

# ADMINISTRACIJA OTVORENIH OPERACIJSKIH SUSTAVA

Lokalna administracija I.:  
Boot process, Disk



# Linux boot

BIOS obavlja radnje specifične za hardware

BIOS pokreće boot kôd s označenog uređaja

Pokreće se bootloader s opcijama boot-a

Start kernela

Procesi u /etc direktoriju

MBR – Master Boot record

GRUB, LILO...

Driveri...

PID 1 i svi ostali procesi

# Linux bootloaderi

- Dva najčešća
  - **LILO**
    - Agnostičan prema datotečnim sustavima
    - Koristi „sirove” podatke s diska s predefiniranim pozicijama diska
  - **GRUB**
    - Razumije ext2, ext3 i ext4
- Postoje još
  - SYSLINUX (Za boot s USB-a ili CD-a)
    - Razumije FAT, NTFS
  - Loadin (za boot iz DOS-a ili Win9x)

# Kratka povijest procedure pokretanja sustava na Linuxu

- Dugi niz godina, prvi proces koji se pokreće na Linuxu bio je init (PID=1)
- On je bio "parent" proces svim kasnijim procesima i bio je zadužen da pri pokretanju sustava pokrene i ostale procese
- Startanje servisa se odvijalo kroz SysV/init **skripte** (skripta per servis, indeksiranje redoslijeda)
- Service, chkconfig

# CentOS6 vs 7

Item	RHEL6/CentOS6	RHEL7/CentOS7
Boot	init process	systemd - Move in parallel Faster booting
Linux kernel	2.6.x-x	3.10.x-x
File System	ext4	xfs
NIC	eth0 Start Number is 0	eno1 : en = Ether Net , o = On board ens1 : en = Ether Net , s = PCI Start Number is 1
Log	rsyslog	journal(Binary) と rsyslog(Text)
ntp	ntpd	chronyd
FW	iptables	firewalld or iptables (Only one can be used)
Location of runtime data after boot	/var/run	/run

# Systemd (system and service manager) 1/2

- Zamjenio „init.d” (SysVInit)
- Kreirao ga Red Hat
- Prednosti:
  - Brže boot-anje jer „daemons” se pokreću u paraleli
  - Jednostavno za developere jer ne moraju razmišljati koji daemon moram pokrenuti prvo...
  - Daemons se sam može restartati (npr. kod crash-a)
  - Daemons se može sam zaustaviti/pokrenuti po potrebi
  - Bolja sigurnost – auditd (ako netko provali sustav – **biti će zabilježeno** - jer taj daemon se ne može ugasiti/zaustaviti! Čak ni sa root ovlastima)
  - Svaki proces ima svoju cgroup
  - "bye bye, shell scripts for starting services"

# Prije i poslije...

```
donnie@centos-class:/etc/rc5.d
File Edit View Search Terminal Help
[donnie@centos-class rc5.d]$ ls
K01smartd      K75quota_nld      S10network      S26hald daemon
K02oddjobd     K76ypbind         S11auditd       S26udev-post
K05wdaemon     K04wpa_supplicant S11portreserve  S28autofs
K10saslauthd   K86cgred          S12rsyslog      S55sshd
K15htcacheclean K87restorecond   S13cpuspeed     S56xinetd
K15httpd       K88sssd          S13irqbalance  S58ntpd
K15svnserve    K89netconsole    S13rpcbind      S70spice-vdagentd
K35vncserver   K89rdisc         S15mdmonitor    S80postfix
K50dnsmasq     K92pppoe-server  S22messagebus  S82abrt-ccpp
K50kdump       K95firstboot     S23NetworkManager S82abrt d
K60nfs         K95rdma          S24nfslock      S90crond
K61nfs-rdma    K99rngd          S24rpcgssd     S90psacct
K69rpcsvcgssd S01sysstat       S25blk-availability S95atd
K73winbind     S02lvm2-monitor  S25cups         S99certmonger
K75cgconfig    S08ip6tables     S25netfs        S99local
K75ntpdupdate  S08iptables     S26acpid
```

```
[root@dev ~]# pstree
systemd
├── ModemManager──2*[{ModemManager}]
├── NetworkManager──dhclient
│   └──2*[{NetworkManager}]
├── abrt-dbus──(abrt-dbus)
├──2*{abrt-watch-log}
├── abrt d
├── accounts-daemon──2*[{accounts-daemon}]
├── alsactl
├── at-spi-bus-laun──dbus-daemon──(dbus-daemon)
│   └──3*[{at-spi-bus-laun}]
├── at-spi2-registr──(at-spi2-registr)
├── atd
├── auditd──auditd
│   ├── auditd
│   └── sedispatch
│     └──(auditd)
├── avahi-daemon──avahi-daemon
├── bluetoothd
├── caribou──2*[{caribou}]
├── chronyd
├── colord──2*[{colord}]
├── crond
├── cupsd
├──2*{dbus-daemon──(dbus-daemon)}
├── dbus-launch
├── dconf-service──2*[{dconf-service}]
├── dnsmasq──dnsmasq
├── dockerd──docker-containe──10*[{docker-containe}]
│   └──13*[{dockerd}]
├── evolution-calen──5*[{evolution-calen}]
├── evolution-sourc──2*[{evolution-sourc}]
├── firewalld──(firewalld)
├── fprintd
├── gconfd.2
├── Xorg──3*[{Xorg}]
├── gdm──gdm-session-wor──2*[{gdm-session-wor}]
│   ├── gdm-session-wor
│   └── gnome-session──gnome-session
│       ├── abrt-applet──(abrt-applet)
│       ├── evolution-alarm──5*[{evolution-alarm}]
│       ├── gnome-settings──5*[{gnome-settings}]
│       └── gnome-shell──HipChat.bin──QtWebEngineProc──QtWebEngineProc──12*[{QtWebEngineProc}]
│           ├── QtWebEngineProc──5*[{QtWebEngineProc}]
│           ├──31*[{HipChat.bin}]
│           ├── firefox──769*[{firefox}]
│           ├── ibus-daemon──ibus-dconf──3*[{ibus-dconf}]
│           ├── ibus-engine-sim──2*[{ibus-engine-sim}]
│           └──2*[{ibus-daemon}]
├──6*[{gnome-shell}]
├── nautilus──3*[{nautilus}]
├── seapplet──2*[{seapplet}]
└── ssh-agent
```

# Systemd (system and service manager) 2/2

- **systemctl** upravlja različitim tipovima systemd objekata koji se popularno zovu **units**
- ako želimo pogledati vrste, `systemctl -t help`
- uobičajeni tipovi unita:
  - **services** - imaju **.services** ekstenziju - sistemski servisi
  - **socket units** - **.socket** ekstenzija i koriste se za upravljanje IPC-a (inter-process komunikacija)
  - **path units** - **.path** ekstenzija i koriste se za odgađanje aktivacije servisa dok se ne dogodi neka zadana promjena na filesystemu (npr. spooling servisi koji koriste direktorije za spooling)



# Korisne komande

- `systemctl poweroff` – gašenje računala
- `systemctl reboot` – reboot računala
- `systemctl status name.service` – provjera da li servis radi
- `systemctl enable name.service` – pokretanje servisa prilikom bootanja
- `systemctl start name.service` - pokretanje servisa odmah
- `systemctl list-unit-files --type service` - lista svih servisa i provjera da li rade
- `systemctl --failed` - koji servisi se nisu pokrenuli

# Par linkova...

- The Tragedy of systemd
  - [https://www.youtube.com/watch?v=o\\_Alw9bGogo](https://www.youtube.com/watch?v=o_Alw9bGogo)
- Why "systemd"?
  - [https://www.youtube.com/watch?v=KftuGM\\_ylKg](https://www.youtube.com/watch?v=KftuGM_ylKg)

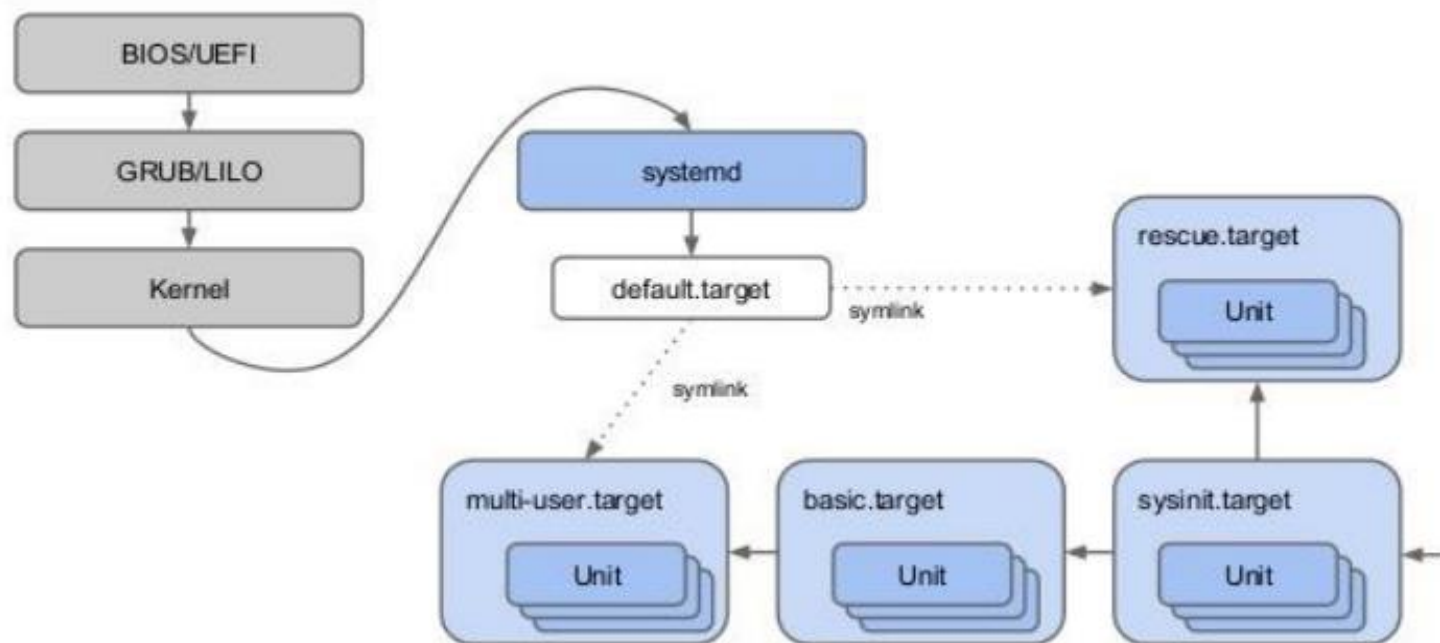
# Demo

- Pokretanja httpd servisa

**BOOT proces +  
demo**

# Boot process – CentOS 7

## systemd boot process



# GRUB (Grand Unified Bootloader)

- GRUB v2
- Konfiguracijske datoteke:
  - /boot/grub2/grub.cfg
  - /etc/default/grub

```
student@vm68-118:/etc/default — less grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="resume=/dev/mapper/cs_vm68--100-swap rd.lvm.lv=cs_vm68-100/
root rd.lvm.lv=cs_vm68-100/swap rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
GRUB_ENABLE_BLSCFG=true
```

# Boot – intrd/intramfs

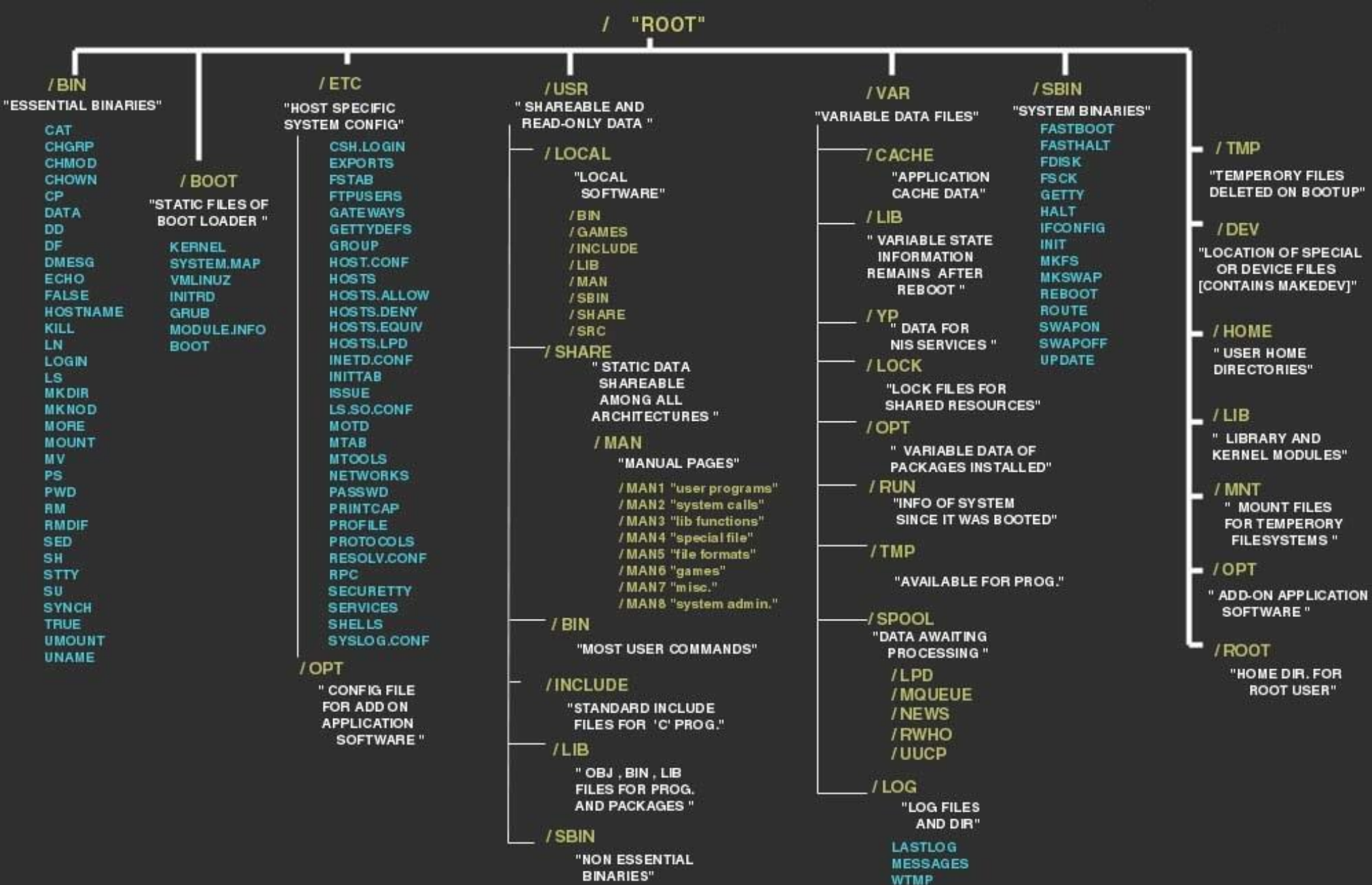
- Sadržaj boot direktorija
- Initramfs - initial ram disk file system
- PID 1 – systemd
  - system and service manager

```
student@vm68-118:~  
[student@vm68-118 ~]$ ls /boot  
config-5.14.0-142.el9.x86_64  
config-5.14.0-165.el9.x86_64  
config-5.14.0-171.el9.x86_64  
efi  
grub2  
initramfs-0-rescue-514619100fc74016aff194acd9ceed2d.img  
initramfs-5.14.0-142.el9.x86_64.img  
initramfs-5.14.0-165.el9.x86_64.img  
initramfs-5.14.0-171.el9.x86_64.img  
loader  
symvers-5.14.0-142.el9.x86_64.gz  
symvers-5.14.0-165.el9.x86_64.gz  
symvers-5.14.0-171.el9.x86_64.gz  
System.map-5.14.0-142.el9.x86_64  
System.map-5.14.0-165.el9.x86_64  
System.map-5.14.0-171.el9.x86_64  
vmlinuz-0-rescue-514619100fc74016aff194acd9ceed2d  
vmlinuz-5.14.0-142.el9.x86_64  
vmlinuz-5.14.0-165.el9.x86_64  
vmlinuz-5.14.0-171.el9.x86_64  
[student@vm68-118 ~]$
```

```
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND  
root      1  0.8  0.4 106312 16092 ?        Ss   11:58   0:02 /usr/lib/systemd/systemd rhgb --switched-root -  
-system --deserialize 31
```

# **Diskovi i particije pod Linuxom**





Gdje je C:?  
CDROM?  
Gdje su diskovi?

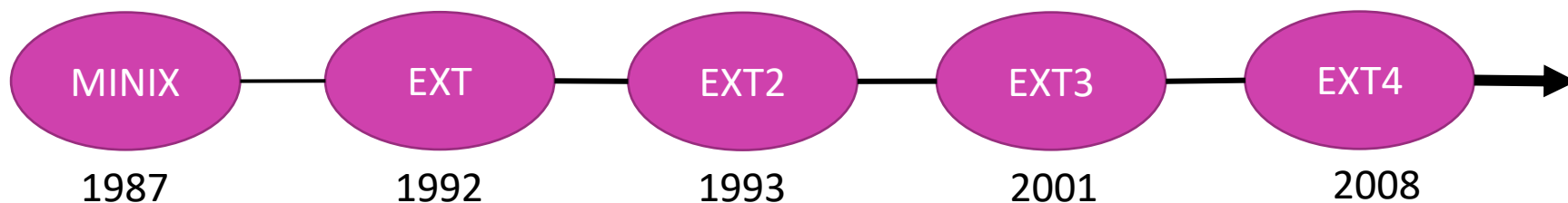
# Datotečni sustav

- Datotečni sustav može se definirati kao način pohranjivanja podataka na određeni način kako bi se mogli lako pronaći kada je to potrebno.
- Ovisno o operativnom sustavu, ovisi o tome koja je vrsta datotečnog sustava podržana.
  - Glavna svrha datotečnog sustava je olakšati korisnicima i operativnom sustavu pohranjivanje datoteka na određeni način tako da sve strane mogu s lakoćom pohraniti ili dohvatiti datoteke.
  - Kao što naziv kaže, datotečni sustav bavi se datotekama i direktorijima koji bi se trebali čuvati u uređaju za pohranu (disk, USB, netdrive...) pomoću bilo kojeg operativnog sustava. Operativni sustav samo igra ulogu posrednika koji olakšava prijenos podataka na uređaj za pohranu kako bi se oni tamo mogli pohraniti.

# Linux datotečni sustavi

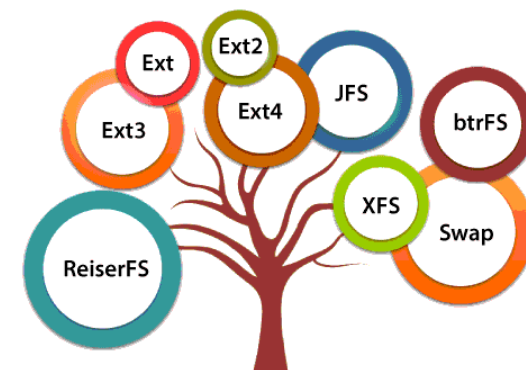
- Windows koristi: FAT i NTFS
- Linux ih ima oko 100...
  - Najpoznatiji datotečni sustavi:
    - EXT (*extended file system*) familiji (ext, ext2, ext3, ext4)
    - XFS (razvio Silicon Graphics)
- Linux koristi dva tipa particije:
  - data partitions
  - swap - za virtualnu memoriji (gdje to Windows zapisuje?)

# Linux datotečni sustav



- Minix - veličina particije do 64MByte
- EXT - veličina particije do 2GiByte, veličina naslova datoteke do 265 karaktera
- EXT2 - nadogradnja EXT, podrška za Linux kernel
  - problemi sa korupcijom podataka
  - problemi sa fragmentacijom diska
- EXT3 - nadogradnja EXT2, veličina particije do 2TByte
- EXT4 - veličina particije do 16TiByte
  - podrška za journaling
- XFS - veličina particije do 8EiByte
  - default za RedHat
- OpenZF - veličina particije do 16EiByte

Types of Linux File System



# Particiniranje diskova

- DEMO

- Naredbe:

- `fdisk`
- `lsblk`
- `parted`
- `df`
- `du`

# Pitanja?



**Hvala na pažnji!**