```
root@kali:~/Desktop# mmls -t dos dfr-11-mft-ntfs.dd
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

      Slot      Start        End          Length       Description
000:  Meta      0000000000   0000000000   0000000001   Primary Table (#0)
001:  -------   0000000000   0000000127   0000000128   Unallocated
002:  000:000   0000000128   0002091135   0002091008   NTFS / exFAT (0x07)
003:  -------   0002091136   0002097152   0000006017   Unallocated
```

1)Use mmls command to display the layout of the image. We can see there is only one partition in the image while start sector is 128 and end at 2091135 the length is 2091008.

```
root@kali:~/Desktop# dclff if=dfr-11-mft-ntfs.dd bs=512 skip=128 count=2091
008 of=ntfs.dd
bash: dclff: command not found
```

2)We capture the partition from 128 to 2091135

```
root@kali:~/Desktop# fsstat -f ntfs ntfs.dd
FILE SYSTEM INFORMATION
--------------------------------------------
File System Type: NTFS
Volume Serial Number: 2ACADB0FCADAD5E3
OEM Name: NTFS
Volume Name: ntfs
Version: Windows XP

METADATA INFORMATION
--------------------------------------------
First Cluster of MFT: 43562
First Cluster of MFT Mirror: 65343
Size of MFT Entries: 1024 bytes
Size of Index Records: 4096 bytes
Range: 0 - 64
Root Directory: 5

CONTENT INFORMATION
--------------------------------------------
Sector Size: 512
Cluster Size: 8192
Total Cluster Range: 0 - 130686
Total Sector Range: 0 - 2091006

$AttrDef Attribute Values:
$STANDARD_INFORMATION (16)   Size: 48-72   Flags: Resident
$ATTRIBUTE_LIST (32)   Size: No Limit   Flags: Non-resident
$FILE_NAME (48)   Size: 68-578   Flags: Resident,Index
$OBJECT_ID (64)   Size: 0-256   Flags: Resident
$SECURITY_DESCRIPTOR (80)   Size: No Limit   Flags: Non-resident
$VOLUME_NAME (96)   Size: 2-256   Flags: Resident
$VOLUME_INFORMATION (112)   Size: 12-12   Flags: Resident
$DATA (128)   Size: No Limit   Flags:
$INDEX_ROOT (144)   Size: No Limit   Flags: Resident
$INDEX_ALLOCATION (160)   Size: No Limit   Flags: Non-resident
$BITMAP (176)   Size: No Limit   Flags: Non-resident
$REPARSE_POINT (192)   Size: 0-16384   Flags: Non-resident
$EA_INFORMATION (208)   Size: 8-8   Flags: Resident
$EA (224)   Size: 0-65536   Flags:
$LOGGED_UTILITY_STREAM (256)   Size: 0-65536   Flags: Non-resident
```

3)fsstat command get the information of the image

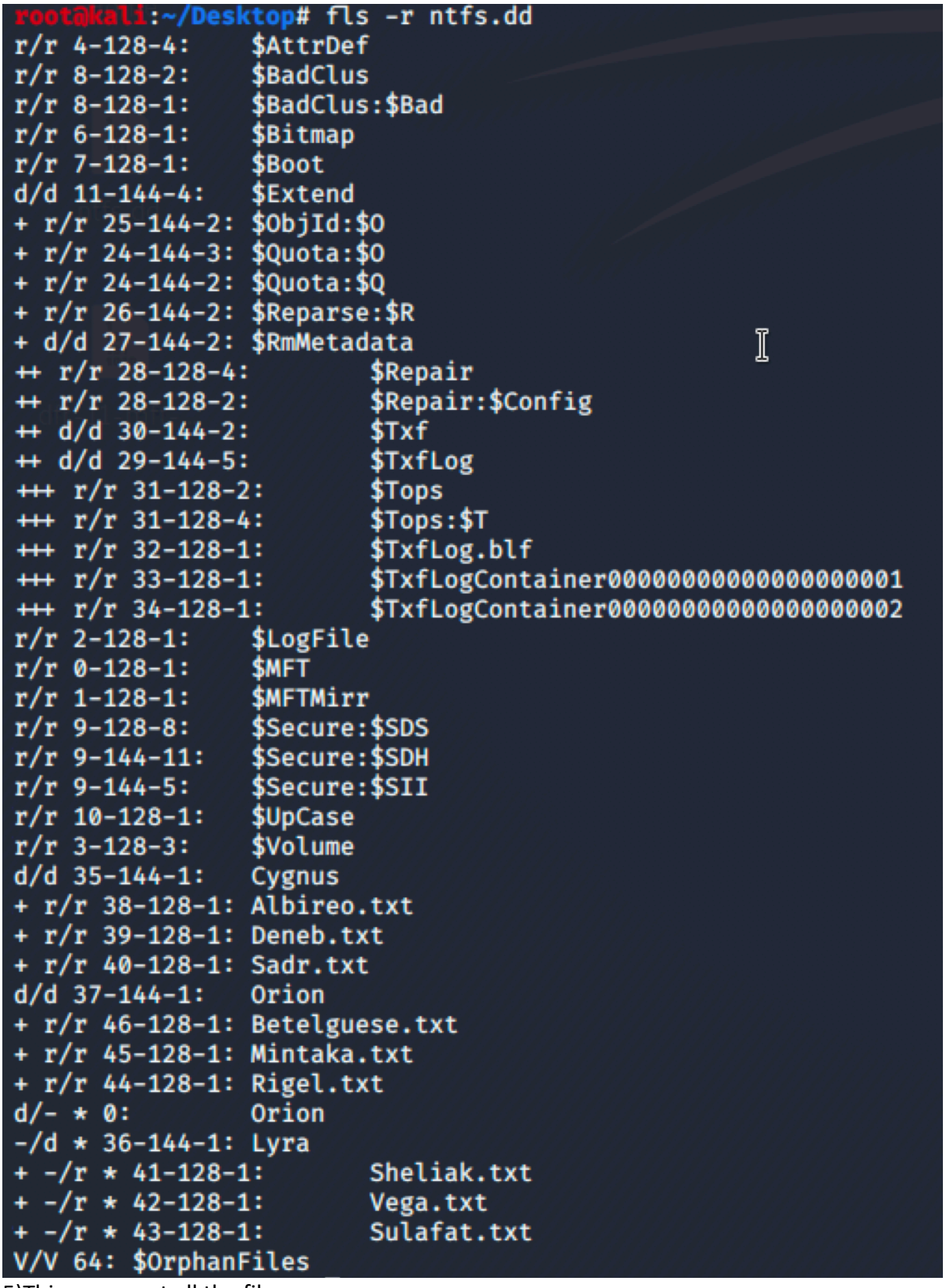
```
FLS(1)                          General Commands Manual                          FLS(1)

NAME
       fls - List file and directory names in a disk image.

SYNOPSIS
       fls [-adDFlpruvV] [-m mnt ] [-z zone ] [-f fstype ] [-s seconds ] [-i imgtype ] [-o imgoffset ] [-b dev_sector_size] image [images] [ inode ]

DESCRIPTION
       fls lists the files and directory names in the image and can display file names of recently deleted files for the directory using the given inode.  If the in-
       ode argument is not given, the inode value for the root directory is used. For example, on an NTFS file system it would be 5 and on  a  Ext3  file  system  it
       would be 2.
```

4) Because we don’t know which file has been delete so that we can not recover it. Therefore, fls command help us to identify which file has recently been deleted.



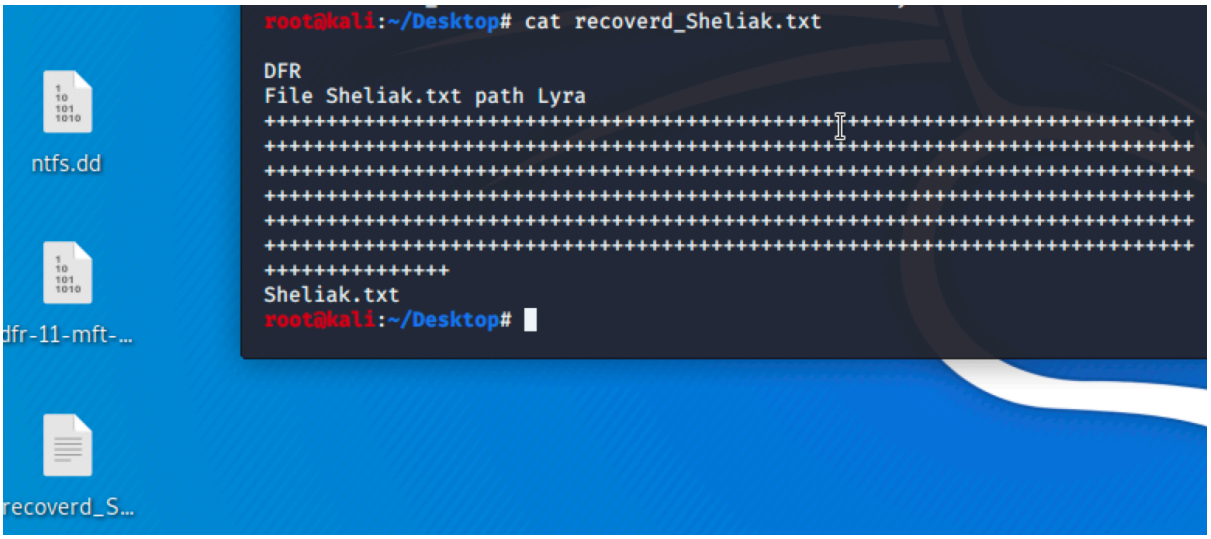
5)This represent all the file

```
root@kali:~/Desktop# fls -r -d ntfs.dd
d/- * 0:        Orion
-/d * 36-144-1: Lyra
-/r * 41-128-1: Lyra/Sheliak.txt
-/r * 42-128-1: Lyra/Vega.txt
-/r * 43-128-1: Lyra/Sulafat.txt
```

6)by using -d, we can find the one have been deleted

```
root@kali:~/Desktop# icat -r ntfs.dd 41 > recoverd_Sheliak.txt
```

7) icat command is used to recover the deleted file while -r mean using the recovery technic.



8)Successful, the file will store in your desktop which you can read it.