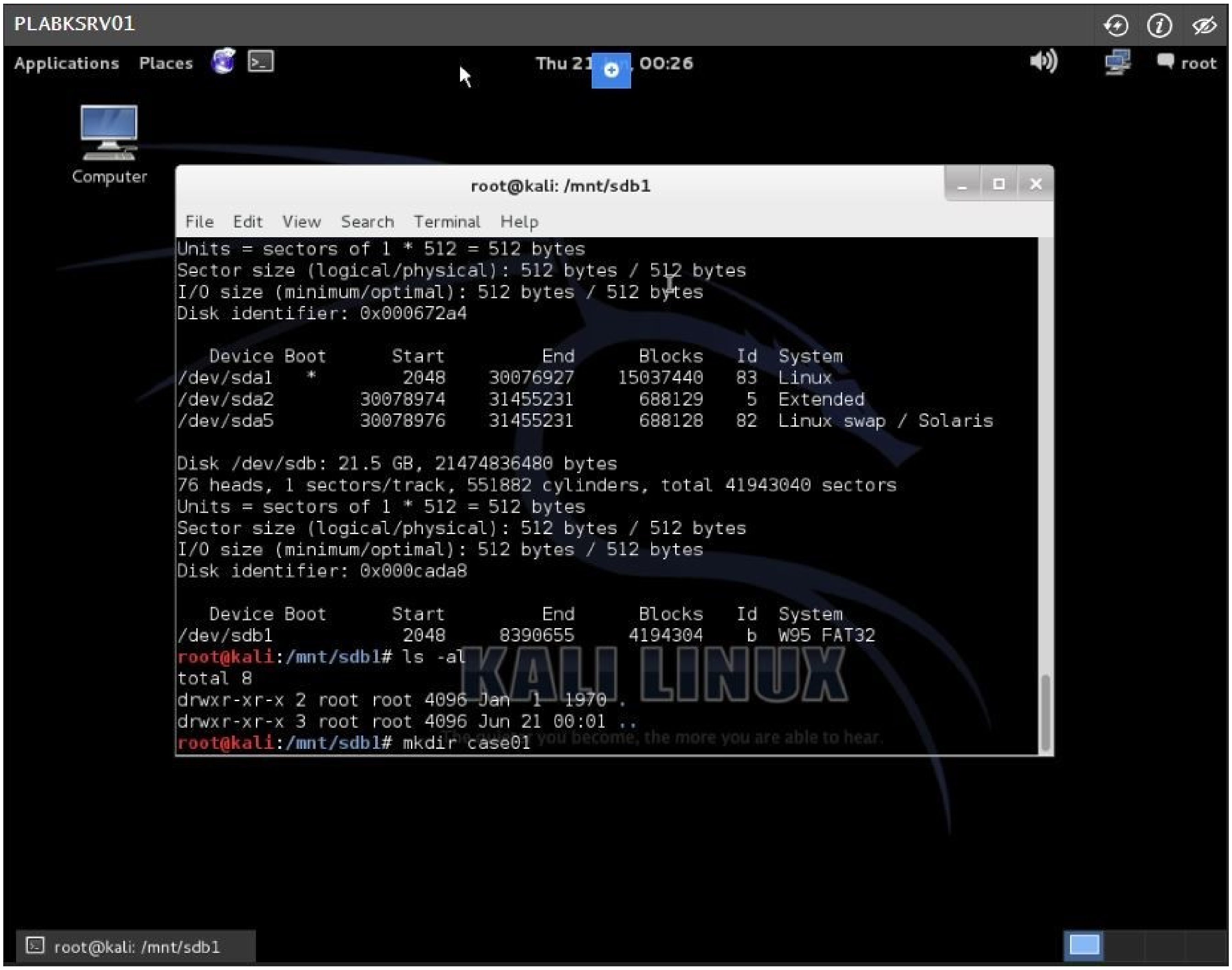
```
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000672a4

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048    30076927    15037440   83  Linux
/dev/sda2        30078974    31455231      688129    5  Extended
/dev/sda5        30078976    31455231      688128   82  Linux swap / Solaris

Disk /dev/sdb: 21.5 GB, 21474836480 bytes
76 heads, 1 sectors/track, 551882 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000cada8

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048     8390655     4194304    b  W95 FAT32
root@kali:/mnt/sdb1# ls -al
total 8
drwxr-xr-x 2 root root 4096 Jan  1  1970 .
drwxr-xr-x 3 root root 4096 Jun 21 00:01 ..
root@kali:/mnt/sdb1# 
```

Step 5

To make a target directory to receive image saves of the suspect drive, type:

```
mkdir case01
```

Press **Enter**.



```
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000672a4

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048    30076927    15037440   83  Linux
/dev/sda2        30078974    31455231      688129    5  Extended
/dev/sda5        30078976    31455231      688128   82  Linux swap / Solaris

Disk /dev/sdb: 21.5 GB, 21474836480 bytes
76 heads, 1 sectors/track, 551882 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000cada8

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048     8390655     4194304    b  W95 FAT32
root@kali:/mnt/sdb1# ls -al
total 8
drwxr-xr-x 2 root root 4096 Jan  1  1970 .
drwxr-xr-x 3 root root 4096 Jun 21 00:01 ..
root@kali:/mnt/sdb1# mkdir case01
```

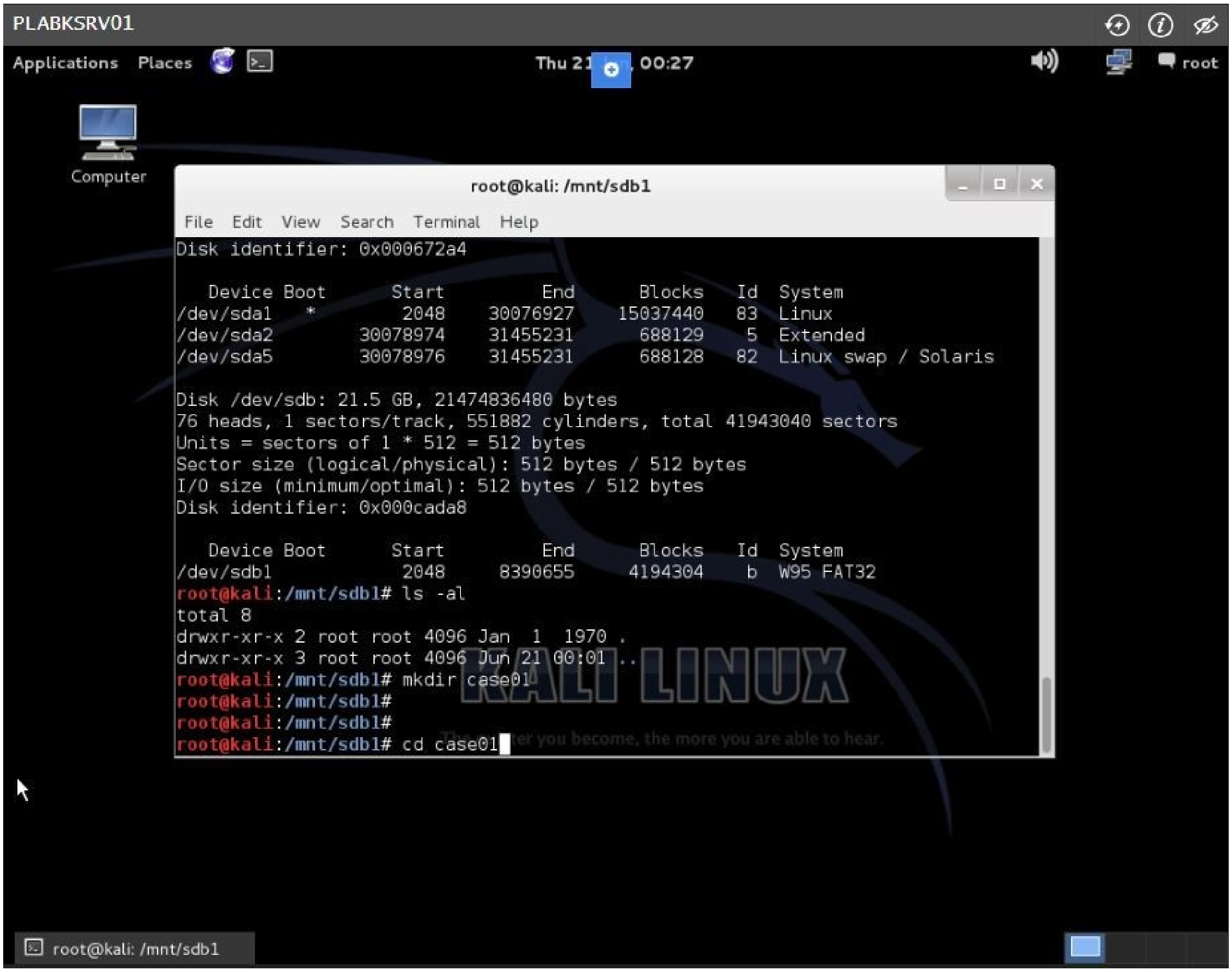
Step 6

To change to the newly created target directory, type:

```
cd case01
```

Press **Enter**.

Don’t close the shell window.



```
Disk identifier: 0x000672a4

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048    30076927    15037440   83  Linux
/dev/sda2        30078974    31455231      688129    5  Extended
/dev/sda5        30078976    31455231      688128   82  Linux swap / Solaris

Disk /dev/sdb: 21.5 GB, 21474836480 bytes
76 heads, 1 sectors/track, 551882 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000cada8

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048     8390655     4194304    b  W95 FAT32
root@kali:/mnt/sdb1# ls -al
total 8
drwxr-xr-x 2 root root 4096 Jan  1  1970 .
drwxr-xr-x 3 root root 4096 Jun 21 00:01 ..
root@kali:/mnt/sdb1# mkdir case01
root@kali:/mnt/sdb1#
root@kali:/mnt/sdb1#
root@kali:/mnt/sdb1# cd case01
```

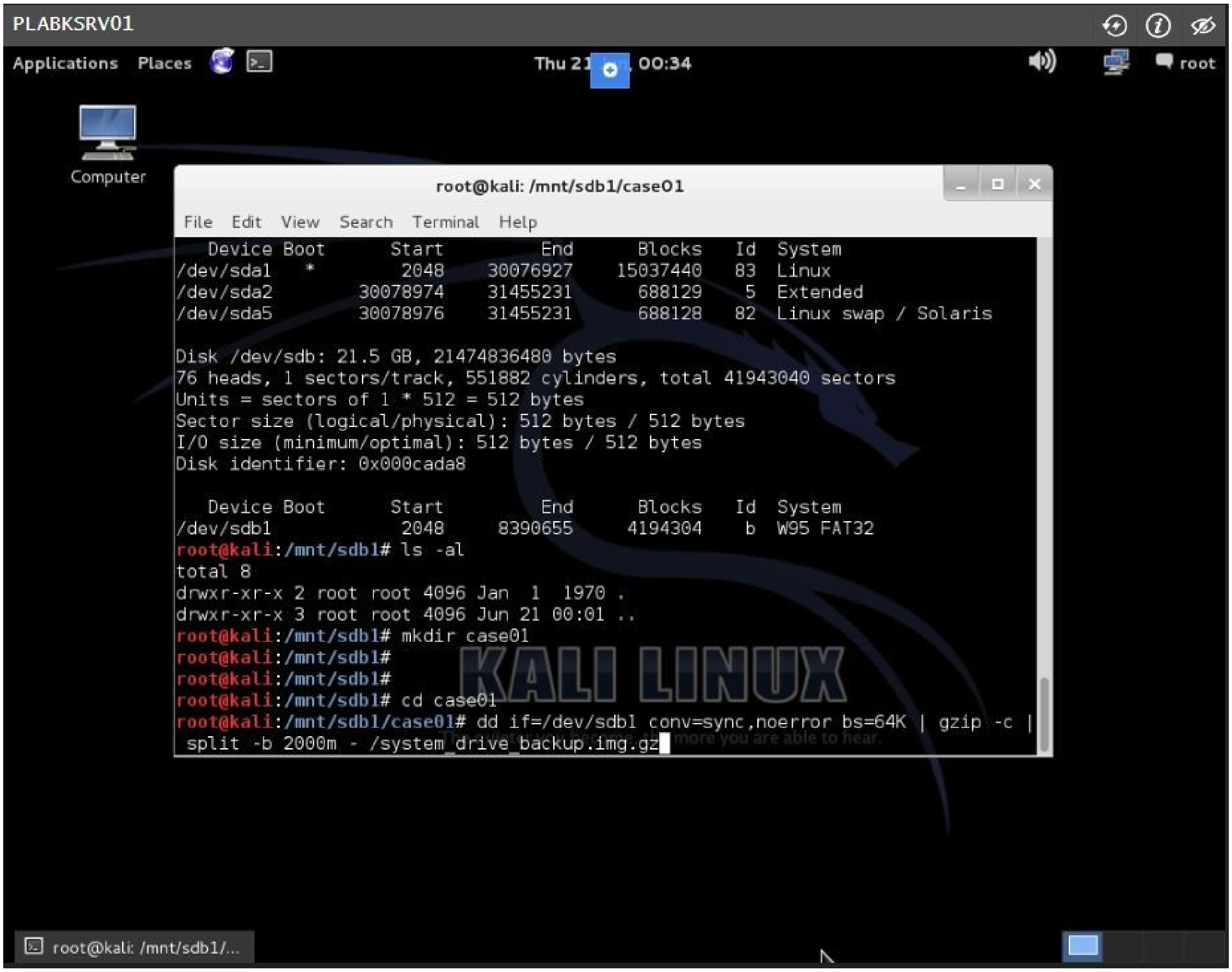
Step 7

Next, you perform a raw format image of the entire suspect drive to the target directory. To do this, you use the split command with the dd command. The split command creates a two-letter extension for each segmented volume. As a general rule, if you plan to use a Windows forensics tool to examine a dd image file created with this switch, the segmented volumes shouldn't exceed 2 GB each because of FAT32 file size limits. This 2 GB limit allows you to copy only up to 198 GB of a suspect's disk. If you need to use the dd command, it's better to use the split command's default of incremented letter extensions and make smaller segments.

Now, type:

```
dd if=/dev/sdb1 conv=sync,noerror bs=64K | gzip -c |
split -b 2000m - /system_drive_backup.img.gz
```

Press **Enter**.

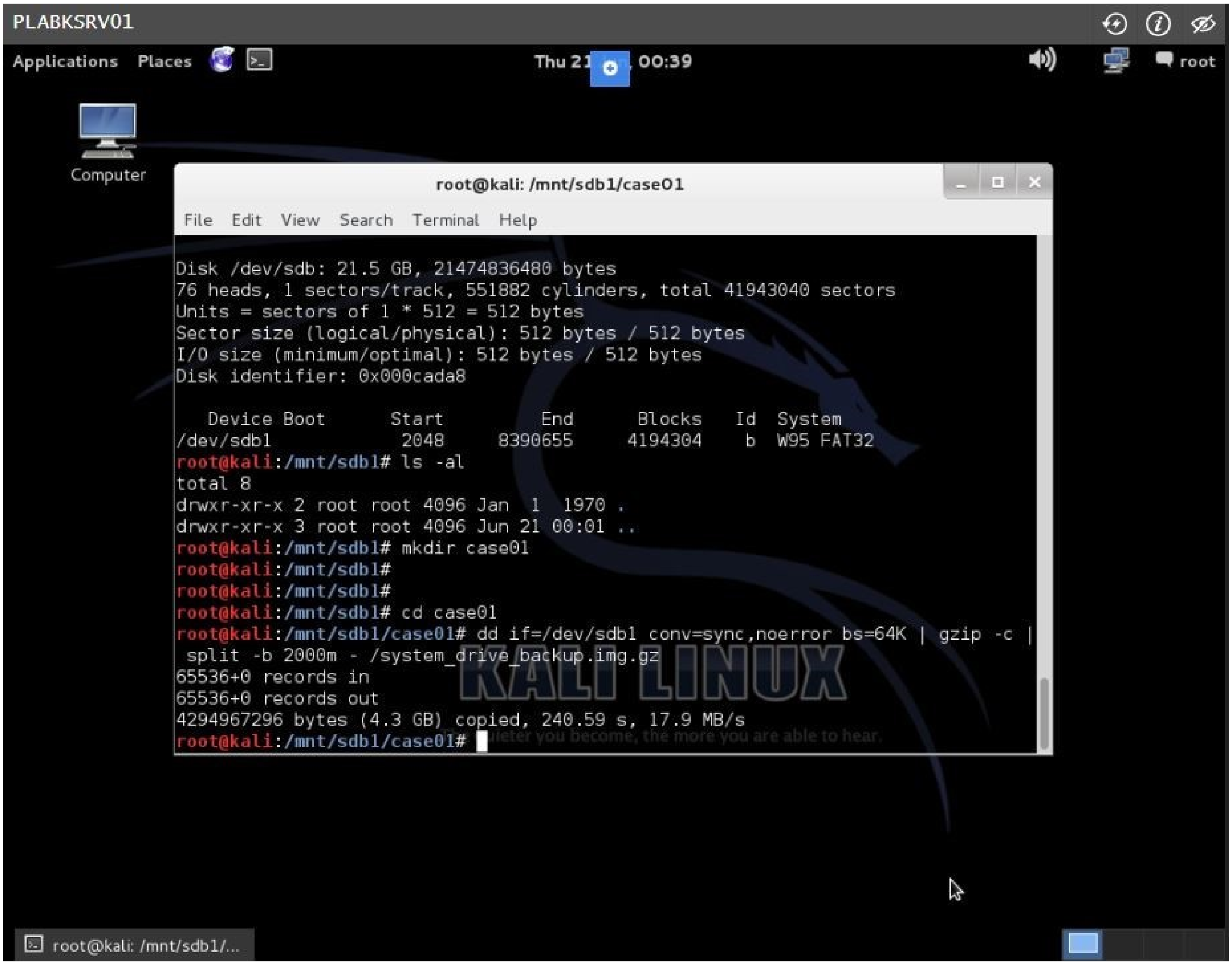

PLABKSRV01
Applications Places
Thu 21 , 00:34
root
Computer
root@kali: /mnt/sdb1/case01
File Edit View Search Terminal Help
KALI LINUX
The quieter you become, the more you are able to hear.
root@kali: /mnt/sdb1/...

```
   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *        2048    30076927    15037440   83  Linux
/dev/sda2        30078974    31455231      688129    5  Extended
/dev/sda5        30078976    31455231      688128   82  Linux swap / Solaris

Disk /dev/sdb: 21.5 GB, 21474836480 bytes
76 heads, 1 sectors/track, 551882 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000cada8

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048     8390655     4194304    b  W95 FAT32
root@kali:/mnt/sdb1# ls -al
total 8
drwxr-xr-x 2 root root 4096 Jan  1  1970 .
drwxr-xr-x 3 root root 4096 Jun 21 00:01 ..
root@kali:/mnt/sdb1# mkdir case01
root@kali:/mnt/sdb1#
root@kali:/mnt/sdb1#
root@kali:/mnt/sdb1# cd case01
root@kali:/mnt/sdb1/case01# dd if=/dev/sdb1 conv=sync,noerror bs=64K | gzip -c |
 split -b 2000m - /system_drive_backup.img.gz
```

Step 8

After a few minutes, you will get the results of the created file image, similar to the following screenshot.

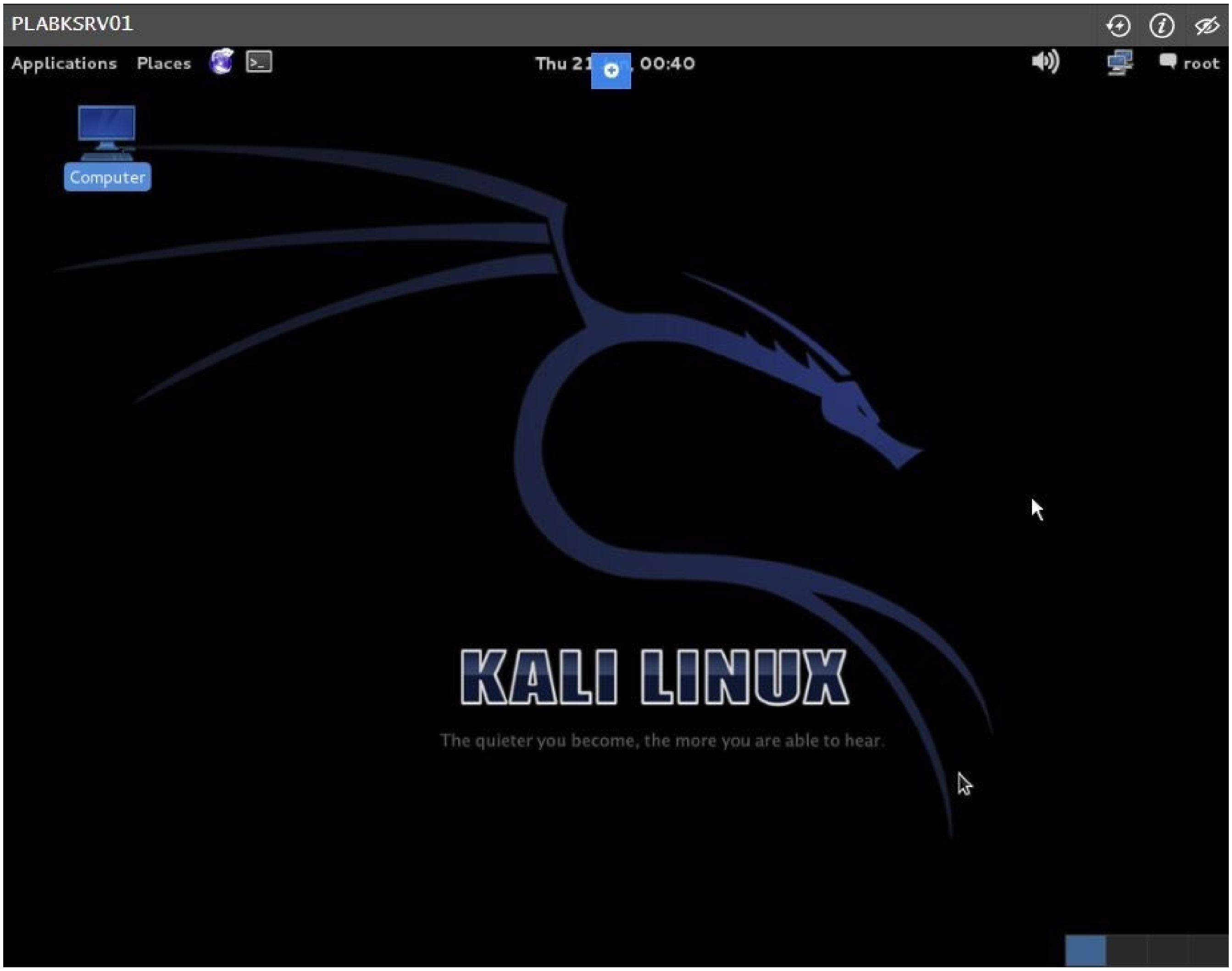
Close **Terminal** window.



```
Disk /dev/sdb: 21.5 GB, 21474836480 bytes
76 heads, 1 sectors/track, 551882 cylinders, total 41943040 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000cada8

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            2048     8390655     4194304    b  W95 FAT32
root@kali:/mnt/sdb1# ls -al
total 8
drwxr-xr-x 2 root root 4096 Jan  1  1970 .
drwxr-xr-x 3 root root 4096 Jun 21 00:01 ..
root@kali:/mnt/sdb1# mkdir case01
root@kali:/mnt/sdb1#
root@kali:/mnt/sdb1#
root@kali:/mnt/sdb1# cd case01
root@kali:/mnt/sdb1/case01# dd if=/dev/sdb1 conv=sync,noerror bs=64K | gzip -c |
 split -b 2000m - /system_drive_backup.img.gz
65536+0 records in
65536+0 records out
4294967296 bytes (4.3 GB) copied, 240.59 s, 17.9 MB/s
root@kali:/mnt/sdb1/case01#
```

Step 9

To view the raw image that have been created from the dd and split commands, double-click **Computer** icon on desktop.

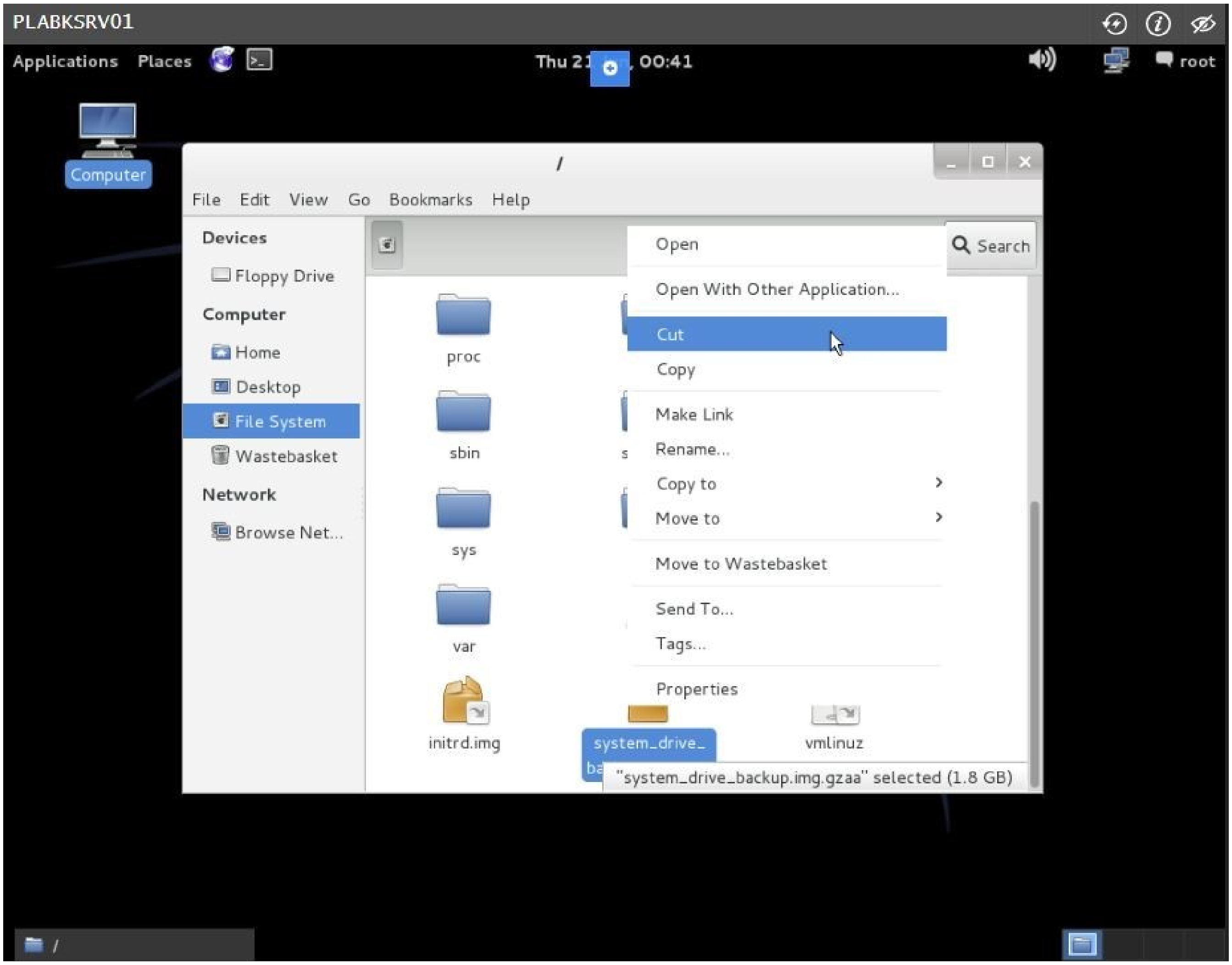


Step 10

On the open window, click **File System** on the left pane.

Notice the **system_drive_backup.img.gzaa** image file.

Right-click the said file and select **Cut**.

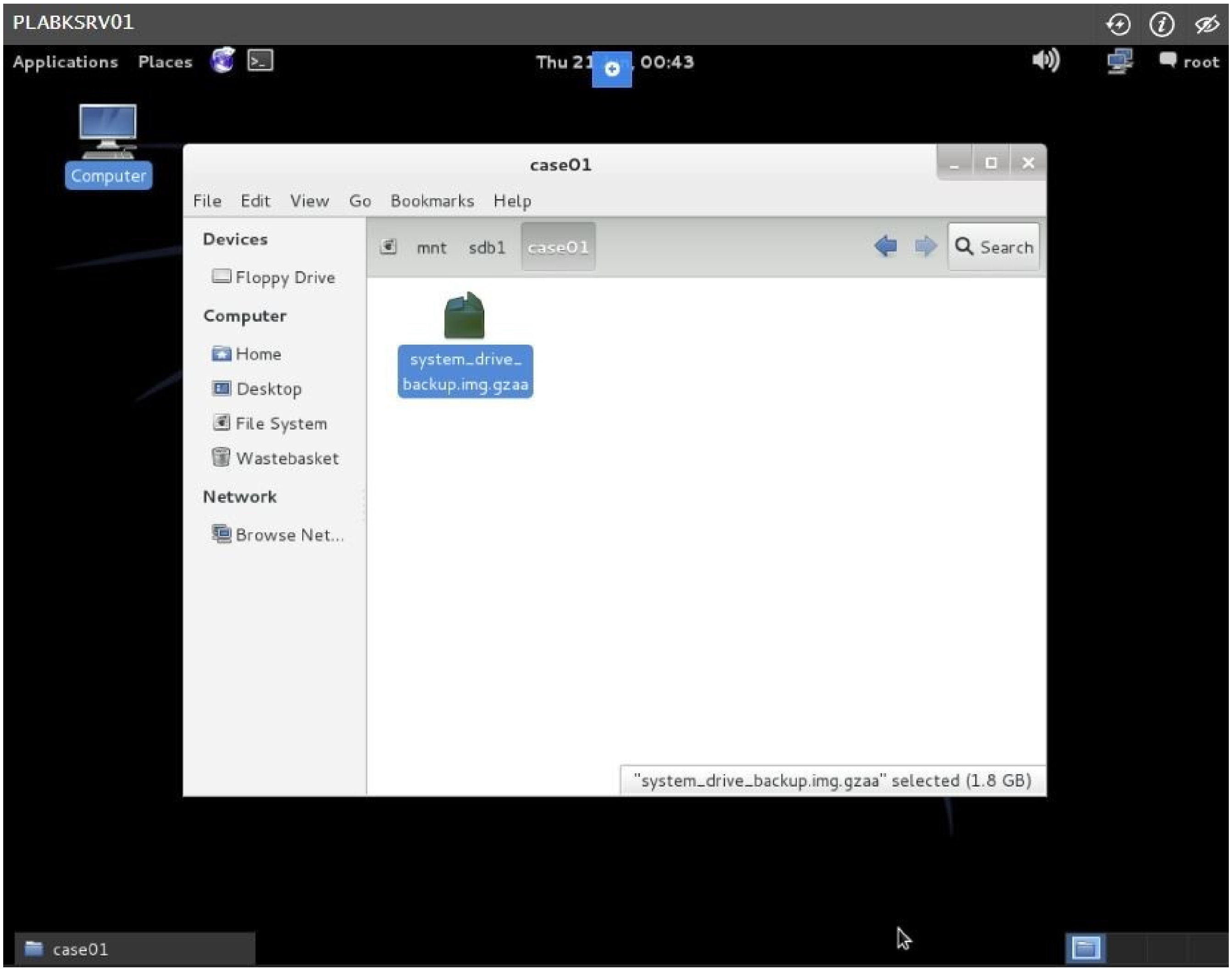


"system_drive_backup.img.gzaa" selected (1.8 GB)

Step 11

Paste the image file to **/mnt/sdb1/case01** folder.

Close **File System** window.



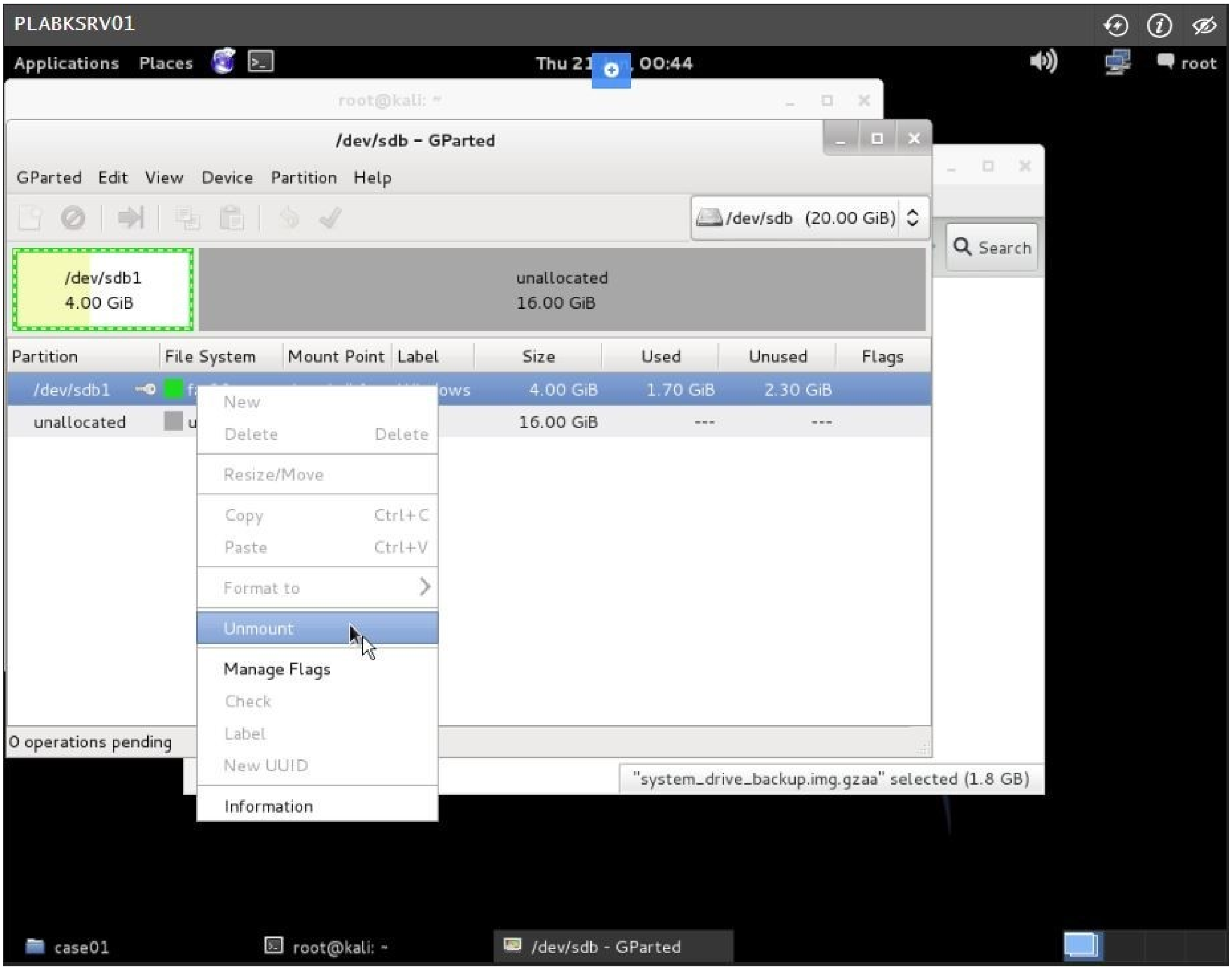
"system_drive_backup.img.gzaa" selected (1.8 GB)

Step 12

To complete this acquisition, reopen **GParted**.

Dismount the target **/dev/sdb1** by right-clicking it and selecting **Unmount.**

Close **GParted** after the device is dismounted.



Partition	File System	Mount Point	Label	Size	Used	Unused	Flags
/dev/sdb1	f		ows	4.00 GiB	1.70 GiB	2.30 GiB	
unallocated	u			16.00 GiB	---	---	

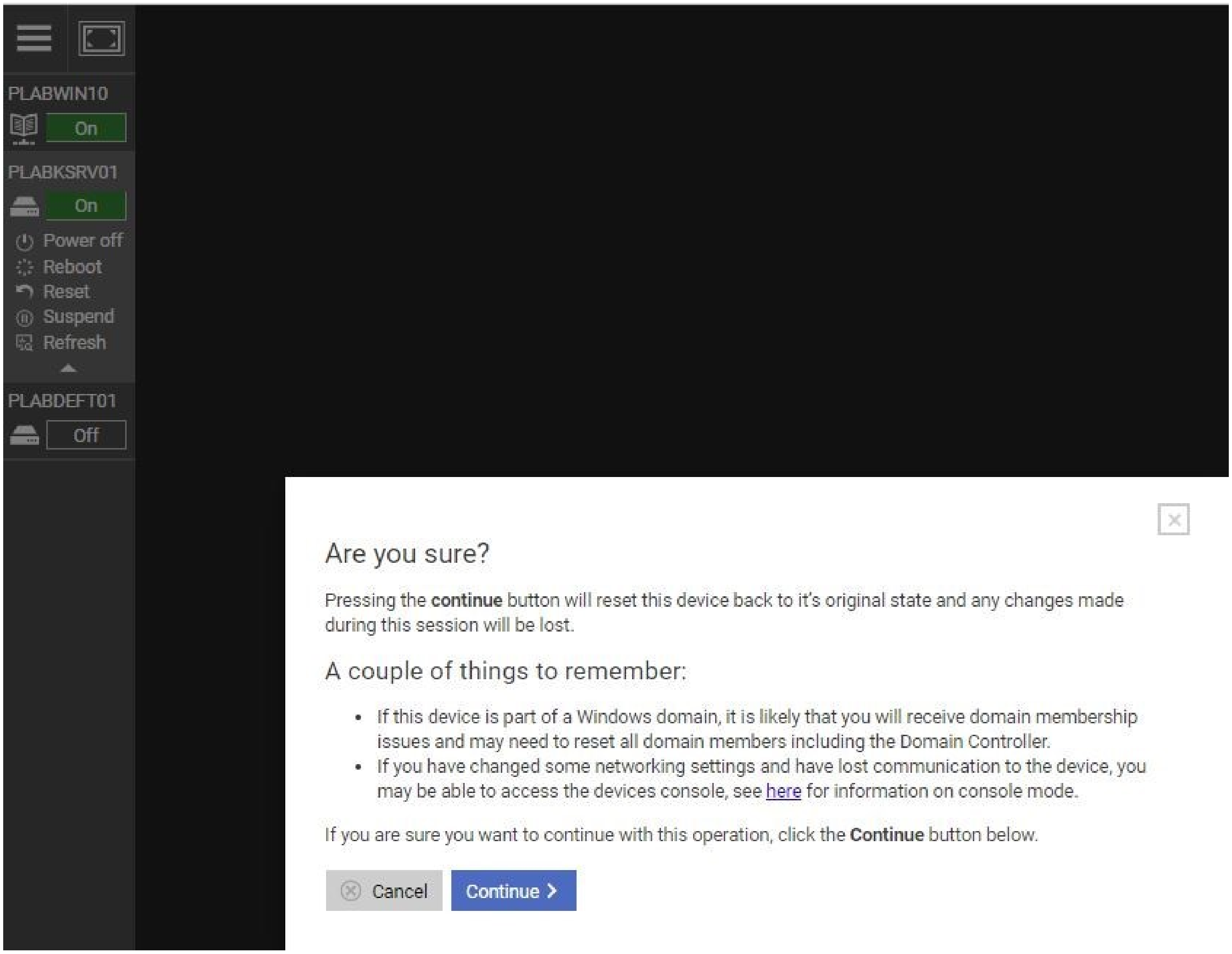
0 operations pending

"system_drive_backup.img.gzaa" selected (1.8 GB)

Step 13

Close **Terminal** window.

Within the Practice Lab Application hover over the **PLABKSRV01** device and select **Reset**. Click Continue in the Are you sure? Box.



Are you sure?

Pressing the **continue** button will reset this device back to it's original state and any changes made during this session will be lost.

A couple of things to remember:

- If this device is part of a Windows domain, it is likely that you will receive domain membership issues and may need to reset all domain members including the Domain Controller.
- If you have changed some networking settings and have lost communication to the device, you may be able to access the devices console, see here for information on console mode.

If you are sure you want to continue with this operation, click the **Continue** button below.

Hands-On Project 3-3

In this project, you prepare a drive and create a FAT32 disk partition using Linux. You need the following:

- A Linux distribution or Linux Live CD
- A disk drive
- A method of connecting a disk drive to your workstation, such as USB, FireWire, external SATA, or internal connections, such as PATA or SATA
- A review of the steps in the "Preparing a Target Drive for Acquisition in Linux" section

To format a drive as FAT32 in Linux, follow these steps:

Step 1

Connect to **PLABKSRV01** device.

Step 2

Follow the steps in the "**Exercise 3-1 - Preparing a Target Disk for Acquisition in Linux**" section.

Step 3

When you've finished formatting the target drive, leave it connected for the next project.

Hands-On Project 3-4

In this project, you use the Linux dd command to make an acquisition split into 30 MB segmented volumes. Then you validate the data by using the Linux md5sum command on the original drive and the image files. The output for md5sum is then redirected to a data file kept with the image files. For this project, you need the following:

- A Linux distribution or Linux Live CD
- The FAT32 drive partitioned and formatted in Hands-On Project 3-3
- A method of connecting the FAT32 drive and the drive created in Hands-On Project 3-3 to your workstation, such as USB, FireWire, external SATA, or internal connections, such as PATA or SATA
- A review of the "Acquiring Data with dd in Linux" and "Validating dd Acquired Data" sections. Follow these steps:

Step 1

Make sure you've connected the drive you prepared in Hands-On Project 3-3 to your Linux workstation.

Step 2

Reset the Linux system, and make the dd acquisition, following the steps in "**Exercise 3-2 - Acquiring Data with dd in Linux**."

Step 3

When the acquisition is done, when you're finished, close the application or shell window, and log off Linux.

Summary

- Forensics data acquisitions are stored in three different formats: raw, proprietary, and AFF. Most proprietary formats and AFF store metadata about the acquired data in the image file.
- The four methods of acquiring data for forensics analysis are disk-to-image file, disk-to-disk copy, logical disk-to-disk or disk-to-data file, or sparse data copy of a folder or file.
- Lossless compression for forensics acquisitions doesn't alter the data when it's restored, unlike lossy compression. Lossless compression can compress up to 50% for most data. If data is already compressed on a drive, lossless compression might not save much more space.
- If there are time restrictions or too much data to acquire from large drives or RAID drives, a logical or sparse acquisition might be necessary. Consult

with your lead attorney or supervisor first to let them know that collecting all the data might not be possible.

- You should have a contingency plan to ensure that you have a forensically sound acquisition and make two acquisitions if you have enough data storage. The first acquisition should be compressed, and the second should be uncompressed. If one acquisition becomes corrupt, the other one is available for analysis.
- Write-blocking devices or utilities must be used with GUI acquisition tools in both Windows and Linux. Practice with a test drive rather than suspect drive, and use a hashing tool on the test drive to verify that no data was altered.
- Always validate your acquisition with built-in tools from a forensics acquisition program, a hexadecimal editor with MD5 or SHA-1 hashing functions, or the Linux md5sum or sha1sum commands.
- A Linux Live CD, such as SIFT, Kali Linux, or Deft, provides many useful tools for digital forensics acquisitions.
- The preferred Linux acquisition tool is dcfldd instead of dd because it was designed for forensics acquisition. The dcfldd tool is also available for Windows. Always validate the acquisition with the hashing features of dcfldd and md5sum or sha1sum.
- When using the Linux dd or dcfldd commands, remember that reversing the output field (of¼) and input field (if¼) of suspect and target drives could write data to the wrong drive, thus destroying your evidence. If available, you should always use a physical write-blocker device for acquisitions.
- To acquire RAID disks, you need to determine the type of RAID and which acquisition tool to use. With a firmware-hardware RAID, acquiring data directly from the RAID server might be necessary.

- Remote network acquisition tools require installing a remote agent on the suspect computer. The remote agent can be detected if suspects install their own security programs, such as a firewall.