universidade
de aveiro

Computer Systems Forensic Analysis
AFSC

Volumes and Partitions

Artur Varanda

School Year 2021-2022

Volumes

- allows joining several small volumes into a larger one
- or split the physical storage space into smaller spaces
- is a collection of sectors
  - ✓ for the OS those sectors are consecutive – volume level
  - ✓ but at the physical level they **may not be consecutive**

Partitions

- is a particular case of volumes
- a partition is a set of **consecutive** sectors
- the confusion between partitions and volumes is common

Volumes

- are structures that define the space occupied by the file system
- you need to know the volume structure to analyze its contents
  - ✓ if a drive is corrupted you may not be able to read the volume structure
  - ✓ a volume might have been deleted in an attempt to hide data

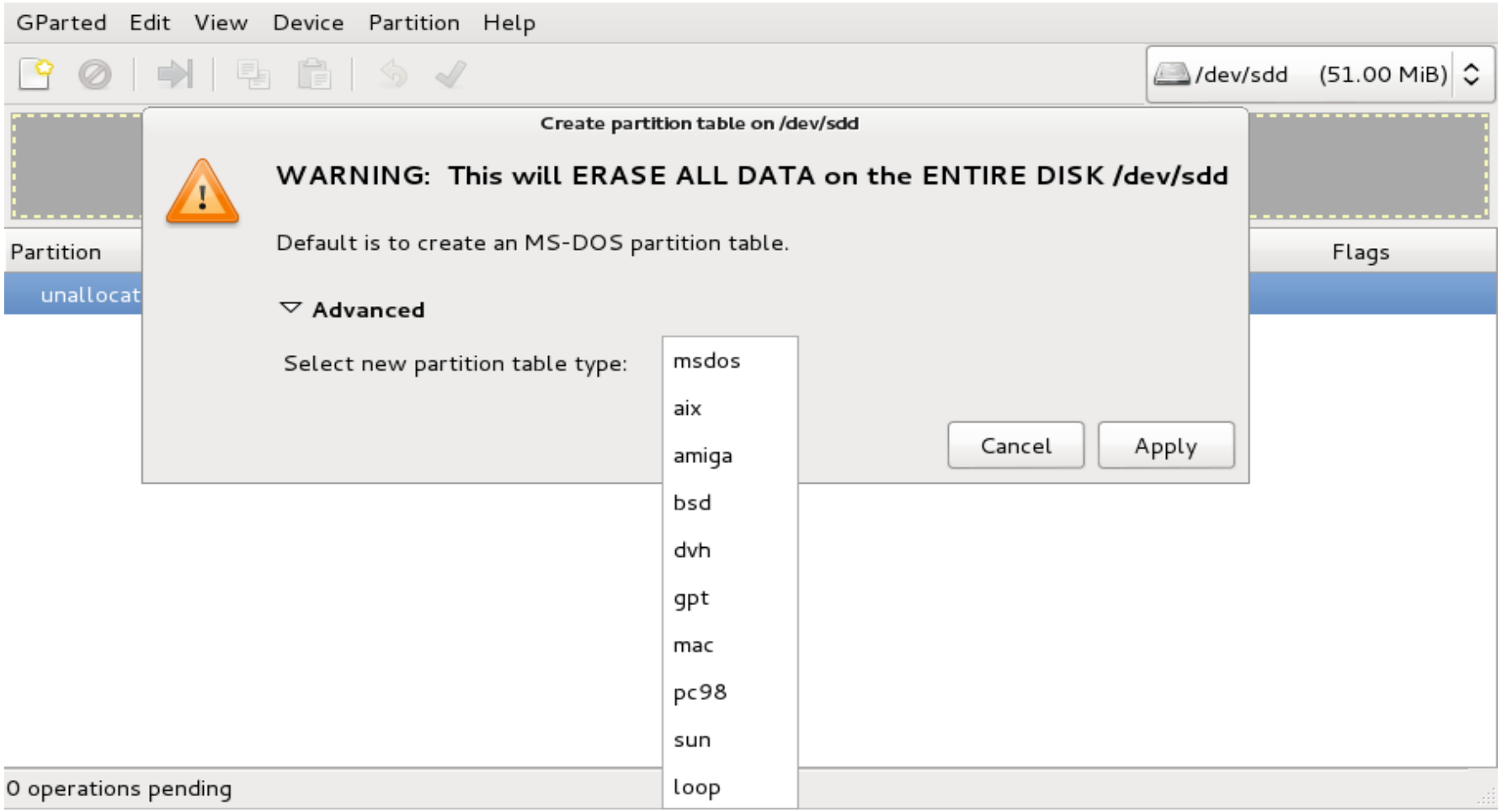
Why use volumes?

- some OS use a volume to store RAM data when they hibernate, *e. g.* linux
- to separate the OS files from the users' files
- to allow dual boot, *e. g.* windows and linux
- to aggregate volumes
  - ✓ to get more space for the file system
  - ✓ to get redundancy and prevent data loss due to drive failures

Partition Tables

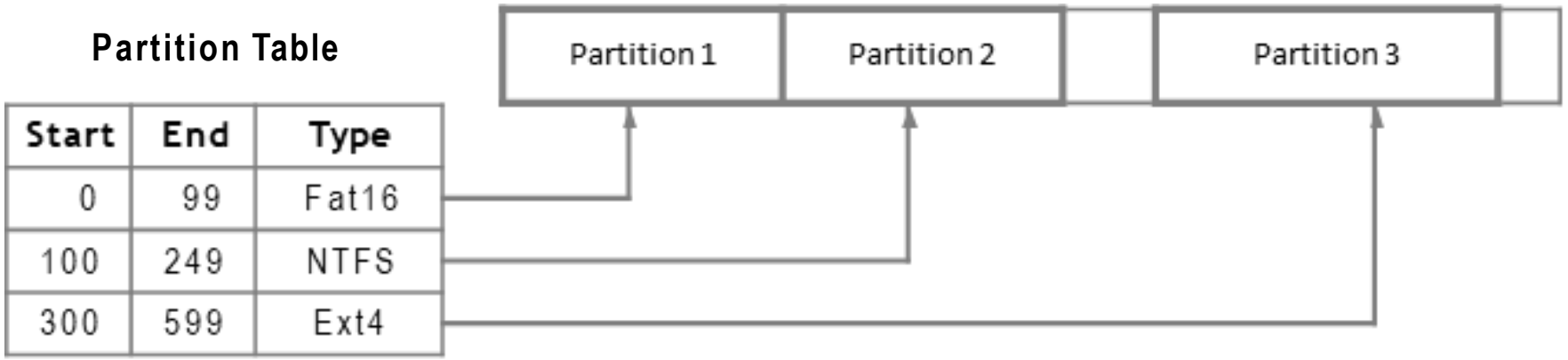
What is the partition table?

- is a data structure to store information about the partitions on a given drive
- there are several types of partition tables
- partition table depend on the used OS and not on the drives' physical interface

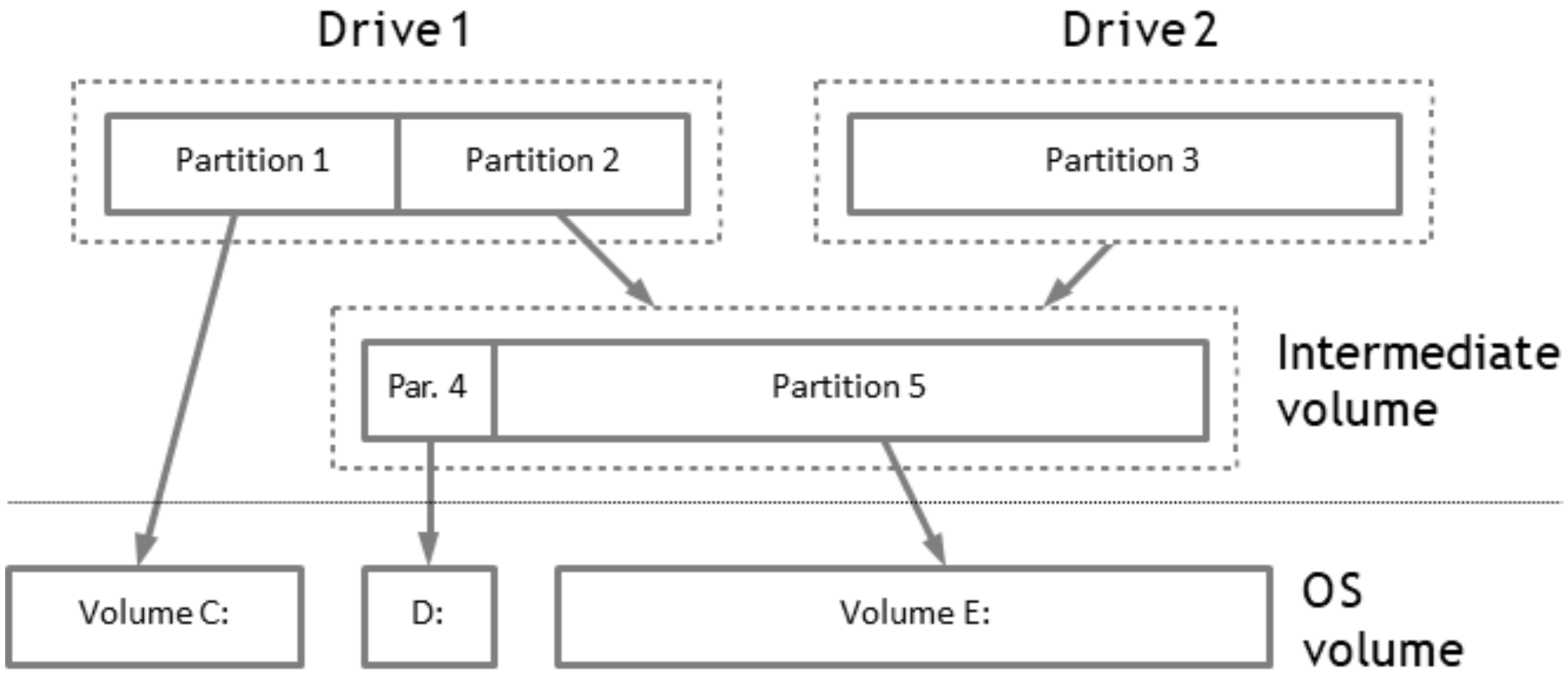


Partition Tables

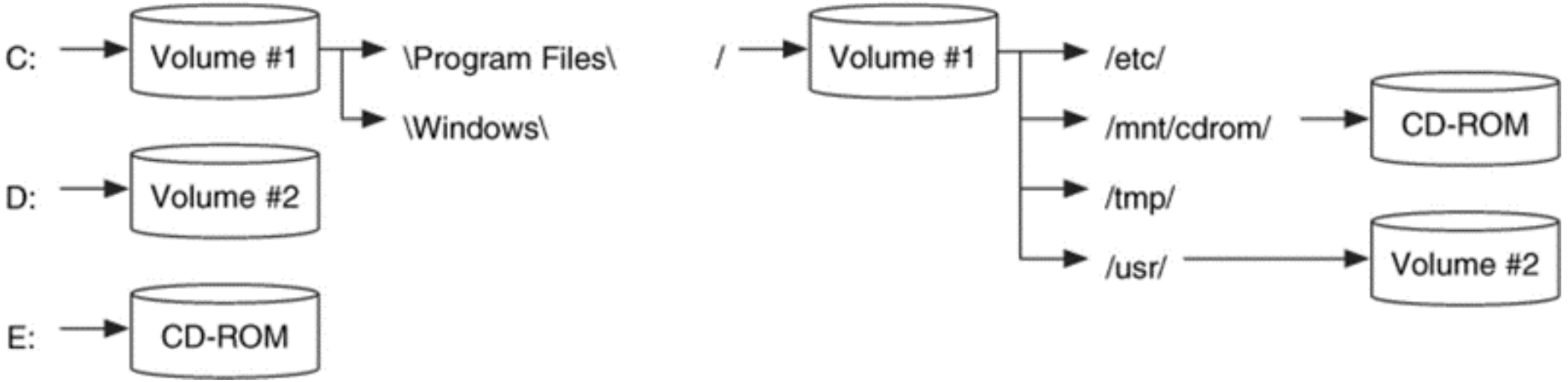
- **essential data:** begin and end sectors
- **non essential data:** type of partitions and description (can be fake)
- the start and end sectors are usually indistinguishable
  - ✓ if the partition table is corrupted it can be estimated based on prior knowledge
- there may be unallocated sectors between partitions



Partitions and volumes



Partitions and volumes – Windows vs Linux



Source: Brian Carrier, "*File System Forensic Analysis*", Addison-Wesley Professional, March 27, 2005

Sectors Addresses

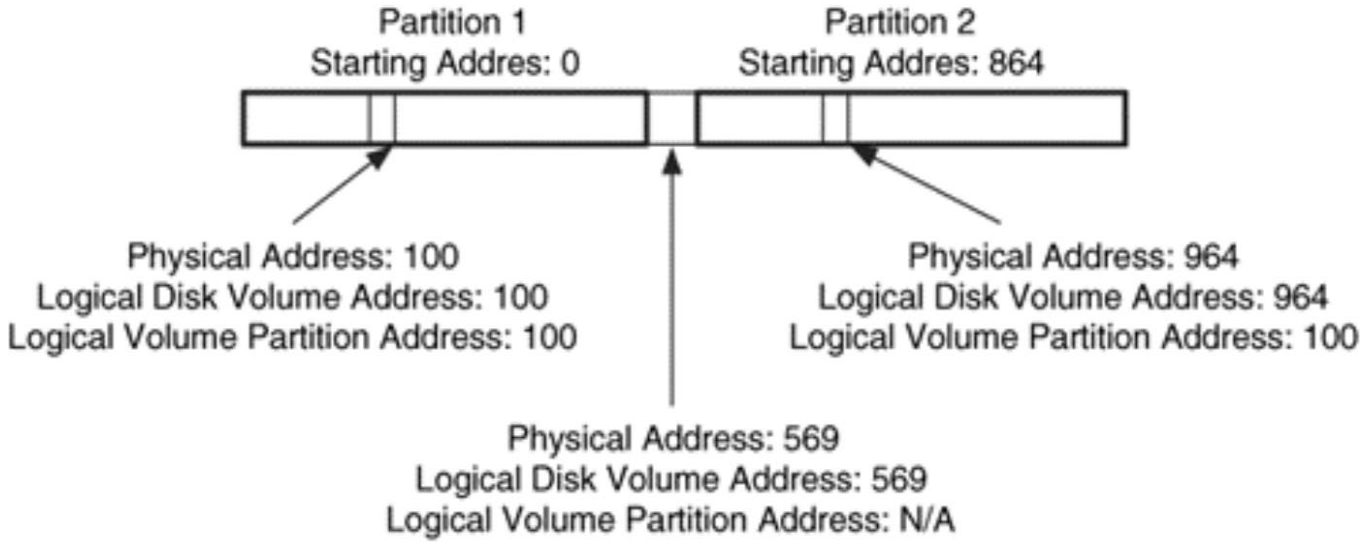
LBA addresses

- maps the physical sectors of the drive
- the first sector is always 0 (zero)
- it cannot be used to address volume sectors
  - ✓ a volume is a collections of sectors, but
  - ✓ may not be consecutive and can even be in different drives

Layers of addresses

- physical address
- logical disk volume addresses – equal to the physical address
- logical volume partition addresses – each partition has its own address space

Logical addresses



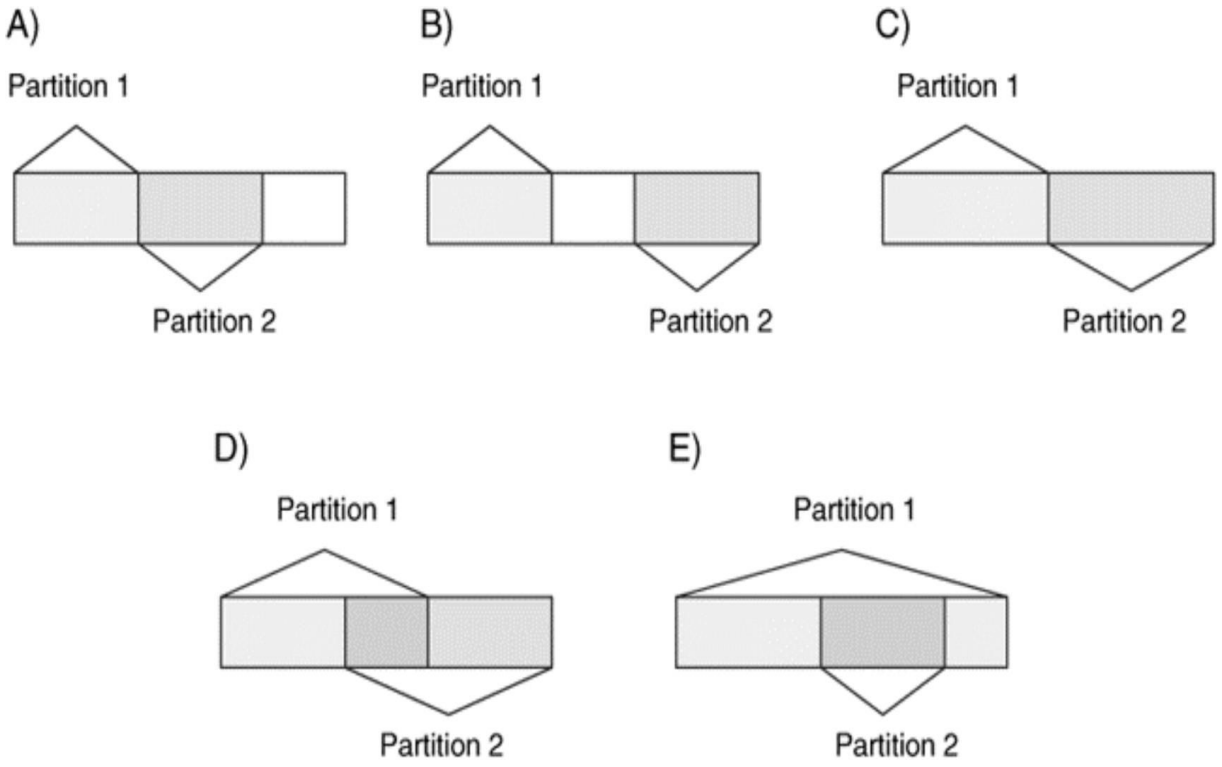
Source: Brian Carrier, "*File System Forensic Analysis*", Addison-Wesley Professional, March 27, 2005

Procedure

- performed automatically by tools most of the times
  - ✓ except if corruption has occurred
- steps that must be performed (by software or manually):
  - ✓ read partition table
  - ✓ identify the partition layout (start and end sectors)
  - ✓ analyze the unallocated space – it may contain data from a previous OS
  - ✓ a partition may be part of a volume with multiple partitions

Consistency checks

- does the last partition ends at the end of the parent volume?
- are the partitions consecutive?
- are there any overlap between partitions?
  - ✓ may happen if the partition table is corrupted



Source: Brian Carrier, "*File System Forensic Analysis*", Addison-Wesley Professional, March 27, 2005

How to recover partitions

- they may have been deleted to hinder the investigation
- or the partition table may have become corrupted
- usually partitions have file system, so we can search for their patterns
  - ✓ FAT has the values `0x55` and `0xAA` on bytes `510` and `511` of the first sector
- `gpart` tool tries to identify partitions based on patterns: `gpart -v disco.dd`
- `testdisk` is another tool to recover partition tables

Types of Partition Tables

- on personal computers (PCs)
  - ✓ MBR, Apple, removable storage media, GPT, ...
    - GPT is required for the UEFI secure boot on servers
  - ✓ GPT, FreeBSD, Sun Solaris, ...
  - ✓ PC partitions can also be used on servers
  - ✓ the main difference is the frequent use of logic volumes

Common Partitions of PCs

DOS partitions

https://en.wikipedia.org/wiki/Master_boot_record

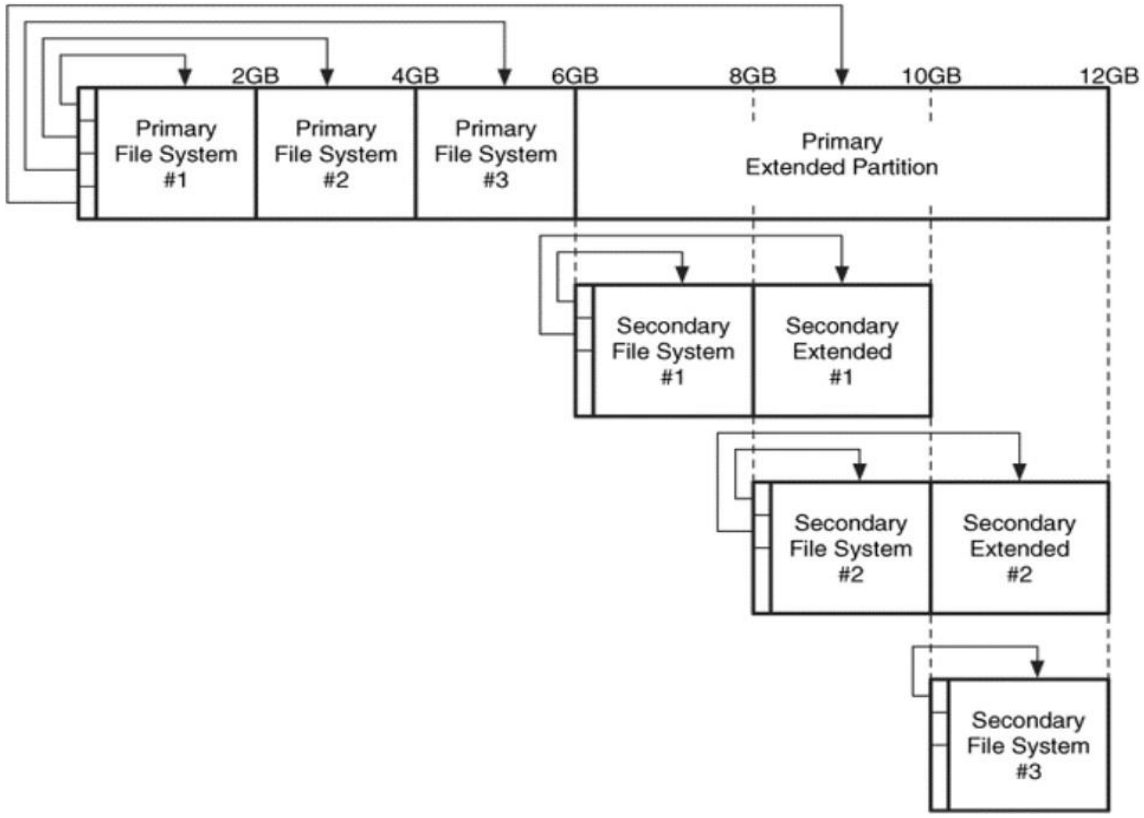
- also known as *Master Boot Record* (MBR)
- created by Microsoft (after Windows 2000 they call them *basic discs*)
- it's the most common partition table type
- it's used in: Microsoft DOS, Microsoft Windows, Linux, FreeBSD and OpenBSD
- MBR:
  - ✓ is located in the first sector (512 bytes)
  - ✓ *boot code* – instructions to process the partition table and to find the OS
  - ✓ partition table
  - ✓ pattern `0xAA55` – to identify the partition table

Structure of a DOS partition

- 4 entries – 4 primary partitions is the limit
- each one has:
  - ✓ begin and end address in CHS (< 8 GB) LBA address for large drives (several TB) amount of sectors in the partition
  - ✓ file system type stored in the partition (FAT, NTFS, EXT4, . . .)
    - Windows depends on this to mount the partition
    - it can be used to hide partitions from Windows OS
    - Linux ignores this value and supports a different FS from the one stored in the partition table
  - ✓ *flags* – allows to mark the boot partition (*bootable*)

Extended Partition

- to overcome the 4 primary partition limit
- always the last entry in the MBR
- allows to create several logical partitions
- types of extended partitions:
  - ✓ *DOS Extended*, *Windows 95 Extended* and *Linux Extended*
- usually there is only one extended partition
  - ✓ but it is possible to create more than one
  - ✓ few forensic tools support this



Source: Brian Carrier, "*File System Forensic Analysis*", Addison-Wesley Professional, March 27, 2005

Boot Code

Characteristics:

- located in the first 446 bytes of the first sector of 512 bytes (MBR)
- the boot code from Microsoft processes the partition table:
  - ✓ searches the bootable partition (with *boot flag* on)
  - ✓ the partition code is specific to each OS
- some virus are known to install themselves in MBR
- multiple OS:
  - ✓ boot selector from Windows
  - ✓ or MBR replaced by a specific application, like GRUB, LILO, . . .

Command to extract MBR in Linux

```
dd if=disco.dd bs=512 skip=0 count=1 | xxd
```

```
0000000: eb63 90d0 bc00 7c8e c08e d8be 007c bf00  .c....|......|..
...
0000180: 7de8 2e00 cd18 ebfe 4752 5542 2000 4765  }.......GRUB .Ge
0000190: 6f6d 0048 6172 6420 4469 736b 0052 6561  om.Hard Disk.Rea
00001a0: 6400 2045 7272 6f72 0d0a 00bb 0100 b40e  d. Error........
00001b0: cd10 ac3c 0075 f4c3 1996 6b49 0000 0020  ...<.u....kI...
00001c0: 2100 1cfe ffff 0008 0000 0000 2003 80fe  !........... ...
00001d0: ffff 07fe ffff 0008 2003 00c0 4917 00fe  ........ ...I...
00001e0: ffff 05fe ffff fecf 691a 0288 ce1f 0000  ........i.......
00001f0: 0000 0000 0000 0000 0000 0000 0000 55aa  ..............U.
```

- MBR with GRUB boot loader from Linux

Partitions on removable storage:

- floppy disks – don't have partition table
  - ✓ has a FAT12 file system as a single partition
- memory cards – typically use DOS partitions
  - ✓ FAT32 – max volume size 32 GB, max file size 4 GB
  - ✓ exFAT (or FAT64) – max volume size 128 PB, max file size 128 PB
- external hard drives
  - ✓ DOS partitions, commonly sold already with a NTFS file system in place

Optical disks:

- CDs – ISO 9660
  - ✓ the ISO 9660 is very strict about file names
  - ✓ there are extensions (Joliet, Rock Ridge) to overcome these limitations
  - ✓ there are also hybrid CDs:
    - ISO 9660 + Joliet
    - ISO 9660 + Apple HPS+
  - ✓ bootable CDs for *intel* PCs can contain a DOS partition

- CD-R
  - use sessions, each one can be treated as a partition
  - each time data is added to the CD-R a new session is created
  - most OS only show the last created session
    - ✓ on OS X it's possible to see all sessions
    - ✓ on Linux it's possible to manually *mount* previous sessions
    - ✓ on Windows specific software is required *e. g. Iso Buster* - *https://www.isobuster.com/*

Common Partitions of Servers

GPT (GUID Partition Table)

https://en.wikipedia.org/wiki/GUID_Partition_Table

- GUID – globally unique identifiers
- introduced on system with 64-bit Intel Itanium (IA64) processors
- is part of the *Unified Extensible Firmware Interface* (UEFI) standard

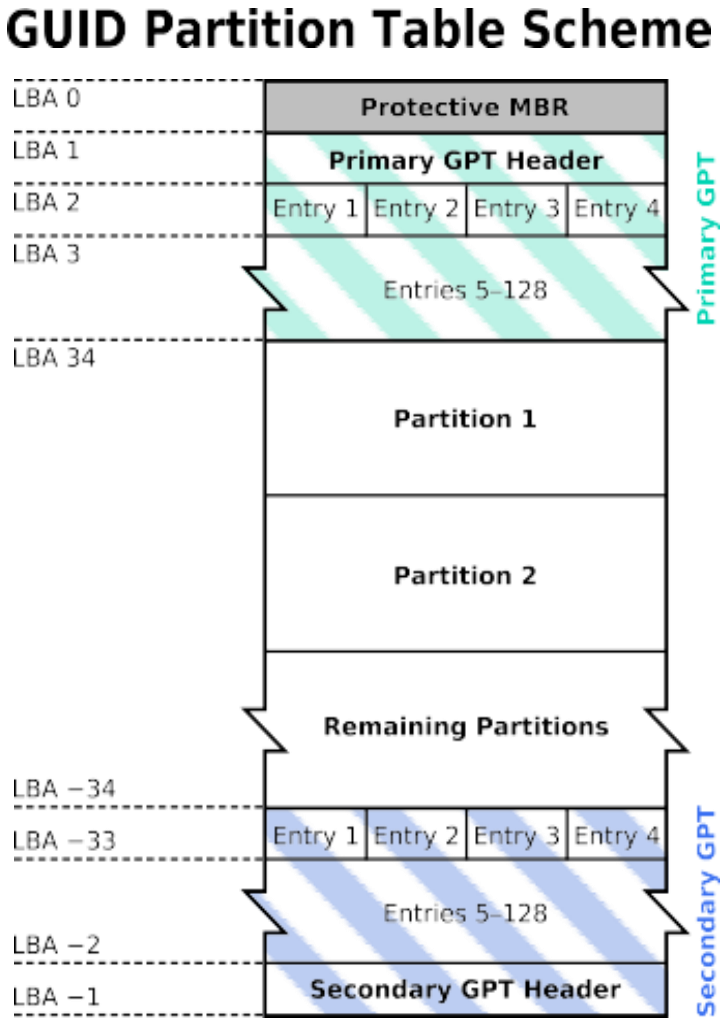
  ✓ replaces the BIOS and can also be used in PCs
- drives are identified with *globally unique identifiers* (GUID)
- uses 64 bits LBA
- Microsoft added support since Windows 2008

GPT Partitions

Structure:

- protective MBR – to prevent non-compatible OS to format the drive
- GPT header – starts on sector 1, defines the size and location of the partition table and has also a checksum
- partition table – supports up to 128 partitions, contains begin and end sectors, type, name, attributes and GUID values (128 bits)
- partition area – the main drive area
- backup area – located in the last sectors of the drive

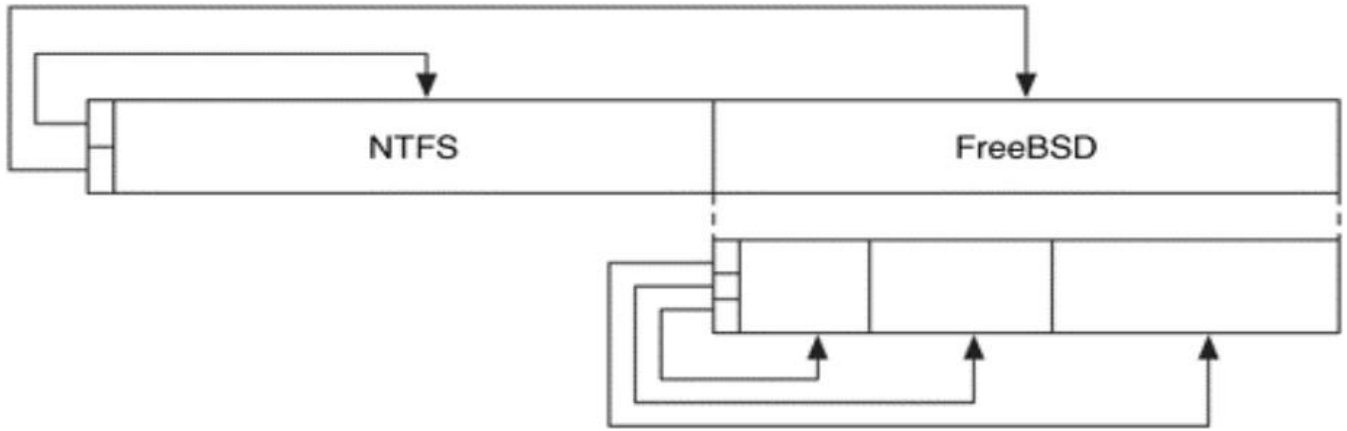
GPT Partitions



Source: https://en.wikipedia.org/wiki/GUID_Partition_Table

BSD Partitions

- used on BSD Unix: FreeBSD, OpenBSD and NetBSD
- BSD partitions can coexist with DOS partitions
  - ✓ BSD partition table will be located inside one of the DOS primary partition
  - ✓ FreeBSD allows access to DOS partitions

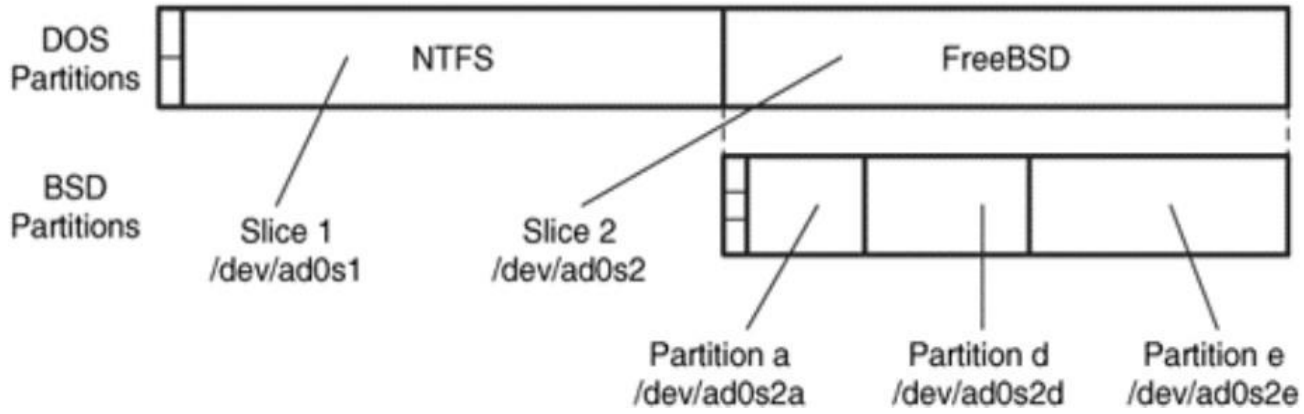


Source: Brian Carrier, "*File System Forensic Analysis*", Addison-Wesley Professional, March 27, 2005

BSD Partitions

Partitions naming scheme:

- ATA devices: `/dev/ad0`
- each DOS partition is a slice (`s`) → `/dev/ad0`**`s`**`1`, `/dev/ad0`**`s`**`2`, **`...`**
- each FreeBSD slice has a letter:
  - ✓ `a` – root partition
  - ✓ `b` – swap space
  - ✓ `c` – full FreeBSD partition
  - ✓ `d`, `e`, **`...`** – the remaining partitions

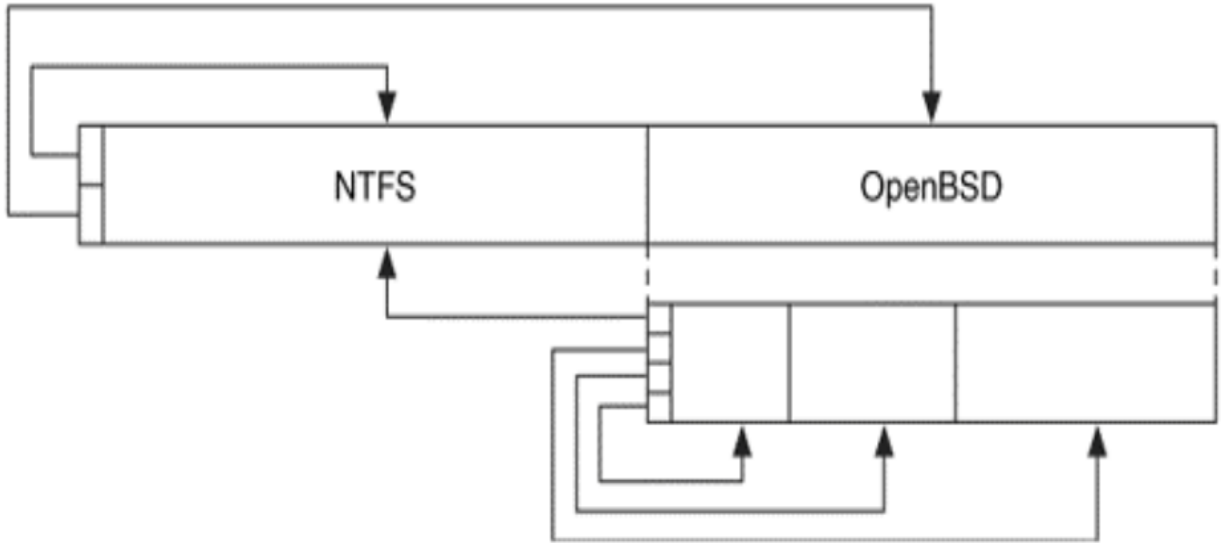


Source: Brian Carrier, "*File System Forensic Analysis*", Addison-Wesley Professional, March 27, 2005

OpenBSD and NetBSD Partitions

Main differences with FreeBSD partitions

- after boot, the OS ignores the DOS partition table
- allows to refer partitions outside its main area
- base name for the ATA devices: `/dev/wd0`
  - ✓ there are no *slices*
  - ✓ attributes letters to partitions like FreeBSD: `a`(root partition), `b`(*swap*), . . .



Source: Brian Carrier, "*File System Forensic Analysis*", Addison-Wesley Professional, March 27, 2005

Volumes' Aggregation

Why?

- improve performance
- prevent faults by adding redundancy
- gather free space from several drives

Types:

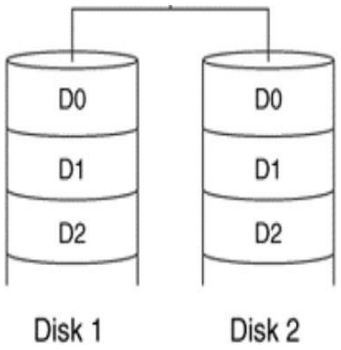
- RAID (Redundant Arrays of Inexpensive Disks)
  - ✓ common in high performance systems has many variants: RAID-1, RAID-5, . . .
  - ✓ can be implemented both in hardware, or software (usually at OS level)
- *spanning*
  - ✓ creates a logical volume by adding free space from several smaller volumes
  - ✓ just sums up space, it doesn't have any redundancy or performance gains

RAID Levels

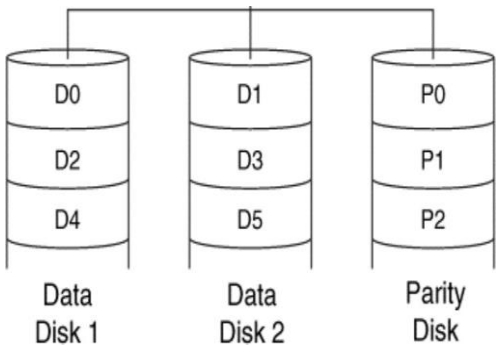
- RAID-0 – has no redundancy, but may increase performance
- RAID-1 – consists of an exact copy (or mirror) of a set of data on two or more disks
- RAID-2 – stripes data at the bit (rather than block) level, and uses a Hamming code for error correction **rarely used**
- RAID-3 – consists of byte-level striping with a dedicated parity disk **rarely used**
- RAID-4 – consists of block-level striping with a dedicated parity disk, provides good read performance
- RAID-5 – block-level striping with distributed parity among the drives
- RAID-6 – extends RAID 5 by adding another parity block and support up to 2 drives failures
- RAID hybrid – combination of more than one RAID level, *e. g.* RAID 10

RAID Schematics

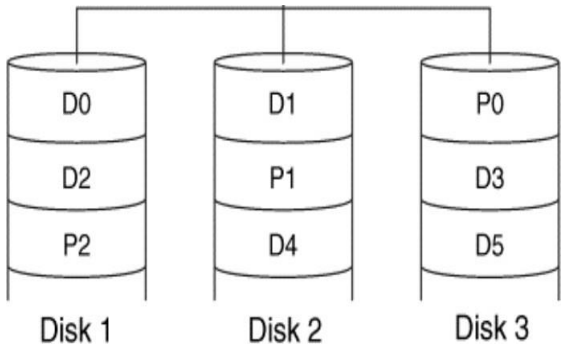
RAID-1



RAID-3/4



RAID-5



[Calculator of usable free space available on RAID volumes](http://www.raid-calculator.com/default.aspx) - http://www.raid-calculator.com/default.aspx

RAID by Hardware

Characteristics:

- requires specific controller
- guarantees best performance
- but is more expensive
- may require installation of drivers

Data acquisition:

- it's easier to acquire at logical level, as if it was a single disc
- acquisition OS must support RAID controllers
- individual acquisition of RAID discs:
  - ✓ only when the OS doesn't support RAID controllers
  - ✓ analysis is more complex – the RAID volume must be rebuilt

RAID by Software

Characteristics:

- implemented in the OS (supported by most modern OS)
- less efficient, depends on the CPU to calculate the parity bits data splits
- Windows – *Logical Disk Management* (LDM)
  - ✓ requires dynamic volumes
  - ✓ RAID volume configuration is stored on each drive
  - ✓ supports RAID 0, 1 and 5

Linux

- ✓ uses *Logical Volume Manager*
- ✓ saves meta data of the volume inside the drives
- ✓ uses volumes on DOS partitions
- ✓ supports RAID 0, 1, 5 and 6
- ✓ supports Windows LDM (may require kernel recompilation)
- ✓ allows the creation of snapshots – records only the changes and can be reverted to previous state

RAID by Software

Acquisition:

- it's easier to acquire at logical level, as if it was a single disc
- individual acquisition of RAID discs:
  - ✓ it's easier than hardware RAID systems
  - ✓ there are some tools to automatically rebuild the RAID volume
- Windows – with this OS a write blocker must be used
- Linux – it's possible to do a read only mount and it also supports LDM from Windows OS

Volume and Partition Tools

disktype tool

`disktype`

- identies many partitions types – Homepage: http://disktype.sourceforge.net/
- run this command to install it under Linux: `apt-get install disktype`
- download disk images for testing: http://dftt.sourceforge.net/

Example of a Joliet partition:

```
disktype iso9660-joliet.image

--- iso9660-joliet.image
Regular file, size 256 KiB (262144 bytes)
ISO9660 file system
    Volume name "ISO9660 Joliet"
    Application "MKISOFS ISO 9660/HFS FILESYSTEM BUILDER & CDRECORD CD-R/DVD CREATOR (C) 1993 E.YOUNGDALE
    Data size 256 KiB (262144 bytes, 128 blocks of 2 KiB)
    Joliet extension, volume name "ISO9660 Joliet"
```

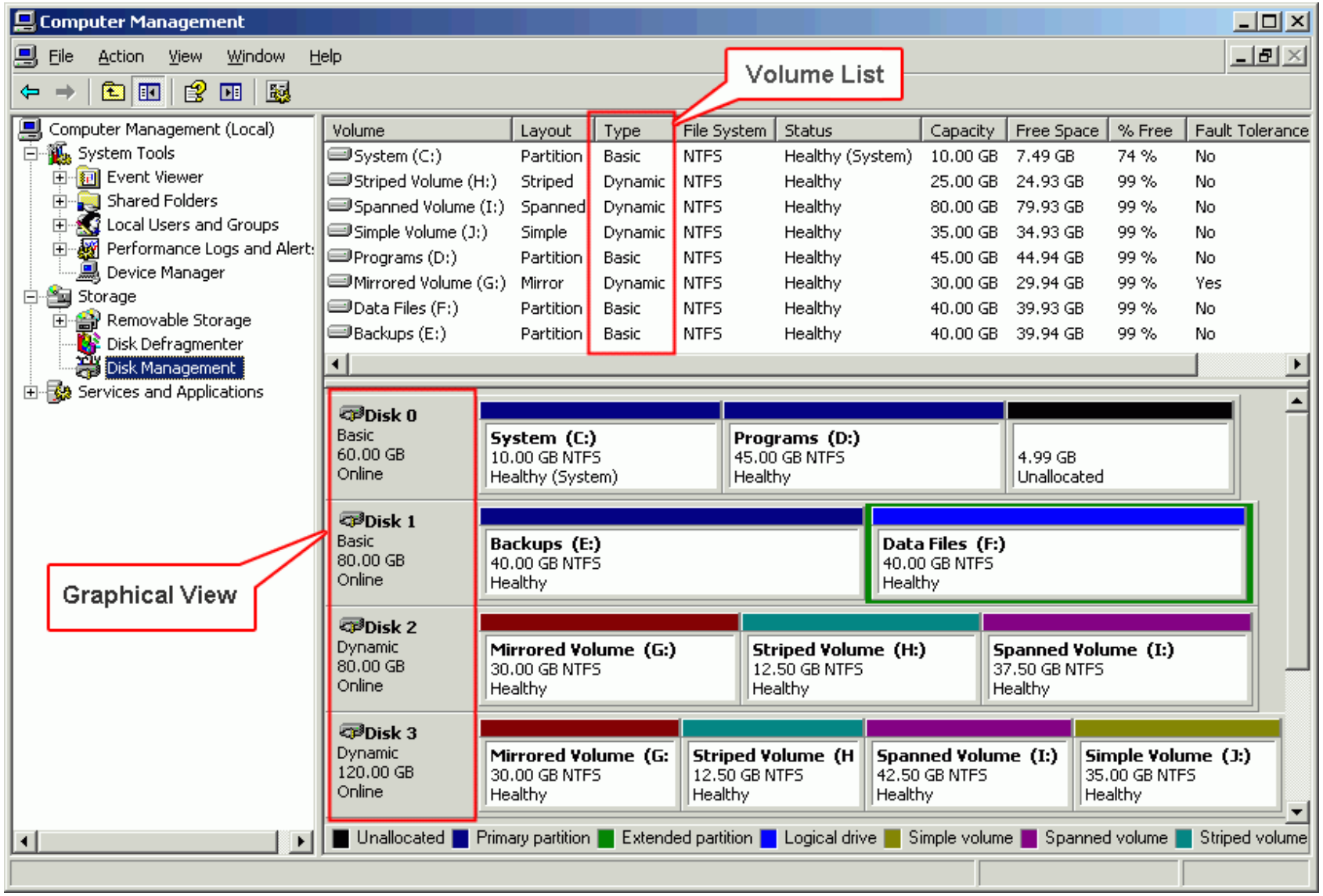
Example of a DOS partition

```
disktype /dev/sda

--- /dev/sda
Block device, size 20 GiB (21474836480 bytes)
DOS/MBR partition map
Partition 1: 19.14 GiB (20548943872 bytes, 40134656 sectors from 2048, bootable)
  Type 0x83 (Linux)
  Ext3 file system
    UUID FD935D1A-E410-4F97-BA7A-AE8A6B5C6E84 (DCE, v4)
    Last mounted at "/"
    Volume size 19.14 GiB (20548943872 bytes, 5016832 blocks of 4 KiB)
Partition 2: 880.0 MiB (922747904 bytes, 1802242 sectors from 40138750)
  Type 0x05 (Extended)
  Partition 5: 880 MiB (922746880 bytes, 1802240 sectors from 40138750+2)
    Type 0x82 (Linux swap / Solaris)
    Linux swap, version 2, subversion 1, 4 KiB pages, little-endian
      Swap size 880.0 MiB (922738688 bytes, 225278 pages of 4 KiB)
```

Example of a GTP partition

```
disktype gpt.image

--- gpt.image
Regular file, size 8 MiB (8388608 bytes)
DOS/MBR partition map
Partition 1: 8.000 MiB (8388096 bytes, 16383 sectors from 1)
  Type 0xEE (EFI GPT protective)
GPT partition map, 128 entries
  Disk size 8 MiB (8388608 bytes, 16384 sectors)
  Disk GUID 96117FCE-B25F-7D42-876F-8D5CB784DCF7
Partition 1: 7.967 MiB (8354304 bytes, 16317 sectors from 34)
  Type Basic Data (GUID A2A0D0EB-E5B9-3344-87C0-68B6B72699C7)
  Partition Name "test-ext2"
  Partition GUID 500E199D-6002-1248-8A72-88FB112FA191
  Ext2 file system
    UUID 726EA38B-087D-4FD9-9DD8-E42DD8D2E930 (DCE, v4)
    Volume size 7.967 MiB (8353792 bytes, 8158 blocks of 1 KiB)
Partition 2: unused
```

Windows Disc Management



Linux Volumes – LVM

Command line tool - See tutorial: https://www.thegeekstuff.com/2010/08/how-to-create-lvm/

- install: `apt-get intall lvm2`
- phase 1

  `pvcreate` – initialize drive or partition

  `pvscan` – search physical volumes

  `pvdisplay` – shows physical volumes attributes
- phase 2

  `vgcreate` – creates an aggregated volume – *volume group*

  `vgdisplay` – shows the attributes of the aggregated volume
- phase 3

  `lvcreate` – creates a logical volume

  `lvdisplay` – shows the attributes of the logic volume
- optional

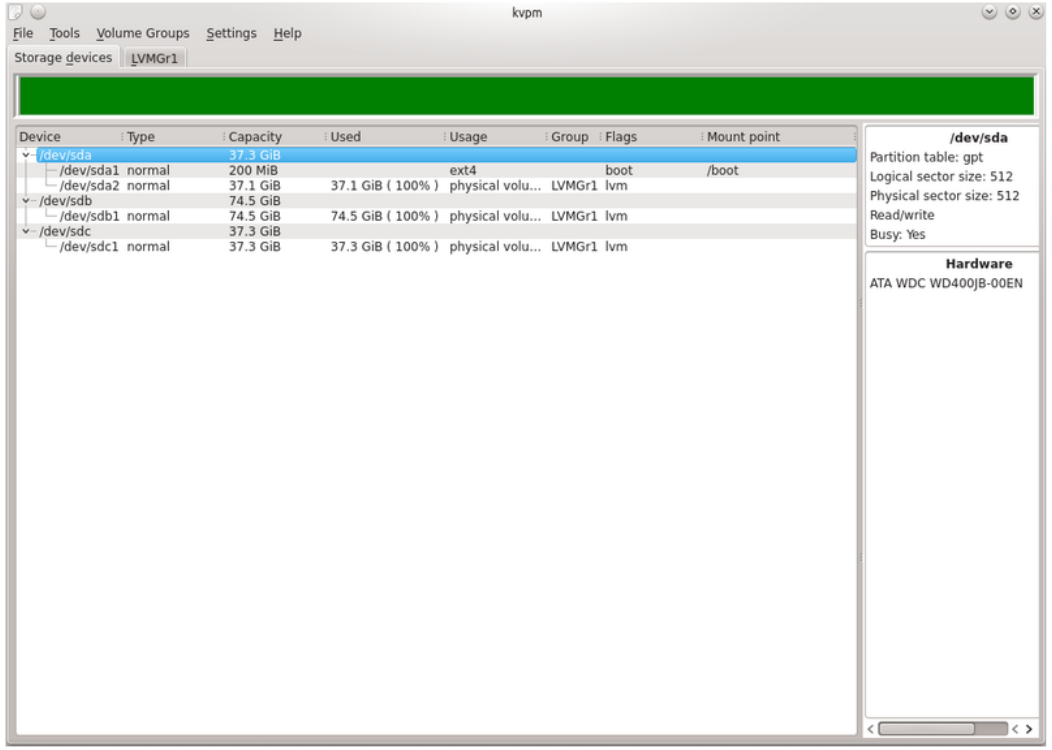
  `lvextend` – change the size of a logic volume

GUI tools

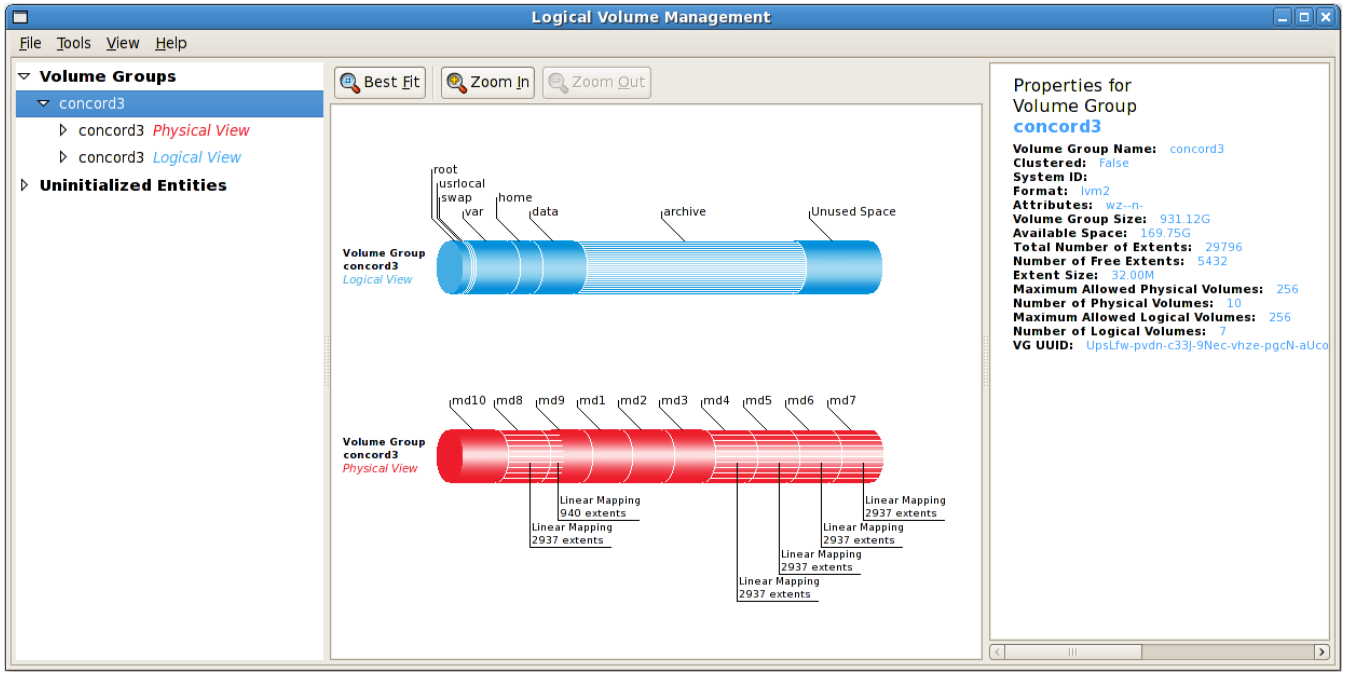
KVPM – KDE interface `apt-get install kvpm`

system-config-lvm – Gnome interface `apt-get install system-config-lvm`

KVPM



system-config-lvm



Exercises

07-Lab 1 – Identify partitions types with different tools

QUESTIONS?