Assignment No. 3: **Write a Computer Forensics Application Program inJava/Python/C++ for recovering Deleted Files and Deleted Partitions**

Name:	**Class:** B.E.	**Division:** A / B/C
Roll No:	**Date of Submission:**	
Marks Obtained: **/ 10**	**Signature of subject teacher:**	

3.1 Prerequisite:

a) Knowledge about Partitions in Ubuntu. b) Path of Trash folder.

3.2 Learning Objectives:

- Understand the concept of Recovery of deleted files.
- Implementation of recovery of deleted Partitions.

3.3New Concepts:

a. Recovery of files in LINUX OS.

3.4 Theory

3.4.1 Introduction

Have you accidentally deleted an important file because you are in a habit of using "Shift+Del" rather than delete only?? Well don't panic.There are many utilities in Ubuntu and other Linux distributions which helps you in recovering the so called "permanently deleted" files. Actually when you delete a file permanently (accidentally or intentionally), It doesn't get removed from your hard disk.It get stored in certain blocks of the storage device and they continue to exist in the blocks unless you overwrite them with newer files. There are many Tools available to recover permanently deleted files Scalpel.

Scalpel is a platform independent command based tool which is small yet very powerful. But, if the file is deleted i.e. by just pressing Delete button the file is stored in Trash folder in Ubuntu OS. So it is easy to recover the deleted files from Trash Folder. Just we need to know the path of trash folder.

Path is: ="/home/gurukul/.local/share/Trash/files"

There are sub-Folders in Trash Folder namely :

1. files- contains files which are deleted
2. info- contains information of files deleted
3. expunged

3.4.2 Introduction to file systems:

File systems are one of the things any newcomer to linux must become acquainted with. In the world of Microsoft you never really have to worry about it, the default being NTFS. Linux however, being built on a world of open source and differing opinions, is not limited in this way and so the user should have an understanding of what a file system is, and how it affects the computer.

At the core of a computer, it's all 1s and 0s, but the organization of that data is not quite as simple. A *bit* is a 1 or a 0, a *byte* is composed of 8 bits, a kilobyte is 1024 (i.e. 2) bytes, a megabyte is 1024 kilobytes and so on and so forth. All these *bits* and *bytes* are permanently stored on a Hard Drive. A hard drive stores all your data, any time you save a file, you're writing thousands of 1s and 0s to a metallic disc, changing the magnetic properties that can later be read as 1 or 0. There is so much data on a hard drive that there has to be some way to organize it, like a library of books and the old card drawers that indexed all of them, without that index, we'd be lost. Libraries, for the most part, use the Dewey Decimal System to organize their books, but

there exist other systems to do so, none of which have attained the same fame as Mr. Dewey's invention. File systems are the same way. The ones most users are aware of are the ones Windows uses, the vFat or the NTFS systems, these are the Windows default file systems.

Ubuntu (like all UNIX-like systems) organizes files in a hierarchical tree, where relationships are thought of in teams of children and parent. *Directories* can contain other directories as well as *regular files*, which are the "leaves" of the tree. Any element of the tree can be references by a *path name*; an *absolute path name* starts with the character / (identifying the *root directory*, which contains all other directories and files), then every child directory that must be traversed to reach the element is listed, each separated by a / sign.

3.4.3 Main directories

The standard Ubuntu directory structure mostly follows the File system Hierarchy Standard, which can be referred to for more detailed information.

Here, only the most important directories in the system will be presented.
/bin is a place for most commonly used terminal commands, like ls, mount, rm, etc.

/boot contains files needed to start up the system, including the Linux kernel, a RAM disk image and bootloader configuration files.

/dev contains all *device files*, which are not regular files but instead refer to various hardware devices on the system, including hard drives.

/etc contains system-global configuration files, which affect the system's behavior for all users. **/home** home sweet home, this is the place for users' home directories.

/lib contains very important dynamic libraries and kernel modules

/media is intended as a mount point for external devices, such as hard drives or removable media (floppies, CDs, DVDs).

/mnt is also a place for mount points, but dedicated specifically to "temporarily mounted" devices, such as network filesystems.

/opt can be used to store addition software for your system, which is not handled by the package manager.

/proc is a virtual filesystem that provides a mechanism for kernel to send information to processes.

/root is the superuser's home directory, not in /home/ to allow for booting the system even if /home/ is not available.

/sbin contains important administrative commands that should generally only be employed by the superuser.

/srv can contain data directories of services such as HTTP (/srv/www/) or FTP.

/sys is a virtual filesystem that can be accessed to set or obtain information about the kernel's view of the system.

/tmp is a place for temporary files used by applications.

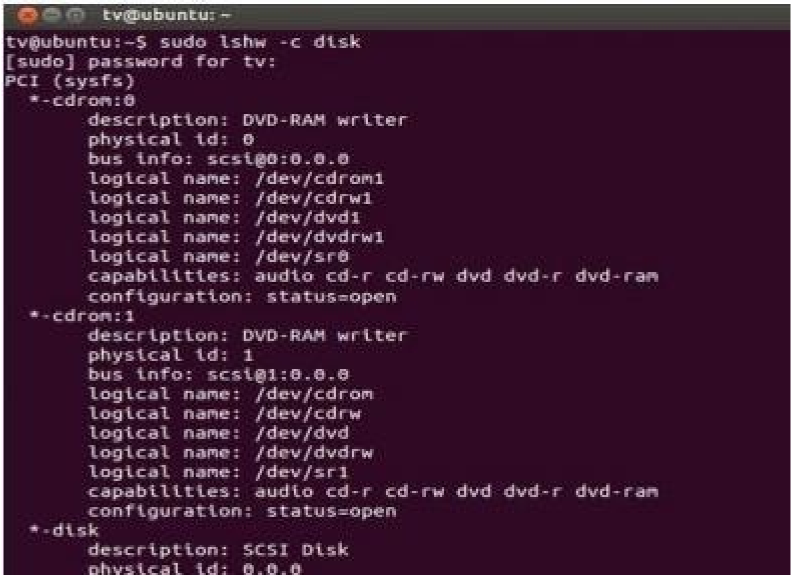
/usr contains the majority of user utilities and applications, and partly replicates the root directory structure, containing for instance, among others, /usr/bin/ and /usr/lib.

/var is dedicated variable data that potentially changes rapidly; a notable directory it contains is /var/log where system log files are kept. **Steps to Partition HardDisk Drive in Ubuntu:-**

Step 1. If you are trying to format or partition your hard drive it is assumed that bios is able to detect the device. To determine the path and other specific information about your drive open a terminal window and enter this command:

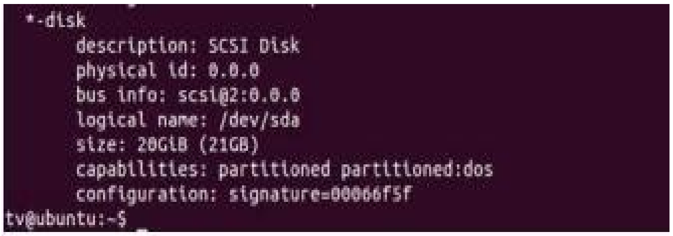
sudo lshw -C disk

Step 2. After entering this command Ubuntu should return something similar to this. Take note of the "logical name" because this will be used throughout the partitioning process if done via terminal window**.**



```
tv@ubuntu:~$ sudo lshw -c disk
[sudo] password for tv:
PCI (sysfs)
  *-cdrom:0
       description: DVD-RAM writer
       physical id: 0
       bus info: scsi@0:0.0.0
       logical name: /dev/cdrom1
       logical name: /dev/cdrw1
       logical name: /dev/dvd1
       logical name: /dev/dvdrw1
       logical name: /dev/sr0
       capabilities: audio cd-r cd-rw dvd dvd-r dvd-ram
       configuration: status=open
  *-cdrom:1
       description: DVD-RAM writer
       physical id: 1
       bus info: scsi@1:0.0.0
       logical name: /dev/cdrom
       logical name: /dev/cdrw
       logical name: /dev/dvd
       logical name: /dev/dvdrw
       logical name: /dev/sr1
       capabilities: audio cd-r cd-rw dvd dvd-r dvd-ram
       configuration: status=open
  *-disk
       description: SCSI Disk
       physical id: 0.0.0
```

Step 3. The part we will be most concerned with will be the hard drive information that is displayed in the terminal window.

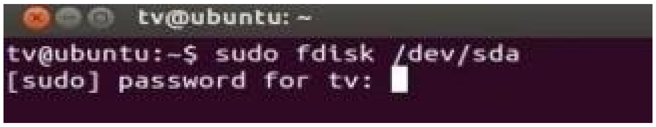


```
  *-disk
       description: SCSI Disk
       physical id: 0.0.0
       bus info: scsi@2:0.0.0
       logical name: /dev/sda
       size: 20GiB (21GB)
       capabilities: partitioned partitioned:dos
       configuration: signature=00066f5f
tv@ubuntu:~$
```

If you plan on using the hard drive only for Ubuntu then the recommended file system to use is either ext3/ext4 depending on whether or not you need backwards compatibility with previous versions of Linux. If you will need to share files between Ubuntu and Windows machines fat 32 is the recommended file system to use, but NTFS will also work well also.

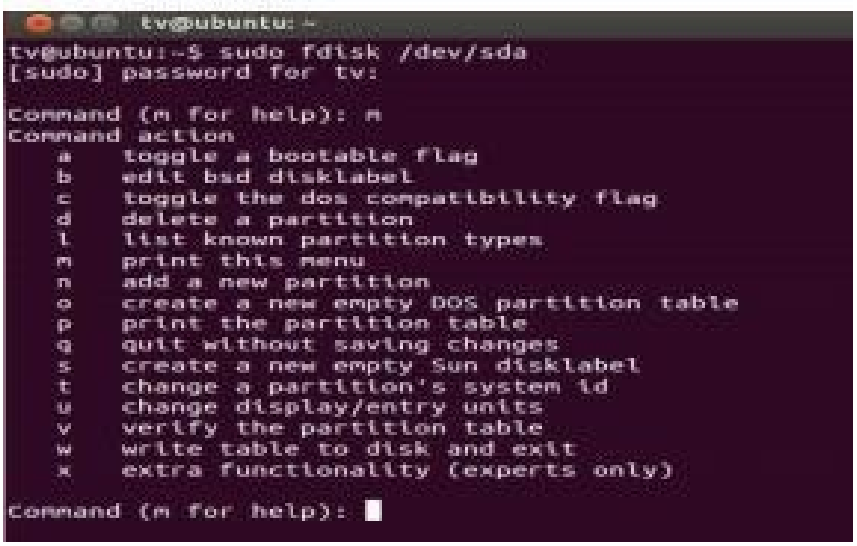
3.4.4 Partition using command line in Terminal: Step

1. Start fdisk with this command



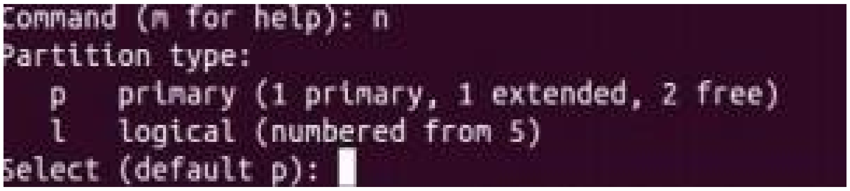
```
tv@ubuntu:~$ sudo fdisk /dev/sda
[sudo] password for tv:
```

Step 2. Press **"m"** then hit **enter**. This will return a menu like the one below showing all of the available commands for the fdisk program.



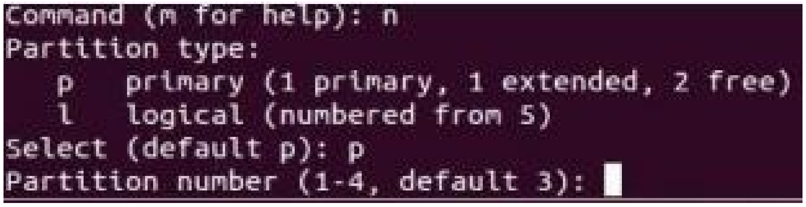
```
tv@ubuntu:~$ sudo fdisk /dev/sda
[sudo] password for tv:

Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)

Command (m for help):
```

Step 3. Since we want to add a new partition press **"n"** and then **enter**.



```
Command (m for help): n
Partition type:
   p   primary (1 primary, 1 extended, 2 free)
   l   logical (numbered from 5)
Select (default p):
```

Step 4. To create a primary partition (what we want) press **"p"** and then hit **enter**.



```
Command (m for help): n
Partition type:
   p   primary (1 primary, 1 extended, 2 free)
   l   logical (numbered from 5)
Select (default p): p
Partition number (1-4, default 3):
```

Step 5.

If you only want 1 partition press **"1"** and hit **enter**. You may be provided with a default response, you may choose this as the Partition number if you would like. Next you will be prompted for the locations of where you would like the first and last sectors of the partition to be. You may again be provided with default responses choose these if you want.

Step 6.

Now choose w to write the partition to the disk. Type "w" then press enter.Your drive is now partitioned. Now we need to format it. By default Linux will recognize this partition as dev/sdb1.

Step 7. To format the partition with an ext3 filesystem.

```
sudo mkfs -t ext3 /dev/sdb1
```

3.5 Algorithm:

1. Start
2. Initialize variables as path="/home/gurukul/.local/share/Trash/files"

   infopath="/home/gurukul/.local/share/Trash/info"
3. Check the list of files present in files folder.
4. Find the path of file to restore it using info folder.
5. Copy the contents of file which is deleted and is in Trash folder into new file at original location.
6. Delete the file from Trash folder.
7. End

3.6 Mathematical Model:

I= P (path of trash folder) **Functions:**
re.findall(r'/.*',line)
destipath.lstrip('[') destipath.rstrip(']')
destipath[:-1]
destipath[1:] **Output:**
R- Recovered file

Conclusion:

Hence we conclude that using **Forensics Application Program in Python we can recover Deleted Files.**

3.7 Assignment Questions:

1. What is Path of Trash folder in Ubuntu and what are the different folders?
2. How to see hidden files and filesystem of ubuntu?
3. What are different file systems in Ubuntu also state main directories of it??
4. How to list different files and what are the various options of ls used for file related function?