Innopolis University

SYSTEM AND NETWORKING ENGINEERING

Classical Internet Applications

LABORATORY REPORT 1

Booting(1)

Student Name	Student ID
Sergey Grebennikov	47611

Lecturer:

Rasheed Hussain

Submission Date : August 25, 2017

Contents

1 Introduction

This laboratory work introduces concepts as booting options and disk partitions. It consists of two parts:

1. Questions

2. Partitions

2 Main Part

2.1 Questions

1. UEFI PXE

   (a) What is **UEFI PXE** booting?

   Answer:

   It is possibility of Unified Extensible Firmware Interface (UEFI) to boot over network an image of operating system using Preboot eXecution Environment (PXE) [1].

   (b) How does it work?

   Answer:

   UEFI PXE booting uses network protocols (IPv4, IPv6, UDP, DHCP and TFTP) to deploy an Operating System (OS) on clients from a server. The PXE-enabled client broadcasts the DHCPDISCOVER packet to the PXE-enabled DHCP server (the standard DHCP servers don't support PXE-specific options). The DHCP server provides the client with the network settings, the location of the TFTP server, the Network Bootstrap Program (NBP), and optional files. The client sets its own network options and refers to the network located booting resource (IP address of TFTP and name of the NBP). The client puts the NBP into its own random-access memory using TFTP, and finally boots from it. NBP allows loading a small OS executive (such as WindowsPE, or Linux-kernel with initrd), which install the necessary protocols to download the full OS [2].

   (c) How does it compare to booting from the hard disk or a CD?

   Answer:

   Firstly, this difference in the environment of data transmission. Secondly, PXE is the fastest way to install an operating system on a client as compared to a hard disk or a CD. [1]

2. GPT

   (a) What is a **GPT**?

   Answer:

   GPT stands for GUID Partition Table. It is a standard for the layout of the partition table on a physical storage device (HDD or SSD) using globally unique identifiers (GUID). GPT is a new disk layout associated with UEFI. [3]

   (b) What is its layout?

   Answer:

   Basic GPT disk layout consists of logical blocks each of 512 bytes. The first block (LBA 0) is Protective MBR. The second block (LBA 1) contains Primary GPT Header to define the range of logical block addresses that are usable by partition entries. Starting with LBA2, there's a partition entry array, which is 128 bytes per partition entry (4 entries per 1 LBA). Maximum support 128 partition entries will need 32 LBAs, which make LBA34 be the first usable sector on the disk. The last LBA contains the backup GPT header followed by a backup GPT partition entry array. [3] [4] [5]

   (c) Why would you want to partition a disk?

   Answer:

Partitioning a disk uses for some reasons:

i. To separate user files from operating system and program files
ii. To install multiple operating system on the same computer
iii. To have a separate area for virtual memory
iv. To raise overall computer performance on systems where smaller file systems are more efficient

3. gdisk

(a) What is **gdisk**?

Answer:

gdisk is utility that means GPT fdisk. gdisk allows to create and manipulate partition tables. [6]

(b) How does it work?

Answer:

It convert old MBR partition table to newer GPT format or load GPT. [6]

(c) What can you do with it?

Answer:

With gdisk utility we can do some useful things: [6]

i. Save partition data to backup file
ii. Change the GPT name of a partition
iii. Delete a partition
iv. Show detailed partition information
v. Create a new partition
vi. Rebuild GPT header from backup
vii. Load main partition table
viii. Load MBR and build fresh GPT from it
ix. Convert GPT into MBR
x. Verify disk
xi. Change partition GUID
xii. Create a new protective MBR and print it
xiii. Print the partition table
xiv. other

4. Protective MBR

(a) What is a **Protective MBR** and why is it in the **GPT**?

Answer:

Protective MBR is the special space reserved in the GPT specification. It protects GPT disks from overwriting with MBR-based disk utilities. [3]

2.2 Partitions

1. Boot into Ubuntu and use the dd utility to dump the Protective MBR and GPT into a file in your home directory.

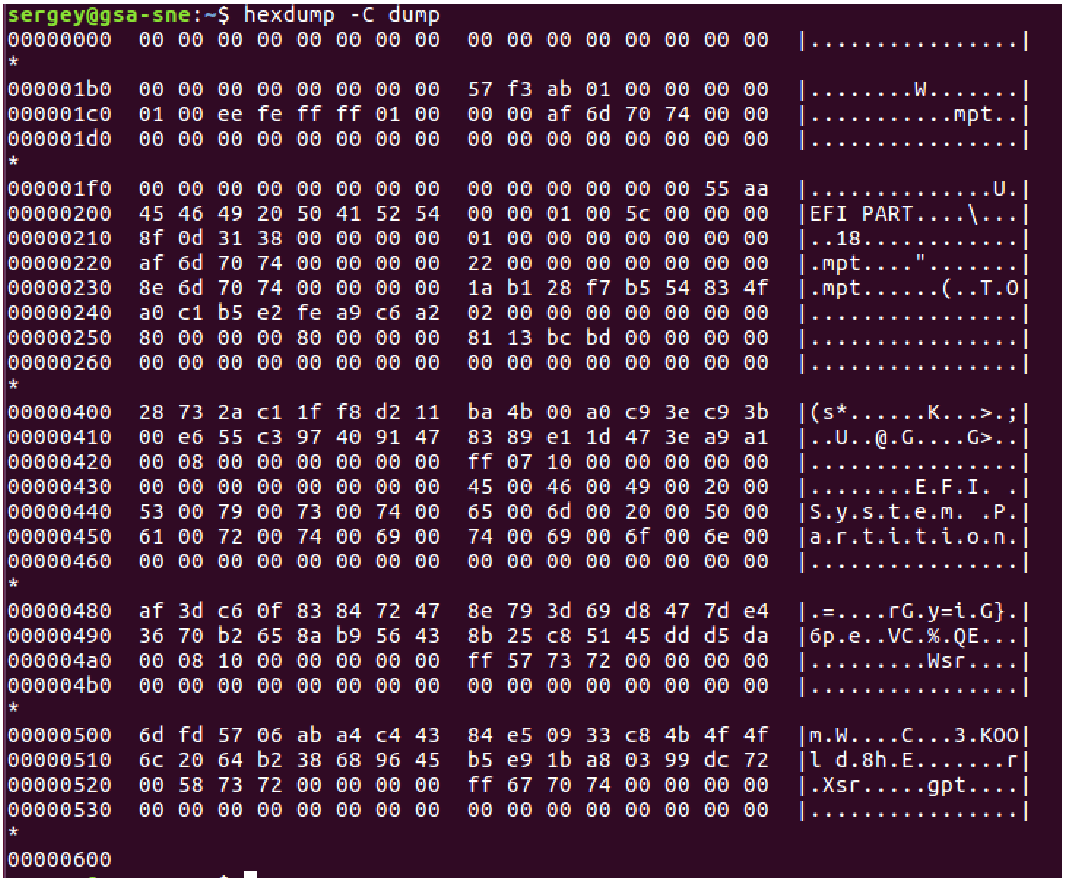
Solution:

The first three logical block addresses (LBA) of 512 bytes should be read. The first LBA (LBA0) is the Protective MBR, the second one (LBA1) is the Primary GPT Header, and the third (LBA2) is the partition table (or Partition Entry Array).

```
sudo dd if=/dev/sda of=~/dump bs=512 count=3
```

2. Use a hex dump utility to look at the raw data in the file (Figure 1).

   Result:



```
sergey@gsa-sne:~$ hexdump -C dump
00000000  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
000001b0  00 00 00 00 00 00 00 00  57 f3 ab 01 00 00 00 00  |........W.......|
000001c0  01 00 ee fe ff ff 01 00  00 00 af 6d 70 74 00 00  |...........mpt..|
000001d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
000001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
00000200  45 46 49 20 50 41 52 54  00 00 01 00 5c 00 00 00  |EFI PART....\...|
00000210  8f 0d 31 38 00 00 00 00  01 00 00 00 00 00 00 00  |..18............|
00000220  af 6d 70 74 00 00 00 00  22 00 00 00 00 00 00 00  |.mpt....".......|
00000230  8e 6d 70 74 00 00 00 00  1a b1 28 f7 b5 54 83 4f  |.mpt......(..T.O|
00000240  a0 c1 b5 e2 fe a9 c6 a2  02 00 00 00 00 00 00 00  |................|
00000250  80 00 00 00 80 00 00 00  81 13 bc bd 00 00 00 00  |................|
00000260  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000400  28 73 2a c1 1f f8 d2 11  ba 4b 00 a0 c9 3e c9 3b  |(s*......K...>.;|
00000410  00 e6 55 c3 97 40 91 47  83 89 e1 1d 47 3e a9 a1  |..U..@.G....G>..|
00000420  00 08 00 00 00 00 00 00  ff 07 10 00 00 00 00 00  |................|
00000430  00 00 00 00 00 00 00 00  45 00 46 00 49 00 20 00  |........E.F.I. .|
00000440  53 00 79 00 73 00 74 00  65 00 6d 00 20 00 50 00  |S.y.s.t.e.m. .P.|
00000450  61 00 72 00 74 00 69 00  74 00 69 00 6f 00 6e 00  |a.r.t.i.t.i.o.n.|
00000460  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000480  af 3d c6 0f 83 84 72 47  8e 79 3d 69 d8 47 7d e4  |.=....rG.y=i.G}.|
00000490  36 70 b2 65 8a b9 56 43  8b 25 c8 51 45 dd d5 da  |6p.e..VC.%.QE...|
000004a0  00 08 10 00 00 00 00 00  ff 57 73 72 00 00 00 00  |.........Wsr....|
000004b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000500  6d fd 57 06 ab a4 c4 43  84 e5 09 33 c8 4b 4f 4f  |m.W....C...3.KOO|
00000510  6c 20 64 b2 38 68 96 45  b5 e9 1b a8 03 99 dc 72  |l d.8h.E.......r|
00000520  00 58 73 72 00 00 00 00  ff 67 70 74 00 00 00 00  |.Xsr.....gpt....|
00000530  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000600
```

Figure 1: Using hexdump utility

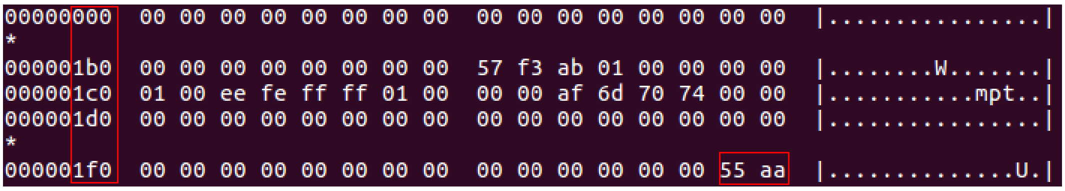
3. Describe the purpose of every field, and translate all fields that have a numerical value into human readable, decimal format.

   (a) **Protective MBR**

   The first address block is Protective MBR. It takes the first sector of HDD. The sector size is **512** bytes (address space from **0x00** to **0x01ff** or from **0** to **511** in the human readable format). The first 446 bytes (from 0x0000 to 0x01bd) in the Protective MBR is the bootstrap code area. Then there are four partition entries each of 16 bytes. In accordance with the specification, the MBR partition table scheme ends with value **55 aa** (Figure 2).

   (b) **The Primary GPT Header**

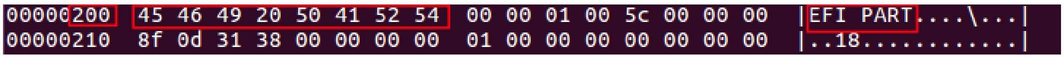
   The following sector is started with 512 byte (0x0200) and contains the Primary GPT Header. The first 8 bytes describe the signature (”EFI PART”, 45h 46h 49h 20h 50h 41h 52h 54h)(Figure 3). The partition table header defines the usable blocks on the disk, such as the number and size of the partition entries,



```
00000000  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
000001b0  00 00 00 00 00 00 00 00  57 f3 ab 01 00 00 00 00  |........W.......|
000001c0  01 00 ee fe ff ff 01 00  00 00 af 6d 70 74 00 00  |...........mpt..|
000001d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
000001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

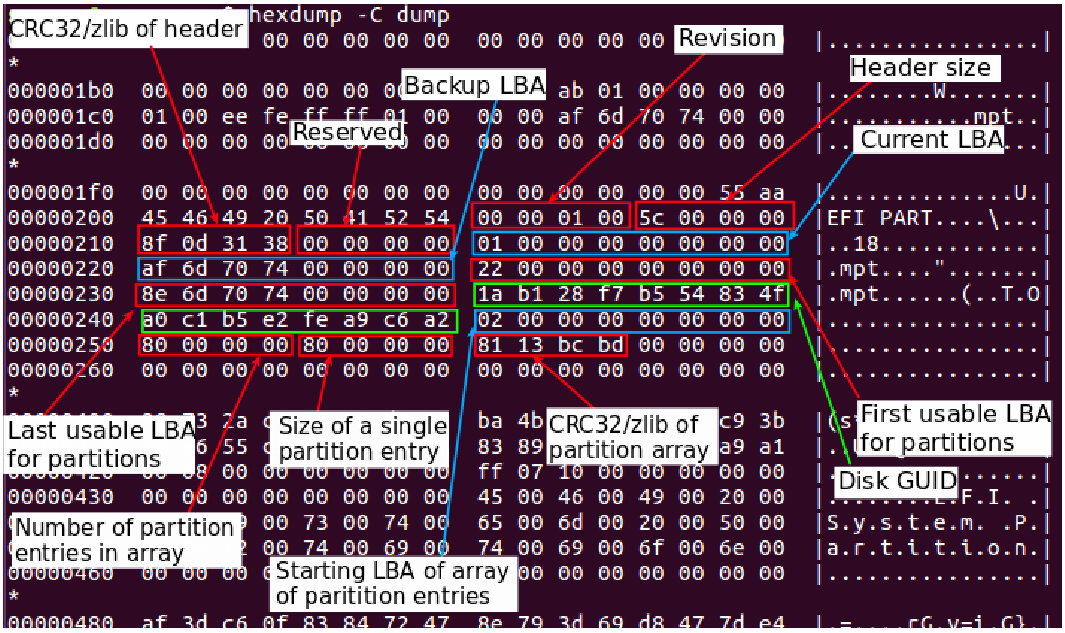
Figure 2: Protective MBR

the disk GUID, own size and location, location of the secondary GPT header, Cyclic Redundancy Check (CRC32), location of this header copy (current LBA), location of the other header copy (backup LBA) (Figure 4).



```
00000200  45 46 49 20 50 41 52 54  00 00 01 00 5c 00 00 00  |EFI PART....\...|
00000210  8f 0d 31 38 00 00 00 00  01 00 00 00 00 00 00 00  |..18............|
```

Figure 3: Primary GPT Header



```
hexdump -C dump
          00 00 00 00 00  00 00 00 00 00                    |................|
*
000001b0  00 00 00 00 00 00 00            ab 01 00 00 00 00  |........W.......|
000001c0  01 00 ee fe ff ff 01 00  00 00 af 6d 70 74 00 00  |...........mpt..|
000001d0  00 00 00 00          00  00 00 00 00 00 00 00 00  |..             ...|
*
000001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
00000200  45 46 49 20 50 41 52 54  00 00 01 00 5c 00 00 00  |EFI PART....\...|
00000210  8f 0d 31 38 00 00 00 00  01 00 00 00 00 00 00 00  |..18............|
00000220  af 6d 70 74 00 00 00 00  22 00 00 00 00 00 00 00  |.mpt....".......|
00000230  8e 6d 70 74 00 00 00 00  1a b1 28 f7 b5 54 83 4f  |.mpt......(..T.O|
00000240  a0 c1 b5 e2 fe a9 c6 a2  02 00 00 00 00 00 00 00  |................|
00000250  80 00 00 00 80 00 00 00  81 13 bc bd 00 00 00 00  |................|
00000260  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
             3 2a                   ba 4b              c9 3b  |(s
             6 55                   83 89              a9 a1  |.
                8 00 00 00 00 00 00  ff 07 10 00 00 00 00 00  |.              .....|
00000430  00 00 00 00 00 00 00 00  45 00 46 00 49 00 20 00  |........E.F.I. .|
                   00 73 00 74 00  65 00 6d 00 20 00 50 00  |S.y.s.t.e.m. .P.|
                   00 74 00 69 00  74 00 69 00 6f 00 6e 00  |a.r.t.i.t.i.o.n.|
00000460  00 00 00                    00 00 00 00 00 00 00  |................|
*
00000480  af 3d c6 0f 83 84 72 47  8e 79 3d 69 d8 47 7d e4  |.=....rG.y=i.G}.|
```

CRC32/zlib of header
Revision
Backup LBA
Header size
Reserved
Current LBA
Last usable LBA for partitions
Size of a single partition entry
CRC32/zlib of partition array
First usable LBA for partitions
Disk GUID
Number of partition entries in array
Starting LBA of array of paritition entries

Figure 4: Field values of the Primary GPT Header

The values of block in the human readable format:

Header size: 92 bytes (0x5c)

CRC32/zlib of header: 942 738 831 (0x8f0d3138)

Current LBA: 1 (0x0100000000000000)

Backup LBA: 1 953 525 167 (0xaf6d707400000000)

First usable LBA for partitions: 34 (0x2200000000000000)

Last usable LBA: 1 953 525 134 (0x8e6d707400000000)

Disk GUID: Disk identifier: F728B11A-54B5-4F83-A0C1-B5E2FEA9C6A2 (0x1ab128f7b55 0xa0c1b5e2fea9c6a2)

Starting LBA of array of partition entries: 2 (0x0200000000000000)

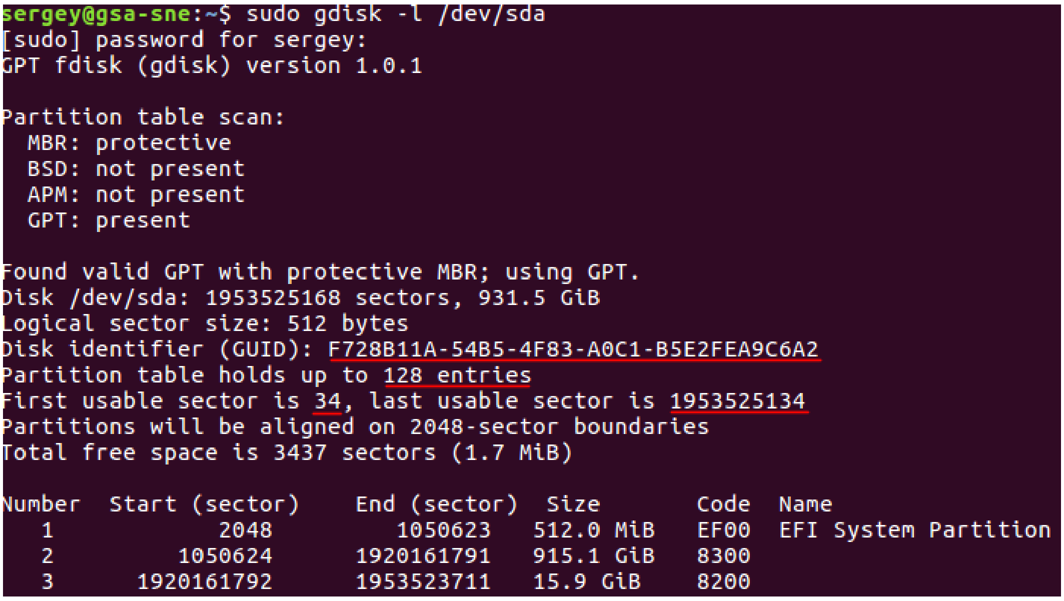
Number of partition entries in array: 128 (0x80000000)

Size of a single partition entry: 128 bytes (0x80000000)

CRC32/zlib of partition array: 3 183 219 585 in UINT32 and -1 111 747 711 in INT32 (0x8113bcbd00000000)

Reserved: The following 420 bytes are filled with zeros. This value is obtained by subtracting 92 bytes of header size from 512 bytes of block size (0x0400 - 0x025c = 0x014a).

This information can be verified using the **gdisk** utility (Figure 5).

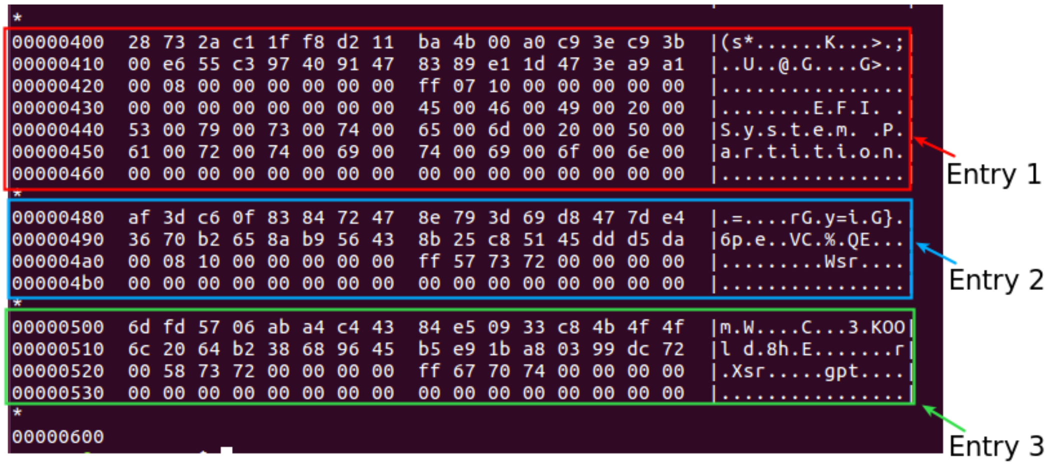


```
sergey@gsa-sne:~$ sudo gdisk -l /dev/sda
[sudo] password for sergey:
GPT fdisk (gdisk) version 1.0.1

Partition table scan:
  MBR: protective
  BSD: not present
  APM: not present
  GPT: present

Found valid GPT with protective MBR; using GPT.
Disk /dev/sda: 1953525168 sectors, 931.5 GiB
Logical sector size: 512 bytes
Disk identifier (GUID): F728B11A-54B5-4F83-A0C1-B5E2FEA9C6A2
Partition table holds up to 128 entries
First usable sector is 34, last usable sector is 1953525134
Partitions will be aligned on 2048-sector boundaries
Total free space is 3437 sectors (1.7 MiB)

Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048         1050623   512.0 MiB   EF00  EFI System Partition
   2         1050624      1920161791   915.1 GiB   8300
   3      1920161792      1953523711   15.9 GiB    8200
```

Figure 5: Using the **gdisk** utility to verify values from the Primary GPT Header

(c) **Partition Entry Array**

After the Header follows the Partition Entry Array that describes partitions using the entry blocks. The information about the size of the entry and the starting location of the array are given in the GPT Header (Figure 6).



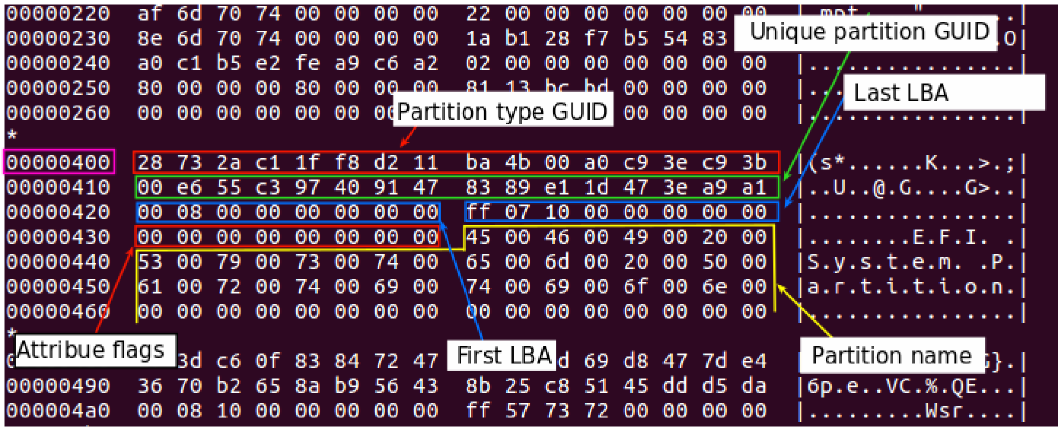
```
*
00000400  28 73 2a c1 1f f8 d2 11  ba 4b 00 a0 c9 3e c9 3b  |(s*......K...>.;
00000410  00 e6 55 c3 97 40 91 47  83 89 e1 1d 47 3e a9 a1  |..U..@.G....G>..
00000420  00 08 00 00 00 00 00 00  ff 07 10 00 00 00 00 00  |................
00000430  00 00 00 00 00 00 00 00  45 00 46 00 49 00 20 00  |........E.F.I. .
00000440  53 00 79 00 73 00 74 00  65 00 6d 00 20 00 50 00  |S.y.s.t.e.m. .P.
00000450  61 00 72 00 74 00 69 00  74 00 69 00 6f 00 6e 00  |a.r.t.i.t.i.o.n.
00000460  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................
*
00000480  af 3d c6 0f 83 84 72 47  8e 79 3d 69 d8 47 7d e4  |.=....rG.y=i.G}.
00000490  36 70 b2 65 8a b9 56 43  8b 25 c8 51 45 dd d5 da  |6p.e..VC.%.QE...
000004a0  00 08 10 00 00 00 00 00  ff 57 73 72 00 00 00 00  |.........Wsr....
000004b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................
*
00000500  6d fd 57 06 ab a4 c4 43  84 e5 09 33 c8 4b 4f 4f  |m.W....C...3.KOO|
00000510  6c 20 64 b2 38 68 96 45  b5 e9 1b a8 03 99 dc 72  |l d.8h.E.......r|
00000520  00 58 73 72 00 00 00 00  ff 67 70 74 00 00 00 00  |.Xsr.....gpt....|
00000530  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00000600
```

Figure 6: Partition entries

Each entry contains useful information for partitions, e.g. partition type GUID, unique partition GUID, first LBA, last LBA, attributes flags, and partition name.

i. **Entry 1:**

The Entry 1 (Figure 7) is started with 1024 byte (0x0400) and it consists of:
Partition type GUID: EFI System partition (C12A7328-F81F-11D2-BA4B-00A0C93EC93B)
Unique partition GUID: C355E600-4097-4791-8389-E11D473EA9A1
First LBA: 2 048 sector (0x0008000000000000)
Last LBA: 1 050 623 sector (0xff07100000000000)
Attribute flags: 0 (0x0000000000000000)
Partition name: EFI System Partition



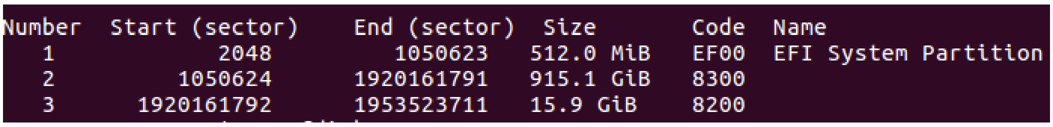
```
00000220  af 6d 70 74 00 00 00 00  22 00 00 00 00 00 00 00  |.mpt....".......|
00000230  8e 6d 70 74 00 00 00 00  1a b1 28 f7 b5 54 83     O|
00000240  a0 c1 b5 e2 fe a9 c6 a2  02 00 00 00 00 00 00 00  |................|
00000250  80 00 00 00 80 00 00 00  81 13 bc bd 00 00 00 00  |................|
00000260  00 00 00 00 00 00 00                 00 00 00 00  |................|
*
00000400  28 73 2a c1 1f f8 d2 11  ba 4b 00 a0 c9 3e c9 3b  |(s*......K...>.;|
00000410  00 e6 55 c3 97 40 91 47  83 89 e1 1d 47 3e a9 a1  |..U..@.G....G>..|
00000420  00 08 00 00 00 00 00 00  ff 07 10 00 00 00 00 00  |................|
00000430  00 00 00 00 00 00 00 00  45 00 46 00 49 00 20 00  |........E.F.I. .|
00000440  53 00 79 00 73 00 74 00  65 00 6d 00 20 00 50 00  |S.y.s.t.e.m. .P.|
00000450  61 00 72 00 74 00 69 00  74 00 69 00 6f 00 6e 00  |a.r.t.i.t.i.o.n.|
00000460  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
             3d c6 0f 83 84 72 47             d 69 d8 47 7d e4                }.|
00000490  36 70 b2 65 8a b9 56 43  8b 25 c8 51 45 dd d5 da  |6p.e..VC.%.QE...|
000004a0  00 08 10 00 00 00 00 00  ff 57 73 72 00 00 00 00  |.........Wsr....|
```

Figure 7: Entry 1

ii. **Entry 2:**
The Entry 2 (Figure 6) is started with 1152 byte (0x0480) and it contains:
Partition type GUID: Linux file system data (0FC63DAF-8483-4772-8E79-3D69D8477DE4)
Unique partition GUID: 65B27036-B98A-4356-8B25-C85145DDD5DA
First LBA: 1 050 624 sector (0x0008100000000000)
Last LBA: 1 920 161 791 sector (0xff57737200000000)
Attribute flags: 0
Partition name:

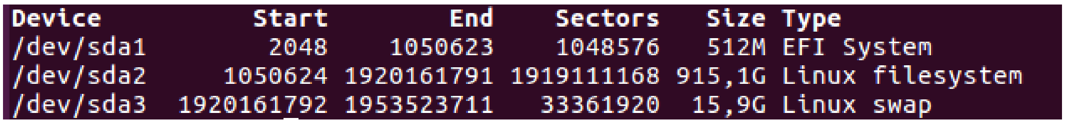
iii. **Entry 3:**
The Entry 3 (Figure 6) is started with 1280 byte (0x0500) and it contains:
Partition type GUID: Swap partition (0657FD6D - A4AB - 43C4 - 84E5 - 0933C84B4F4F)
Unique partition GUID: B264206C-6838-4596-B5E9-1BA80399DC72
First LBA: 1 920 161 792 (0x0058737200000000)
Last LBA: 1 953 523 711 (0xff67707400000000)
Attribute flags: 0
Partition name:

This information can be verified using the **gdisk** utility (Figure 8) and the **fdisk** utility (Figure 9).



```
Number  Start (sector)    End (sector)  Size       Code  Name
   1            2048         1050623   512.0 MiB   EF00  EFI System Partition
   2         1050624      1920161791   915.1 GiB   8300
   3      1920161792      1953523711   15.9 GiB    8200
```

Figure 8: Using the **gdisk** utility to verify values from the Partition Entry Array



```
Device           Start        End    Sectors   Size Type
/dev/sda1         2048    1050623    1048576   512M EFI System
/dev/sda2      1050624 1920161791 1919111168 915,1G Linux filesystem
/dev/sda3   1920161792 1953523711   33361920  15,9G Linux swap
```

Figure 9: Using the **fdisk** utility to verify values from the Partition Entry Array

4. At what byte index from the start of the disk do the real partition table entries start?

   Answer: 1024 (0x0400). Because it is LBA2, each LBA of 512 bytes.

5. What byte index would the partition table start if your server had a so-called "4K native" (4Kn) disk?

   Answer: The partition table would start from a 0 byte, and the partition table entries would start from an 8 192 byte.

6. If you wanted to add a 47 612 GiB FreeBSD ZFS partition, called ØŚ3 (U+00D8 U+015A U+0033) to the table by hand, what values would you have to use for the entry (including the name) in the raw table on disk? Assume the disk is large enough to hold the extra partition.

   Answer:

   I would add the following values:

   Partition type GUID: 0xba7c6e51cf6ed611 0x8ff800022d09712b (FreeBSD ZFS partition's type GUID is 516E7CBA-6ECF-11D6-8FF8-00022D09712B)

   Unique partition GUID: 0x41cffa523ea7cf4c 0xafca434fe4511290 (52FACF41-A73E-4CCF-ACAF-434FE4511290)

   First LBA: 0x0068707400000000 (1 953 523 712)

   Last LBA: 0x67f0b317000000ff (101 803 124 735)

   Attribute flags: 0

   Partition name: 0xd8005a0133000000 (ØŚ3)

7. Until recently, machines would have a BIOS and boot from an MBR. Name two differences between primary and logical partitions in an MBR partitioning scheme.

   Answer:

   The first difference is that primary partitions can be only four, and logical partitions can be any number limited by the size of the disk. The second difference is that the primary partition always contains only one file system, and the logical partition does not contain its own file system.

3 Conclusion

This laboratory work allows understanding what is UEFI PXE booting, GPT, gdisk utility. It was also useful to find out what GPT consists of and what each field means. The GPT field is called LBA that means logical block addressing and is usually 512 bytes. The first LBA (LBA0) is the Protective MBR that serves to protect the GPT from a legacy MBR-based utility. The second LBA (LBA1) is the Primary GPT Header that contains information about disk usable blocks. The following LBA consist of the Partition Entry Array that contains entries about the partitions on the disk. The Partition Entry Array is followed by partitions. And the last LBA is the Secondary GPT Header which is used to backup the Primary GPT Header.

4 References

[1] Wikipedia: Unified Extensible Firmware Interface [https://en.wikipedia.org/wiki/Unified_Extensible_Firmware_Interface]

[2] Wikipedia: Preboot Execution Environment [https://en.wikipedia.org/wiki/Preboot_Execution_Environment]

[3] Wikipedia: GUID Partition Table [https://en.wikipedia.org/wiki/GUID_Partition_Table]

[4] What's the Difference Between GPT Disk and MBR Disk? [http://www.disk-partition.com/gpt-mbr/difference-between-mbr-and-gpt-1203.html]

[5] Basic and Dynamic Disks [https://msdn.microsoft.com/ru-ru/library/windows/desktop/aa363785(v=vs.85).aspx#guid_partition_table]

[6] Manual page

[7] Wikipedia: Disk partitioning https://en.wikipedia.org/wiki/Disk_partitioning

[8] Online Hex Converter http://www.scadacore.com/field-tools/programming-calculators/online-hex-converter/

[9] Online GUID Converter http://robobunny.com/cgi-bin/guid

[10] Wikipedia: Online GUID Converter https://en.wikipedia.org/wiki/Master_boot_record