**Network File systems**

The different between xfs to ext4 is that in xfs there is a better way to extending and the way to control inodes.

Block in the file system:

1. Super block - including meta-data of the file system, means info on the file system (free blocks, free inodes, file system type, last mount point, mount count, extended attribute (ACL), journal).

Without the super block we cannot do mount to the file system.

1. Inode – inode is the cell where the file is located and the meta-data of the files (file names, file size, modification time, access time, permissions, owner).

the inodes point to the 1st 12 data blocks (4B\*12=12k).

if there is more than 12 data-block so the inode will point on the ID (in-direct block), the ID point to another data-block with **ID(12k\*12=256k**), if it's still bigger then 256k the ID will point to **ID^2 (256ID\*256D=64M)**, if it's bigger then 64M it point to **ID^3 (256ID^2\*256ID\*256D=16B)**

the main different between **ext2** to **ext3** is that in ext3 there is journal and in ext2 there is no.

if I need server with a lot of small files it is recommended to use **xfs**, if I have a server with a lot of big files it is recommended to use **ext4.**

Ext2+ext3 can use 2TB not like ext4 that can use up to 16TB

Ext2 = journalאין

Ext3 = journalיש

**etc/fstab** you edit the mount options (here you mount the mount boot options)

/dev/vdb1 /mnt/fs1 ext4 noauto,nosync 0 1

UUID=\*\*\*\*\*\*\*\*\*\*\*\*\* swap swap 0 0

1st – part what disk we talk about

2nd – what point

3rd – what file type

4th – mount option

5th – file system check

6th – incremental dump backup

In 1st you can also describe the file system you want to mount wit UUID (you can find in with blkid)

In MBR you have 46b for partition table,

You have **Primary partition** that it is a static partition you cannot add logic partition, also you have **Extended partition** ere you can create a lot of Logic partitions

Sysctl –a | grep swap – show me the swapiness hardening (located in a file in /proc/sys/vm/swappiness

**\*\*What is swap?**

When you get to the limit of you RAM and you need to run another application but you don’t want to drop your process memory, you can put your file in available disk space – **swap = 82**, then when you launch the application again your data is taken from the swap space and reload to the same point it was before.

**Auto Master**

**Automount –** 1st use the service that called autofs. used for a temporary mount:

# auto.master

#auto.master.d 🡪 #.autofs 🡪#/guests or /etc/auto.guests

Inside /etc/auto.guests there is a configuration that sais for example that if guest 1 (that under /guest/guest1) so he will be mounted to a few mount options and I need to say where the share is located:

**\* -rw,nfs4 nfsserver.example.com:/home/guests/&**

*guest1 –MOUNT DESTINATION server1:/exports/guests/guest1*

In /etc/auto.master.d/ make file "\*.autofs" and inside this file you config on which folder he rule and which file is configuration file

Dmsg = netdev

df –h = mountשעשו להם file systemלראות

partprobe = לוקח מידע מהדיסק לעדכן את הקרנל

Disk – show disk space percent

blkid – shoe me the block info,

lsblk = mount devices list

lsof = open files that mounted

tune2fs –l – show me the details of ext superblock

tune2fs –j = file systemיוצר שם ל

tune2fs –L = journalיוצר

dumpe2fs – I can see the details like tune and also the groups detail of the inodes

fdisk – create the partitions for specific disk (MBR) [n-new, p-primary, e-extended, w-write and exit, p- output the details]

fdisk –l – show me the disks information

gdisk – create the partitions for specific disk (GPT)

parted – like fdisk

mkfs.ext4 /dev/vdb – convert to ext4 **(ext4 = 83)**

mkswap /dev/vdb6 – make partition as swap

swapon –a – apply the swap partition for all swaps follow the /etc/fstab

swapon /dev/vdb2 – enable swap on the specific path

swapon –s – shoe me the swap details

sysctl –a | grep swap = /proc/sys/vm/swappiness