Dikerjakan oleh: Alvaro Austin (2106752180)

Exercise Section

1. Section 2.4

```c
#include <stdio.h>
struct Student {
    int id;
    char province[3]; //ON, BC etc. + terminating null
    int age;
};
int main( ){
    struct Student student1;
    student1.id = 100364168;
    strncpy(student1.province, "ON\0", sizeof(student1.province));
    student1.age = 18;
    printf("The size of struct member id is %d bytes\n", sizeof(student1.id
        ));
    printf("The size of struct member province is %d bytes\n", sizeof
        (student1.
    province));
    printf("The size of struct member age is %d bytes\n", sizeof(student1
        .age));
    printf("The size of struct Student is %d bytes\n", sizeof(struct
        Student));
    return 0;
}
```

Melalui kode tersebut kita dapat memperoleh size dari variable/data type diatas.

Variable or Data Type		Sizes in bytes
student1.id	100	4
student1.province	101	3
student1.age	102	4
Add Lines 100, 101, 102	103	11
Struct student	104	12

Q1. Which is bigger? Ukuran dari struct Student lebih besar daripada penjumlahan bytes dari 3 attribute yang didalam struct. Hal ini karena kebutuhan untuk maintain kelipatan 4 dari int yang ada.

Part B

Table 2.8 Data presentation of variables in the above C code

Variables	Representation of values (in hexadecimal) (after a value is assigned to the variable)
student1.id	0x05FB6F88
student1.province	0x00004E4F
student1.age	0x00000012
Student1	0x00E32E48

Part C

Table 2.9 Memory adresses and stored values of char array elements in the above C code

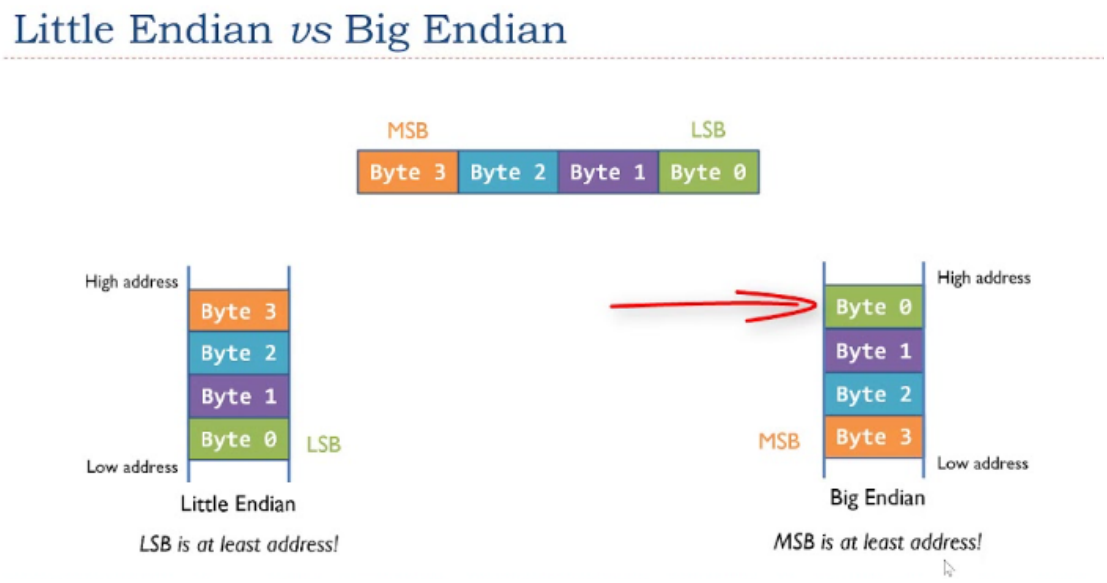
Item of array digits	First item	Second item	Third item	Fourth item
Memory address	0x61FF18	0x61FF19	0x61FF1A	0x61FF1B
Stored value	0x12	0x34	0x56	0x78

Q2: What is the value in hexadecimal format at the memory address specified by the integer pointer ptr?

Nilai dari integer pointer ptr pada memory address (0x61FF1C) dalam hexadecimal format is **0x78563412**.

Q3: According to your debug output, which endianness (big or little endian) is used in your system? Briefly explain your rationale for your conclusion. If necessary, give a diagram to help in your explanation.

Dari output yang kita peroleh diatas, kita dapat menyimpulkan bahwa sistem saya menggunakan Little Endian sebagai byte ordernya. Hal ini dapat dibuktikan karena little endian LSB (Least Significant Byte) disimpan pada memory terendah. Oleh karena itu, karena kita mengetahui nilai dari pointer pada memory address 0x61FF1C adalah **0x78563412**. Kita dapat lihat bahwa untuk memory terendah (0x61FF18), LSB nya juga merepresentasikan nilai dari memory tersebut yaitu 0x12. Oleh karena itu sistem yang dipakai adalah Little Endian.

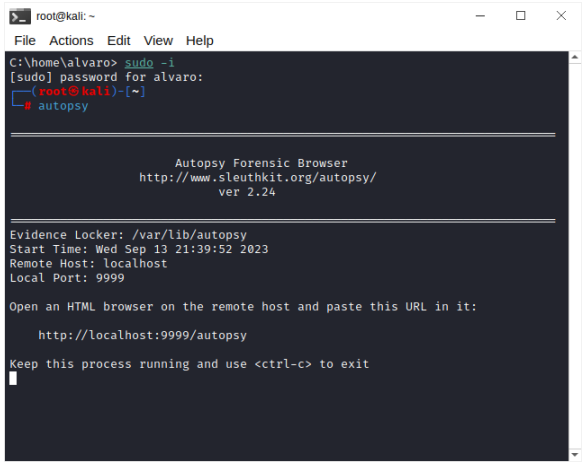


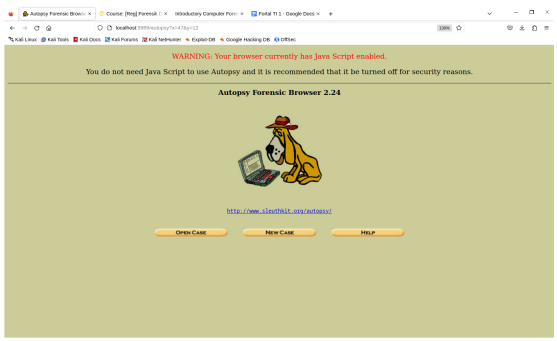
Dapat dilihat bahwa pada value pada low address suatu sistem menjadi LSB, sedangkan value pada high address menjadi MSB.

2. Section 3.5

Part A: Starting Your Autopsy Forensic Browser

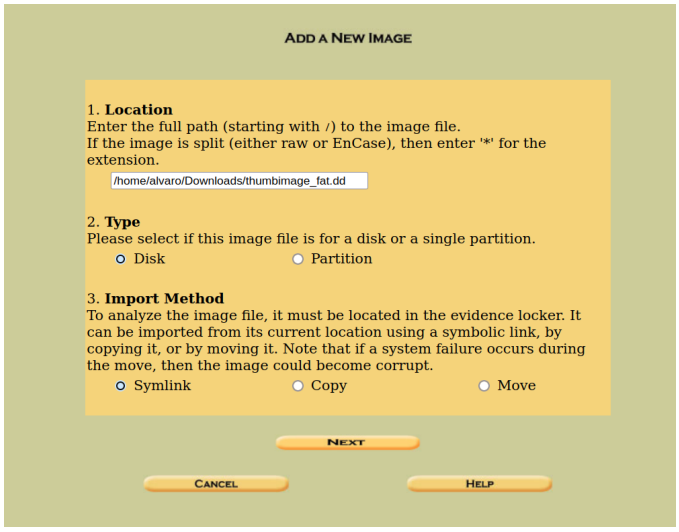
Pada kesempatan kali ini, saya berhasil membuka Kali Linux dan menyalakan server Autopsy Forensic Browser yang dapat diakses pada http://localhost:9999/autopsy. Saya menyalakan server ini dengan login ke root menggunakan command (**sudo -i**) lalu memasukkan kredensial, lalu mengetik command autopsy.





Part B: Starting a New Case in Autopsy

Selanjutnya saya menambahkan kasus baru pada halaman autopsy saya. Kasus saya memiliki **case** bernama **UCS.CompromisedWebServer** (sesuai dengan di buku) dan host bernama HackerPC (berbeda dengan yang dibuku). Setelah saya mengisi case name (UCS.CompromisedWebServer) dan diri saya sebagai investigator, saya selanjutnya menambahkan host bernama HackerPC. Selanjutnya saya menambahkan image dengan mengisi lokasi dari file image tersebut.

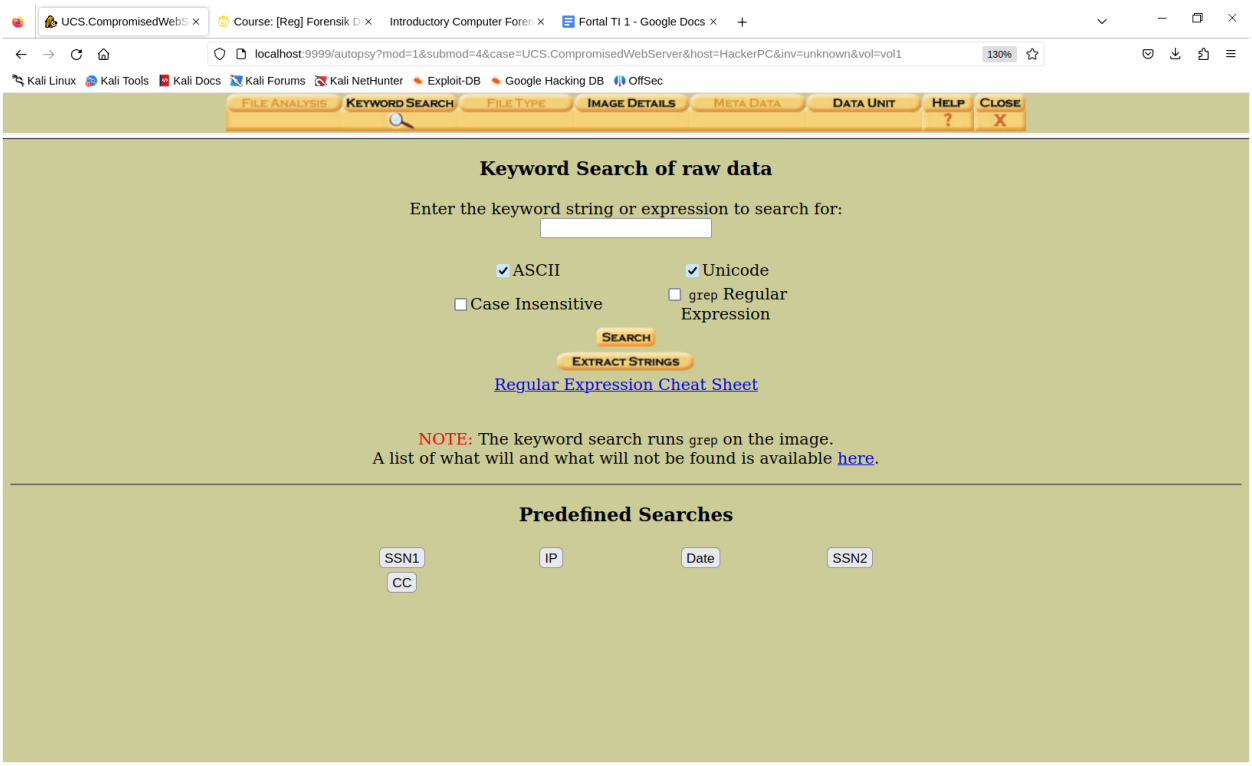


Command yang saya tulis adalah: “/home/alvaro/Downloads/thumbimage_fat.dd”. Selanjutnya saya tekan tombol add image dan mencari hash valuenya menggunakan opsi yang diberikan.

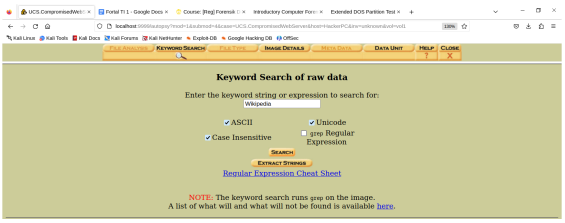
Q1. What is the MD5 hash value of the disk image “thumbimage_fat.dd” calculated in Autopsy? **55BBF61C99E8E91D97E57696859C6FC7**

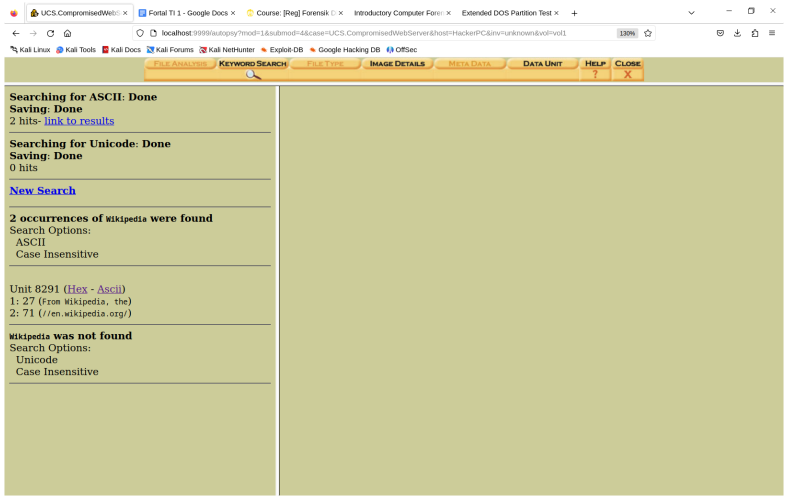
Part C: Using Autopsy for Forensic Disk Analysis

Setelah itu, kita dapat melihat bahwa terdapat tampilan volume yang terlihat. Kita dapat menekan tick “disk” dan tekan tombol **Analyze**.



Menurut perintah, kita diminta untuk mencari keyword wikipedia untuk mendapatkan *list of hits*. Kita juga perlu melakukan tick pada case insensitive seperti yang dihimbau.





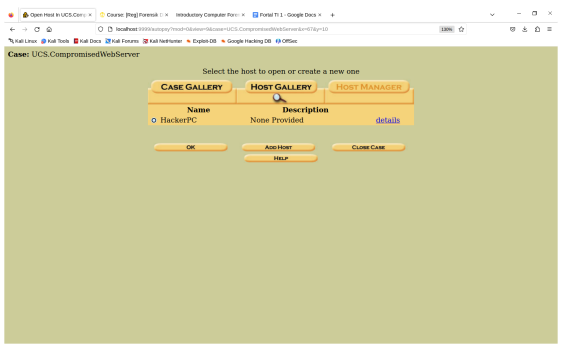
Oleh karena itu, saya akan menjawab pertanyaannya:

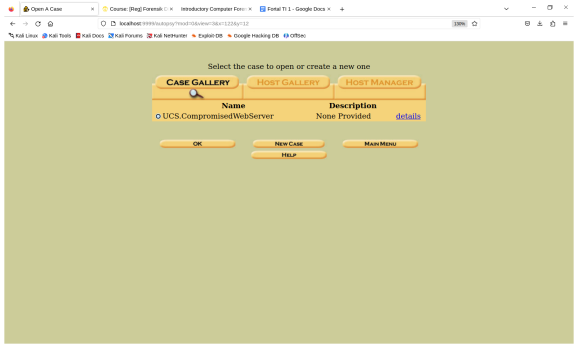
Q2. How many hits when the search keyword is encoded to ASCII format? **2 hit.**

Q3. What is the number (or address) of the data unit where the keyword resides? **8291**

Q4. How many hits when the search keyword is encoded as Unicode? **0 hit.**

Terakhir, saya akan mematikan host dan case saya.

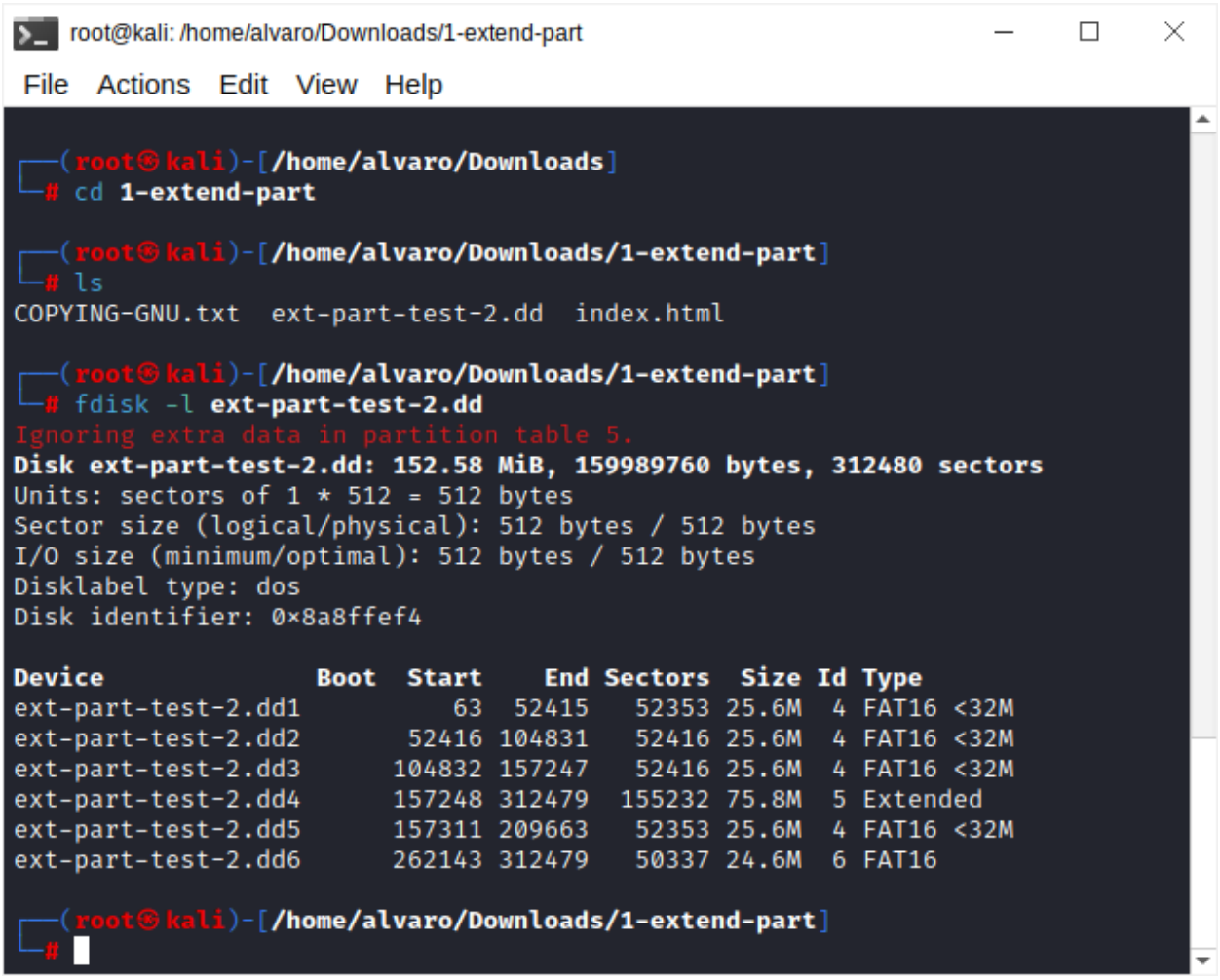




3. Section 4.3

Pada bagian ini saya akan menggunakan autopsi dari Kali Linux yang sudah saya miliki. Pertama-tama saya akan membuka terminal dimana saya akan menggunakan command fdisk untuk melihat partisi tersebut. Pertama-tama saya login pada root lalu pergi ke directory **/home/alvaro/Downloads/1-extend-part** dimana file **ext-part-test-2.dd** berada. Selanjutnya saya menggunakan command **fdisk -l ext-part-test-2.dd** untuk mengeluarkan table dengan detail semua partisi yang ada pada disk image.

Dikerjakan oleh: Alvaro Austin (2106752180)



PART A: Analyze the MBR of the Disk Image "ext-part-test-2.dd", and Fill Out the Following Table with Details of the First Partition

Maka dapat dilihat bahwa partisi pertama merupakan device "**ext-part-test-2.dd1**".

First Partition	
Start LBA Address	63
Number of Sector in Partition	52353
Size Of Partition (MB)	25.6
Type of partition	FAT16 <32M

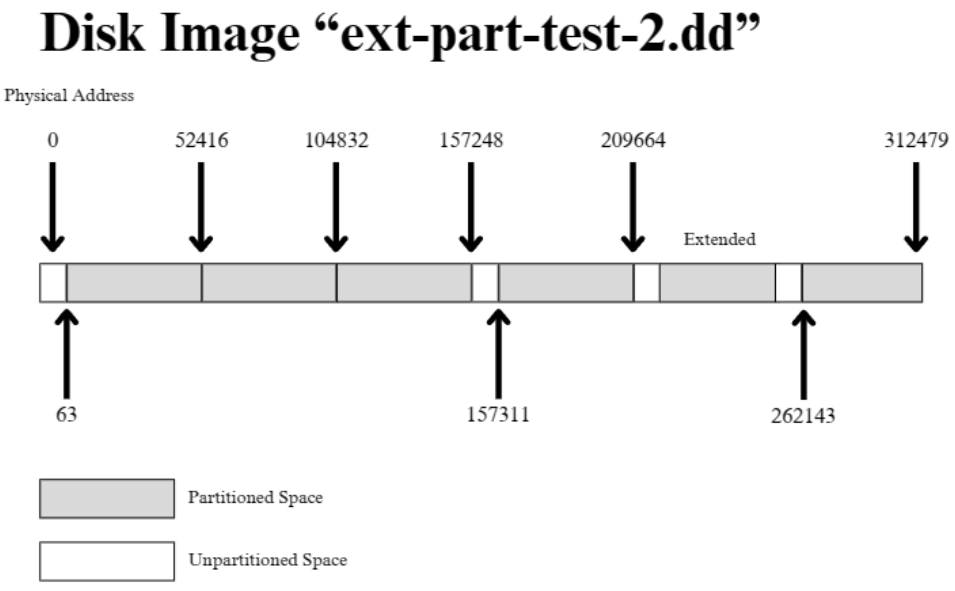
PART B: Analyze the First Extended Partition, and Fill Out the Following Table with Details of the Partition

Untuk bagian B, berdasarkan bukti diatas kita dapat melihat Type Extended yaitu **ext-part-test-2.dd4**. Partisi Extended berarti partisi ini menyimpan beberapa partisi logikal.

First Partition	
Start LBA Address	157248
Number of Sector in Partition	155232
Size Of Partition (MB)	75.8

PART C: Find Out the Layout of the Disk Image "ext-part-test-2.dd"

Berikut adalah *layout* dari Disk Image "ext-part-test-2.dd" yang saya temukan:
ASUMSI: Partitioned Space di **scale down** dan **Unpartitioned Space** di **scale up** agar dapat memperlihatkan layoutnya dengan gambar.



Q1. By looking at the disk layout which you have figured out, is there any unpartitioned space? **Yes**. (Unpartitioned space can be used to hide data.)

Terdapat unpartitioned space terlihat pada data yang belum dialokasikan seperti pada layout yang diatas. Dapat dilihat bahwa pada layout tersebut terdapat 4 space yang tidak dipartisi. Space **yang tidak dipartisi** dilambangkan dengan warna **putih**. Namun unpartitioned space hanyalah berukuran 63 bytes, sehingga tampilannya sangatlah kecil dan perlu dilakukan **scale up**.

PART D: Use dcfldd to Extract the First Partition Image from the Disk Image Provided

Q2. Use the spaces provided to write down the command(s) you issued.

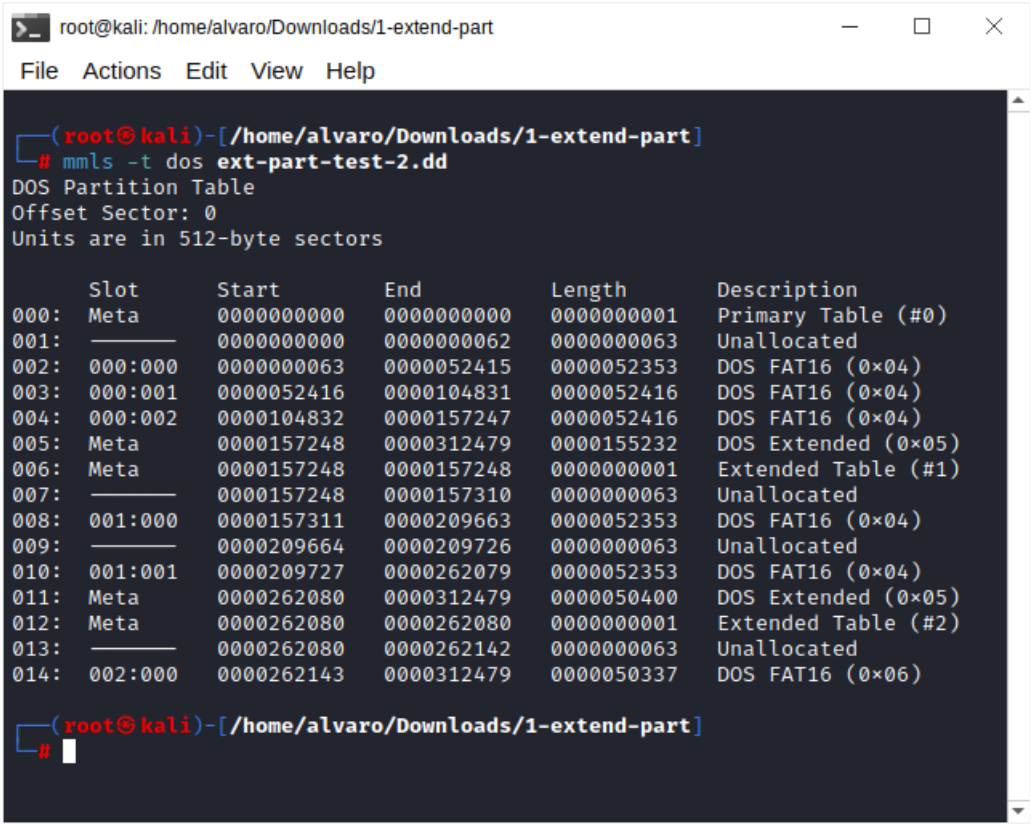
```
dcfldd if=ext-part-test-2.dd of=first_partition.dd bs=512
count=52353 skip=63
```

Command diatas bermaksud bahwa input file merupakan "**ext-part-test-2.dd**" dan di extract pada file "first_partition.dd" dengan melakukan skip 63 sectors (karena first partition mulai pada sektor 63) dan bs merupakan Block Size setiap sector.

Part E: Validate Answers

Untuk memastikan hasil, kita dapat menggunakan command:

```
mmls -t dos ext-part-test-2.dd
```



Kita dapat melihat bahwa layout yang kita buat sesuai dengan data yang didapatkan menggunakan command diatas.

Tugas Menonton Video

1. Pendahuluan

1.1. Apa itu forensik digital?

Forensik digital adalah keseluruhan proses dalam mengidentifikasi, melestarikan, menganalisis, dan menyajikan bukti digital dalam rangka mendukung investigasi kegiatan forensik. Sehingga secara luas, forensik digital adalah kegiatan dalam mengidentifikasi, mengumpulkan, menganalisis, dan menyajikan data elektronik yang berhubungan pada proses investigasi (hukum atau kegiatan forensik lainnya).

1.2. Apa saja bidang/area yang perlu dianalisa pada forensik digital?

Bidang/area yang perlu dianalisa pada forensik digital (menurut video) adalah sebagai berikut:

- Storage Media: Menganalisis isi dari media penyimpanan digital seperti hard drive, USB drive, atau kartu memori untuk mencari bukti relevan
- Hardware dan Sistem Operasi (OS): Memeriksa perangkat untuk melacak retasan yang bisa terjadi.
- Jaringan (Networks): Menyelidiki jejak digital dalam jaringan komputer untuk melacak serangan atau aktivitas ilegal. Ini juga melibatkan pemantauan dan analisis lalu lintas jaringan
- Aplikasi (Application): Menyelidiki aplikasi yang digunakan dalam penyelidikan, termasuk memeriksa jejak penggunaan atau peretasan.

Pada saat saya mencari informasi mengenai hal ini, tidak hanya 4 bidang diatas seperti yang disebutkan dalam video, ternyata bidang/area yang perlu dianalisa pada forensik digital masih lebih banyak lagi. Area-area yang tidak disebutkan seperti kriptografi (analisis enkripsi data), IoT (mengutilisasi perangkat seperti kamera untuk mencari lebih banyak bukti), dsb.

1.3. Apa saja proses forensik digital?

1. Pengumpulan Data (Data Collection)
    a. Mendapatkan otoritas pencarian(Obtain Search Authority): pada saat melakukan suatu forensik digital, kita harus memiliki otoritas legal bahwa kita merupakan pihak yang wajar untuk melakukan forensik ini.
    b. Dokumentasi barang bukti(Document Chain of Custody): agar bukti tidak hilang.
    c. Menduplikasi dan menyimpan bukti(Hash and Duplicate All Evidence): agar dapat dipresentasikan.
2. Analisis dan Pemeriksaan (Analysis and Examination)
    a. Validasi perangkat yang digunakan: mencari bukti.
    b. Melakukan analisis
    c. Mengulangi proses sebagai asuransi: mereplikasikan suatu kegiatan dapat memberikan pandangan yang lebih jelas.
3. Melakukan Laporan (Reporting)

a. Membuat konklusi: memberikan konklusi yang ditemukan berdasarkan bukti.
b. Menyampaikan bukti dan testimoni yang kuat: untuk mendukung konklusi diatas.

1.4. Berikan contoh penggunaan ilmu/teknik forensik digital diluar aktivitas kriminal?

Kita dapat menggunakan ilmu/teknik forensik digital diluar aktivitas kriminal seperti:

- Audit keamanan yang menggunakan teknik *intrusion investigation* dianggap sebagai teknik yang digunakan untuk melakukan perbaikan atas pelanggaran keamanan yang dilakukan. Hal ini akan mengevaluasi sistem keamanan dan memperbaikinya.
- Pengujian keamanan aplikasi: melalui ilmu dan teknik forensik, kita bisa menemukan sisi lemah dari keamanan suatu aplikasi. Hal ini dapat mencegah aktivitas penyerang aplikasi. Melalui ilmu forensik, kita dapat memahami serangan yang terjadi dan mencari respon yang tepat terhadap serangan tersebut.

2. Volume System (File system forensic)

2.1. Apa yang dilakukan BIOS sebelum melihat MBR pada Sector pertama di suatu disk?

- Power-On Self-Test (POST): BIOS melakukan POST, yang merupakan serangkaian tes untuk memastikan bahwa komponen-komponen hardware dalam sistem berfungsi dengan baik.
- Identifikasi Perangkat Boot: BIOS mencari perangkat yang dapat di-boot dan berisi MBR seperti hard drive, kartu grafis, dll.
- Pembacaan MBR: Setelah perangkat boot diidentifikasi, BIOS membaca MBR dari perangkat penyimpanan ke dalam memori fisik. MBR berisi kode boot yang digunakan oleh BIOS untuk mengidentifikasi di mana partisi sistem berada.
- Eksekusi Boot Loader: Setelah menemukan MBR, BIOS akan menjalankan program boot loader yang terdapat di MBR tersebut. Boot

loader ini akan mengambil alih kendali dari BIOS dan memulai proses booting selanjutnya, seperti memuat sistem operasi yang terinstal pada hard drive.

2.2. Berdasarkan video & buku Introductory Computer Forensics, apa perbedaan mendasar dari boot loading sistem operasi Linux dan Windows?

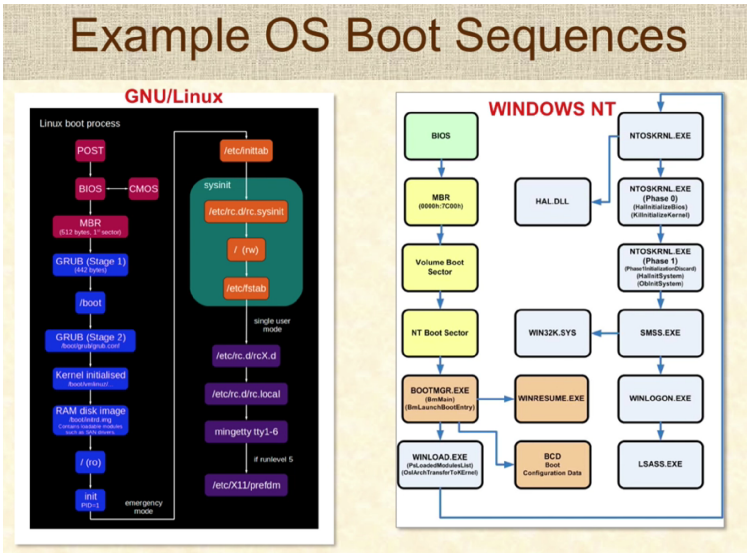
Berdasarkan video dan buku, perbedaan mendasar dari boot loading sistem operasi linux dan windows berada pada **Boot Loader** yang digunakkan. Untuk sistem operasi Linux biasanya menggunakan GRUB (Grand Unified Bootloader) yang memberikan fleksibilitas bagi pengguna untuk memilih operating system yang mereka inginkan sehingga memberikan setup yang mudah untuk *dual-boot configurations*. Di lain sisi, windows menggunakan Windows Boot Manager (BOOTMGR) yang dispesifikasikan hanya untuk windows. Oleh karena itu pada Linux, GRUB memiliki fleksibilitas yang tinggi karena bisa booting OS untuk Linux, Windows, dan MAC OS. Akan tetapi, BOOTMGR tidak menyediakan fleksibiltas ini.

Terdapat juga **Boot Process** yang membedakan langkah-langkah pada kedua OS tersebut seperti pada diagram dibawah.

- Linux mencakupi beberapa proses sebagai berikut:
    - BIOS Firmware (POST)
    - Boot Loader (GRUB) -> Stage 1 & 2
    - Kernel
    - Initial RAM disk
    - Init system (sysinit)
- Windows mencakupi beberapa proses sebagai berikut:
    - BIOS Firmware (MBR, VBS, dll)
    - Windows Boot Manager (BOOTMGR) -> main bootloader
    - Kernel

- Windows Session Manager

Sehingga kesimpulannya perbedaan boot loading system berada pada **Boot Loader** yang menyebabkan perbedaan **boot process**.



2.3. Berikan contoh boot loader yang digunakan pada Linux atau Windows!

Contoh bootloader yang digunakan pada linux:

- GRUB (Grand Unified Bootloader): bootloader utama yang digunakan OS Linux yang dapat digunakan untuk boot OS Linux, Windows, dan juga MAC OS. GRUB merupakan salah satu bootloader paling populer karena menyediakan support untuk UEFI dan BIOS system.
- LILO (Linux Loader): Bootloader yang pernah digunakan Linux sehingga dikatakan sebagai *legacy* bootloader.

Contoh bootloader yang digunakan pada windows:

- Windows Boot Manager (BOOTMGR): Bootloader yang dispesifikasikan untuk OS Windows.
- NTLDR (NT Loader): Bootloader yang dulu sempat digunakan oleh OS Windows seperti Windows XP.