

## Programa particiones

Dispositivo /dev/sdc formateado:

```
root@rafael-vallejo -> /h/sansforensics
# dd if=/dev/zero of=/dev/sdc
dd: writing to '/dev/sdc': No space left on device
2097153+0 records in
2097152+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 20.7262 s, 51.8 MB/s
root@rafael-vallejo -> /h/sansforensics
# fdisk -l /dev/sdc
Disk /dev/sdc: 1 GiB, 1073741824 bytes, 2097152 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@rafael-vallejo -> /h/sansforensics
#
```

Ejecución del script desarrollado:

```
root@rafael-vallejo -> /h/sansforensics
# python particionar.py /dev/sdc
p primary
e extended
q exit
Select (default p): p
Sistemas de archivos disponibles:
82 Linux swap
83 Linux
a5 FreeBSD
07 HPFS/NTFS/exFAT
a6 OpenBSD
Selecciona el sistema de archivos [Valor|Nombre]: 82
Partition number (1-4): 1
size{K,M,G}: 200M
NOTA: Se deben guardar los cambios por cada partición que se cree.
Guardar [w], salir [q], regresar [ENTER]: w
Se creen en la particion 1el tipo de particion Linux swap
p primary
e extended
q exit
Select (default p): p
Sistemas de archivos disponibles:
82 Linux swap
83 Linux
a5 FreeBSD
07 HPFS/NTFS/exFAT
a6 OpenBSD
Selecciona el sistema de archivos [Valor|Nombre]: 07
Partition number (1-4): 2
size{K,M,G}: 400M
```

```
Se creo en la particion 2el tipo de particion HPFS/NTFS/exFAT
p primary
e extended
q exit
Select (default p): p
Sistemas de archivos disponibles:
82 Linux swap
83 Linux
a5 FreeBSD
07 HPFS/NTFS/exFAT
a6 OpenBSD
Selecciona el sistema de archivos [Valor|Nombre]: a5
Partition number (1-4): 3
size{K,M,G}: 300M
NOTA: Se deben guardar los cambios por cada partición que se cree.
Guardar [w], salir [q], regresar [ENTER]: w
Se creo en la particion 3el tipo de particion FreeBSD

p primary
e extended
q exit
Select (default p): p
Sistemas de archivos disponibles:
82 Linux swap
83 Linux
a5 FreeBSD
07 HPFS/NTFS/exFAT
a6 OpenBSD
Selecciona el sistema de archivos [Valor|Nombre]: a6
Partition number (1-4): 4
size{K,M,G}: 100M
NOTA: Se deben guardar los cambios por cada partición que se cree.
Guardar [w], salir [q], regresar [ENTER]: w
Se creo en la particion 4el tipo de particion OpenBSD
p primary
e extended
q exit
Select (default p): q
root@rafael-vallejo -> /h/sansforensics
# Memory-Forensics-
```

Mediante fdisk se comprueba que las particiones fueron creadas:



```
root@rafael-vallejo -> /h/sansforensics
# fdisk -l /dev/sdc
Disk /dev/sdc: 1 GiB, 1073741824 bytes, 2097152 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device      Boot    Start        End    Sectors    Size Id Type
/dev/sdc1                2048     411647     409600    200M 82 Linux swap / Solaris
/dev/sdc2           821248    1640447     819200    400M  7 HPFS/NTFS/exFAT
/dev/sdc3          1435648    2050047     614400    300M a5 FreeBSD
/dev/sdc4          1574912    1779711     204800    100M a6 OpenBSD

root@rafael-vallejo -> /h/sansforensics
#
```

Se renombró el script para coincidir con el formato que se pide en las prácticas:

```
root@rafael-vallejo -> /h/sansforensics
# md5sum particionar.py
a5c70b39966e990c6e43802343046386 particionar.py
root@rafael-vallejo -> /h/sansforensics
# mv particionar.py rvallejo_practica2.py
root@rafael-vallejo -> /h/sansforensics
# md5sum rvallejo_practica2.py
a5c70b39966e990c6e43802343046386 rvallejo_practica2.py
root@rafael-vallejo -> /h/sansforensics
#
```

## Conclusión

La realización de la práctica 1 fue indispensable para poder desarrollar este script ya que en ella se vio como está compuesto el MBR y entender que es lo que cada byte significa.

El lenguaje en el que se desarrolló el script (Python) permitió lograr su implementación debido a que es un lenguaje bastante amigable y poderoso.

Se logró realizar el programa para las particiones, no al nivel de fdisk, pero si para comprender como es que la estructura MBR permite hacer el manejo de las particiones, tanto con el código de arranque, la tabla de particiones (que fue lo más importante aquí) y el fin del MBR.

El script puede mejorarse para hacer un manejo correcto de los sectores de inicio y fin (los bytes correspondientes a cada partición). Sin embargo, de no haber comprendido cada una de las partes del MBR, no habría sido posible terminar el programa para realizar la creación de las particiones o hubiera costado mucho más trabajo lograr realizarlo.

## Referencias

Haider, M. (2012). Analysing the Master Boot Record (MBR) with a hex editor (Hex Workshop). Recuperado el 29de junio de 2019, de <http://blog.hakzone.info/posts-and-articles/bios/analysing-the-master-boot-record-mbr-with-a-hex-editor-hex-workshop/>

Convert bytes to MiB -Conversion of Measurement Units.Recuperado el 29de junio de 2019, de <https://www.convertunits.com/from/bytes/to/MiB>