Operating Systems
CSCI 5806

Spring Semester 2025 — CRN 22968

Term Project — Step 2 — Disk Partition Access

Target completion date: Monday, February 10, 2025

Goals

- Provide the five basic file I/O functions to access disk space inside a disk partition, which is contained in a VDI file.
- Create a structure or class to contain the data necessary to implement the five functions.

Details

As with the lower-level VDI file, you'll want to set up a basic structure or class to hold the data necessary to work with partitions. You will need to maintain the following information:

- A VDI file; open it as part of the partition's `open( )` function, close it when you "close" the partition.
- A partition table with four entries (the standard size for an MBR-based partition table).
- The offset (start) of the partition, measured in bytes from the start of the disk space. This is stored in the partition table entries.
- The size of the partition, measured in bytes. This is also stored in the partition table entries.
- A cursor, indicating the location within the partition of the next byte to be read or written.

Note: The partition table entries store the size and start of each partition in units of 512-byte sectors, not in bytes. Multiply by 512 to get the proper values.

Wikipedia has a good article on Master Boot Records (MBRs) at https://en.wikipedia.org/wiki/Master_boot_record; it has all of the information you need to extract the necessary data for this step.

In addition to the structure, you'll need to implement the five basic file I/O functions:

- **`struct MBRPartition *open(char *fn,int part)`**
  Open the given VDI file and use the given partition number. Return a pointer to the structure you've created.
- **`void close(struct MBRPartition *f)`**
  Close the partition. Deallocate any dynamically created memory regions.
- **`ssize_t read(struct MBRPartition *f,void *buf,ssize_t count)`**
  Read bytes from the partition. Restrict **`count`** so that it does not read beyond the end of the partition.

- **`ssize_t write(struct MBRPartition *f,void *buf,size_t count)`**
  Write bytes to the partition. Restrict **`count`** so that it does not write beyond the end of the partition.
- **`ssize_t lseek(struct MBRPartition *f,ssize_t offset,int whence)`**
  Set the cursor within the partition. Restrict the function so that the cursor remains unchanged if a location outside the partition is requested.

If you are using a class, then the `MBRPartition *` parameter is omitted.

Write a function that takes a partition table entry structure and displays its fields in an easy-to-read manner. Again, your exact format may differ somewhat from my example.

Example 1

This is the output from my step 2 program, on the dynamic VDI file with 1KB blocks. It shows the four entries in the partition table. It then reads a 1KB block from the disk, starting at an offset of 1024. This is displayed using the **`displayBuffer()`** function.

Spoiler alert: That 1KB block is called the *superblock*, and it's *really* important, so it's a critical check here that your program is reading the same bytes you're seeing in this output.

```
Partition table entry 0:
Status: Inactive
First sector CHS: 4-4-1
Last sector CHS: 514-1-2
Partition type: 83 linux native
First LBA sector: 2048
LBA sector count: 260096

Partition table entry 1:
Status: Inactive
First sector CHS: 0-0-0
Last sector CHS: 0-0-0
Partition type: 00 empty
First LBA sector: 0
LBA sector count: 0

Partition table entry 2:
Status: Inactive
First sector CHS: 0-0-0
Last sector CHS: 0-0-0
Partition type: 00 empty
First LBA sector: 0
LBA sector count: 0

Partition table entry 3:
Status: Inactive
First sector CHS: 0-0-0
Last sector CHS: 0-0-0
Partition type: 00 empty
First LBA sector: 0
LBA sector count: 0

Superblock:
Offset: 0x400
   00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f    0...4...8...c...
  +-----------------------------------------------+  +----------------+
00|00 7f 00 00 00 fc 01 00 66 19 00 00 ef 53 00 00|00|        f    S  |
10|af 7d 00 00 01 00 00 00 00 00 00 00 00 00 00 00|10| }              |
20|00 20 00 00 00 20 00 00 f0 07 00 00 40 e1 05 60|20|           @  `|
30|d5 e1 05 60 02 00 ff ff 53 ef 01 00 01 00 00 00|30|   `    S       |
40|0c dd 05 60 00 00 00 00 00 00 00 00 01 00 00 00|40|   `            |
50|00 00 00 00 0b 00 00 00 80 00 00 00 38 00 00 00|50|            8   |
60|02 00 00 00 03 00 00 00 35 66 c4 a2 e7 b3 45 f7|60|        5f    E |
70|91 71 3a 8d f3 ed c7 62 00 00 00 00 00 00 00 00|70| q:    b        |
```

```
80|00 00 00 00 00 00 00 00 2f 6d 65 64 69 61 2f 63|80|        /media/c|
90|73 69 73 2f 33 35 36 36 63 34 61 32 2d 65 37 62|90|sis/3566c4a2-e7b|
a0|33 2d 34 35 66 37 2d 39 31 37 31 2d 33 61 38 64|a0|3-45f7-9171-3a8d|
b0|66 33 65 64 63 37 36 32 00 00 00 00 00 00 00 00|b0|f3edc762        |
c0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01|c0|                |
d0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|d0|                |
e0|00 00 00 00 00 00 00 00 00 00 00 00 fa 72 39 d4|e0|             r9 |
f0|ae e0 4d 83 95 b5 6a 9c cc 93 6a 56 01 00 00 00|f0|  M   j   jV    |
  +-----------------------------------------------+  +----------------+

Offset: 0x500
   00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f    0...4...8...c...
  +-----------------------------------------------+  +----------------+
00|0c 00 00 00 00 00 00 00 0c dd 05 60 00 00 00 00|00|           `    |
10|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|10|                |
20|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|20|                |
30|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|30|                |
40|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|40|                |
50|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|50|                |
60|01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|60|                |
70|00 00 00 00 00 00 00 00 a2 92 01 00 00 00 00 00|70|                |
80|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|80|                |
90|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|90|                |
a0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|a0|                |
b0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|b0|                |
c0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|c0|                |
d0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|d0|                |
e0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|e0|                |
f0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|f0|                |
  +-----------------------------------------------+  +----------------+

Offset: 0x600
   00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f    0...4...8...c...
  +-----------------------------------------------+  +----------------+
00|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|00|                |
10|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|10|                |
20|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|20|                |
30|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|30|                |
40|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|40|                |
50|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|50|                |
60|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|60|                |
70|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|70|                |
80|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|80|                |
90|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|90|                |
a0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|a0|                |
b0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|b0|                |
c0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|c0|                |
d0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|d0|                |
e0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|e0|                |
f0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|f0|                |
  +-----------------------------------------------+  +----------------+

Offset: 0x700
   00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f    0...4...8...c...
```

```
  +-------------------------------------------------+  +----------------+
00|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|00|                |
10|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|10|                |
20|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|20|                |
30|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|30|                |
40|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|40|                |
50|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|50|                |
60|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|60|                |
70|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|70|                |
80|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|80|                |
90|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|90|                |
a0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|a0|                |
b0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|b0|                |
c0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|c0|                |
d0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|d0|                |
e0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|e0|                |
f0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|f0|                |
  +-------------------------------------------------+  +----------------+
```

Example 2

This is the program's output using the fixed-size VDI file with 4KB block size.

```
Partition table entry 0:
Status: Inactive
First sector CHS: 4-4-1
Last sector CHS: 514-1-2
Partition type: 83 linux native
First LBA sector: 2048
LBA sector count: 260096

Partition table entry 1:
Status: Inactive
First sector CHS: 0-0-0
Last sector CHS: 0-0-0
Partition type: 00 empty
First LBA sector: 0
LBA sector count: 0

Partition table entry 2:
Status: Inactive
First sector CHS: 0-0-0
Last sector CHS: 0-0-0
Partition type: 00 empty
First LBA sector: 0
LBA sector count: 0

Partition table entry 3:
Status: Inactive
First sector CHS: 0-0-0
Last sector CHS: 0-0-0
Partition type: 00 empty
```

```
First LBA sector: 0
LBA sector count: 0

Superblock:
Offset: 0x400
   00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f    0...4...8...c...
  +-----------------------------------------------+  +----------------+
00|00 7f 00 00 00 7f 00 00 59 06 00 00 f2 15 00 00|00|        Y       |
10|af 7d 00 00 00 00 00 00 02 00 00 00 02 00 00 00|10| }              |
20|00 80 00 00 00 80 00 00 00 7f 00 00 43 e1 05 60|20|            C  ‘|
30|d5 e1 05 60 02 00 ff ff 53 ef 01 00 01 00 00 00|30|   ‘    S       |
40|be dd 05 60 00 00 00 00 00 00 00 00 01 00 00 00|40|   ‘            |
50|00 00 00 00 0b 00 00 00 80 00 00 00 38 00 00 00|50|            8   |
60|02 00 00 00 03 00 00 00 f6 27 9c 20 d5 85 44 62|60|         ’    Db|
70|86 30 ce a9 9f 2d 3d 2e 00 00 00 00 00 00 00 00|70| 0   -=.        |
80|00 00 00 00 00 00 00 00 2f 6d 65 64 69 61 2f 63|80|        /media/c|
90|73 69 73 2f 66 36 32 37 39 63 32 30 2d 64 35 38|90|sis/f6279c20-d58|
a0|35 2d 34 34 36 32 2d 38 36 33 30 2d 63 65 61 39|a0|5-4462-8630-cea9|
b0|39 66 32 64 33 64 32 65 00 00 00 00 00 00 00 00|b0|9f2d3d2e        |
c0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 07 00|c0|                |
d0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|d0|                |
e0|00 00 00 00 00 00 00 00 00 00 00 00 56 c9 82 d6|e0|            V   |
f0|cd 2b 42 ac a9 83 b8 e1 fc 82 cd 5b 01 00 00 00|f0| +B        [    |
  +-----------------------------------------------+  +----------------+

Offset: 0x500
   00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f    0...4...8...c...
  +-----------------------------------------------+  +----------------+
00|0c 00 00 00 00 00 00 00 be dd 05 60 00 00 00 00|00|           ‘    |
10|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|10|                |
20|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|20|                |
30|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|30|                |
40|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|40|                |
50|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|50|                |
60|01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|60|                |
70|00 00 00 00 00 00 00 00 e4 94 01 00 00 00 00 00|70|                |
80|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|80|                |
90|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|90|                |
a0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|a0|                |
b0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|b0|                |
c0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|c0|                |
d0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|d0|                |
e0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|e0|                |
f0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|f0|                |
  +-----------------------------------------------+  +----------------+

Offset: 0x600
   00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f    0...4...8...c...
  +-----------------------------------------------+  +----------------+
00|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|00|                |
10|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|10|                |
20|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|20|                |
30|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|30|                |
40|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|40|                |
```

```
50|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|50|                 |
60|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|60|                 |
70|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|70|                 |
80|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|80|                 |
90|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|90|                 |
a0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|a0|                 |
b0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|b0|                 |
c0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|c0|                 |
d0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|d0|                 |
e0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|e0|                 |
f0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|f0|                 |
  +-----------------------------------------------+  +----------------+

Offset: 0x700
   00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f    0...4...8...c...
  +-----------------------------------------------+  +----------------+
00|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|00|                 |
10|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|10|                 |
20|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|20|                 |
30|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|30|                 |
40|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|40|                 |
50|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|50|                 |
60|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|60|                 |
70|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|70|                 |
80|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|80|                 |
90|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|90|                 |
a0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|a0|                 |
b0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|b0|                 |
c0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|c0|                 |
d0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|d0|                 |
e0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|e0|                 |
f0|00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00|f0|                 |
  +-----------------------------------------------+  +----------------+
```

Example output for all six test files is available in the repository in the file `step2.log`.