ASSIGNMENT 2  
DUE DATE: Tuesday 19 April, 2011 11:59 PM

This purpose of this assignment is to familiarize you with the basic structure of the NTFS file system.

Description

In this assignment, you will write a Java program called DiskPreview.java that outputs some basic information about the files contained within a disk image. The disk images provided to you have been formatted with NTFS, using a cluster size of 512 bytes (i.e. 1 sector per cluster).

DiskPreview will read the provided disk image, parse through it looking for MFT records (recall that each MFT record occupies 1024 bytes) and output the following information about all file-related entries it encounters. You can ignore folder related information for the time being.

The output information is displayed using the following format.

NAME:
  \( \text{CN :: } \text{FBCN}(+\text{OFF}) [:: \text{NDS}] \)

where,

- NAME = name of the file
- CN = cluster no. where MFT record for the file is located
- FBCN = cluster no. where contents of the file begin
- OFF = offset within cluster FBCN to the first byte of the file
- NDS = number of alternate data streams found in the file, if any

Important Note: You should display this information only if the MFT record has both a file name attribute (0x30) and one or more data attributes (0x80). If either of them is not present, then you can ignore that record.

Example:

afile.txt:
  374465 :: 374465(+288) :: 2

This means – there is a MFT record in cluster 374465 that has information about the file afile.txt. The data for this file begins from byte 288 onwards in cluster no. 374465. This is a resident file since non-resident data typically begins from offset zero within the cluster! Also, this file has two other data streams. The picture will be more
complete if you can also report the starting locations of these streams; but you are not required to do that as part of this assignment.

**Approach**

You should first go over the slides in Lecture 5 to refresh how a MFT record is structured. Some of the concepts that will be needed are:

- how is the MFT header structured?
- how are attributes 0x30 and 0x80 structured?

Second, determine how a given number of bytes can be read from file. You may refer to the `DataInputStream` class. Methods in this class will allow you to read a specified number of bytes from a file.

After skipping the first 16 sectors (8192 bytes), the program should read 1024 bytes at a time and check if it corresponds to an MFT record (check the magic number). If so, then you will have to extract the relevant information. For this, use your knowledge about how information is coded in the header and the following attributes. Remember, output should be displayed only if the record has both the filename and data attributes.

**Useful Coding Tips**

a. Did you know that Java has a data type called `byte`? You can read from the file into a `byte` array, say into something like

   ```java
   byte record[] = new byte[1024];
   ```

b. When comparing a byte to a certain hex value, it is a good idea to ‘AND’ it with 0x000000FF. This ensures that the byte does not get converted into a signed integer before comparison. E.g. use `record[i] & 0x000000FF` instead of just `record[i]`.

**Images**

Two image files (`USB1.dd` and `USB2.dd`) are provided in the assignment page. Output on the `USB1.dd` file is shown on the last page. The tree structure on the `USB2.dd` image as seen in Windows Explorer is also given.

**Submission**

Upload the `DiskPreview.java` file to Blackboard.
Grading

The assignment is worth 10 points. A program that does not compile is a program that you did not submit at all. Remember the GTA is not required to debug your program to give you partial points.

The late policy is available at http://cs.du.edu/2555/assignments.html. You must work alone on this assignment.

USB2.dd as seen in Windows Explorer

|------- Music |
|--------- Kalimba.mp3 (8218 KB) |
|--------- Maid with the Flazen Hair.mp3 (4018 KB) |

|------- Slides |
|--------- Lecture 1.pptx (752 KB) |
|--------- Lecture 2.pptx (134 KB) |
|--------- Lecture 3.pptx (162 KB) |

|------- Empty.txt (0 KB) |
|------- fractal.gif (217 KB) |
|------- MD5.txt (1 KB) |
|------- NTFS.txt (4 KB) |

Sample output on USB1.dd in next page.
> javac DiskPreview.java
> java DiskPreview USB1.dd

$MFT:
  16 :: 326634(+0)

$MFTMirr:
  18 :: 16(+0)

$LogFile:
  20 :: 308330(+0)

$Volume:
  22 :: 22(+488)

$MFT:
  326634 :: 326634(+0)

$MFTMirr:
  326636 :: 16(+0)

$LogFile:
  326638 :: 308330(+0)

$Volume:
  326640 :: 326640(+488)

$AttrDef:
  326642 :: 317867(+0)

$Bitmap:
  326646 :: 326378(+0)

$Boot:
  326648 :: 0(+0)

$BadClus:
  326650 :: 326650(+288) :: 1

$Secure:
  326652 :: 21248(+0)

$UpCase:
  326654 :: 24(+0)

$Repair:
  326690 :: 0(+0) :: 1

$Tops:
  326696 :: 326696(+280) :: 1

$TxfLog.blf:
  326698 :: 319921(+0)

$TxfLogContainer000000000000000000000001:
  326700 :: 359(+0)

$TxfLogContainer000000000000000000000002:
  326702 :: 20839(+0)

testfile.txt:
  326704 :: 288(+0)

usb.txt:
  326706 :: 326706(+280) :: 2

>