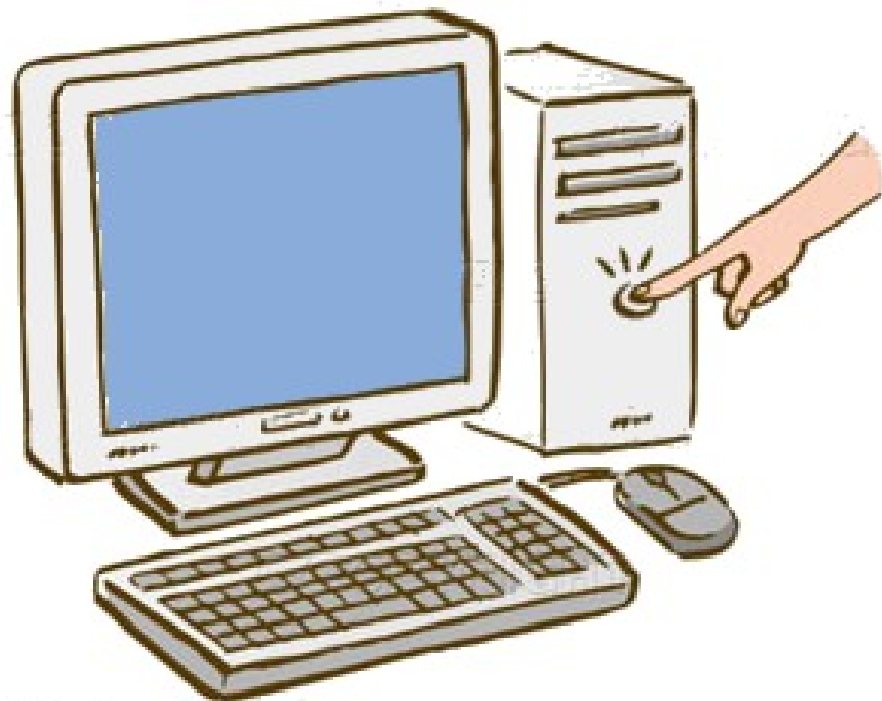


# Zagon in zaustavitev računalnika



# Kaj mora administrator razumeti?

- Zaporedje dogodkov pri zagonu sistema.
- Metode, ki jih uporabimo za spreminjanje zagonskega zaporedja.
- Kako izbiramo alternativne zagonske naprave.
- Delovanje programov “boot manager” in “boot loader”.
- Kako pravilno zaustavimo sistem.

# Zakaj je potreben zagon ?

- Aparaturna oprema ne ve, kje leži operacijski sistem in kako naj ga naloži.
- Za to potrebujemo poseben program – “bootstrap loader”.
  - Na primer: BIOS – Boot Input Output System.
- “Bootstrap loader” locira jedro (operacijskega sistema), ga naloži v pomnilnik in sproži njegovo izvajanje.
- V nekaterih primerih pa preprosto “bootstrap loader” poišče na disku bolj kompleksen zagonski program, ga naloži v pomnilnik, ta pa nato naloži jedro (kernel).



# Kako poteka zagon (boot) ?

- Dogodek “reset” na CPE (vklop računalnika, ponovni zagon) povzroči, da se programski števec nastavi na predoločeni naslov v pomnilniku. Sproži se program “Bootstrap” na tej lokaciji.
- Ta program je pomnjen v ROM, saj je pomnilnik RAM takrat v neznanem stanju. ROM je tako primeren za inicializacijo, nanj pa tudi ne vplivajo virusi.

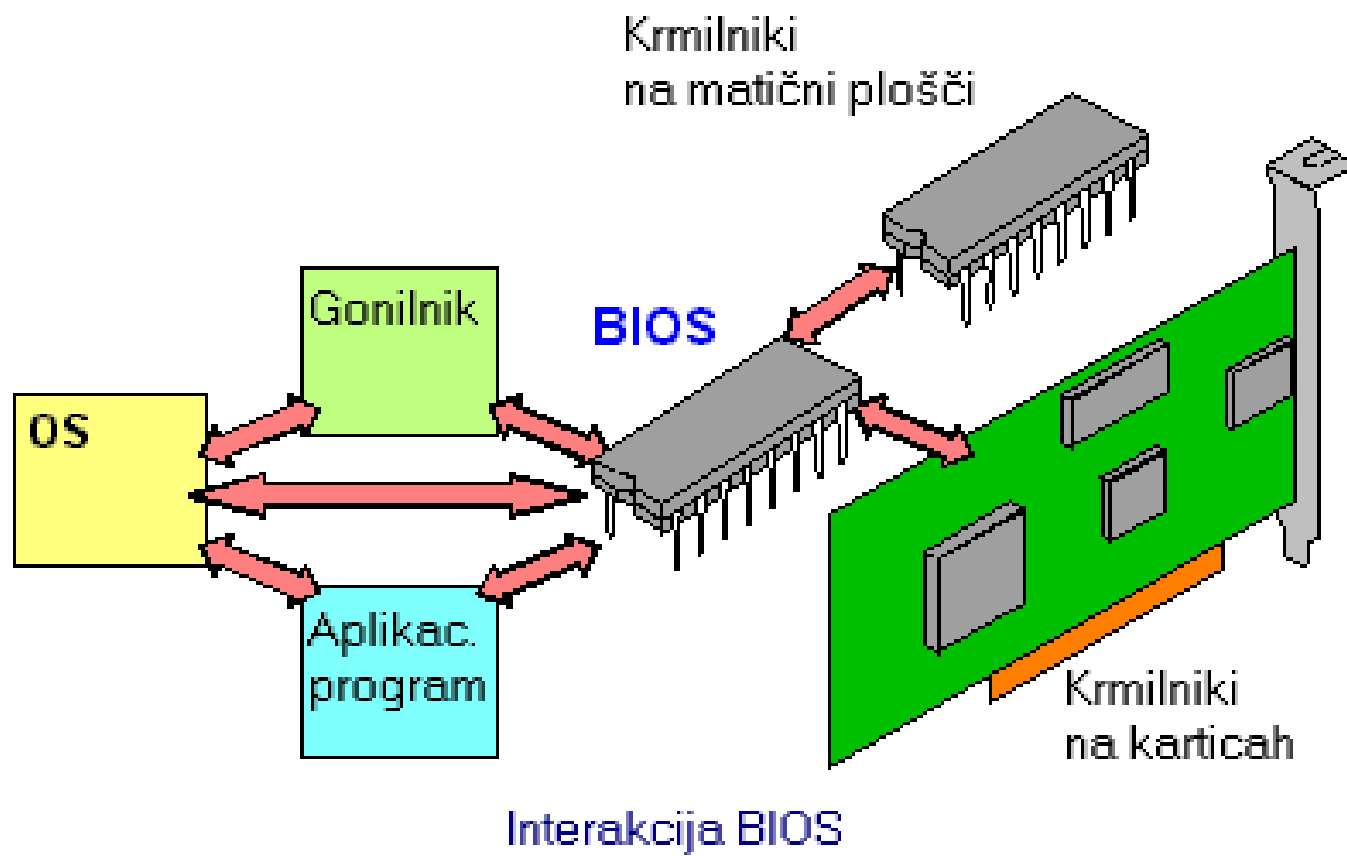


# BIOS Info

- BIOS na matični plošči.
- CMOS pomnilnik
  - pomni ključne začetne podatke



# Interakcija BIOS

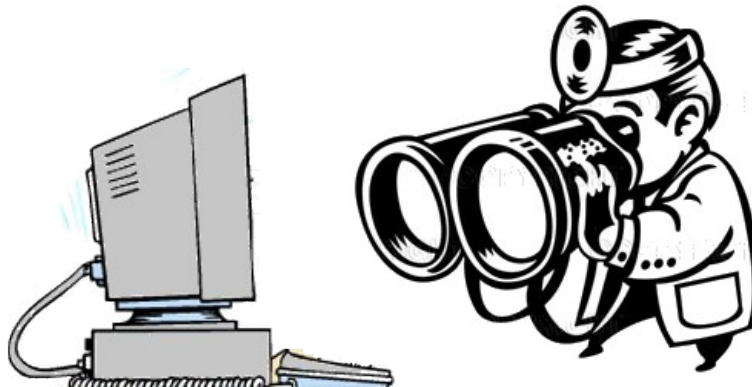


# BIOS Setup

PhoenixBIOS Setup Utility							
Main	Advanced	Security	Power	Boot	Exit		
ATAPI CD-ROM Drive +Removable Devices +Hard Drive Network Boot				Item Specific Help			
				Keys used to view or configure devices: <Enter> expands or collapses devices with a + or - <Ctrl+Enter> expands all <Shift + 1> enables or disables a device. <+> and <-> moves the device up or down. <n> May move removable device between Hard Disk or Removable Disk <d> Remove a device that is not installed.			
F1	Help	↑↓	Select Item	-/+	Change Values	F9	Setup Defaults
Esc	Exit	←	Select Menu	Enter	Select ► Sub-Menu	F10	Save and Exit

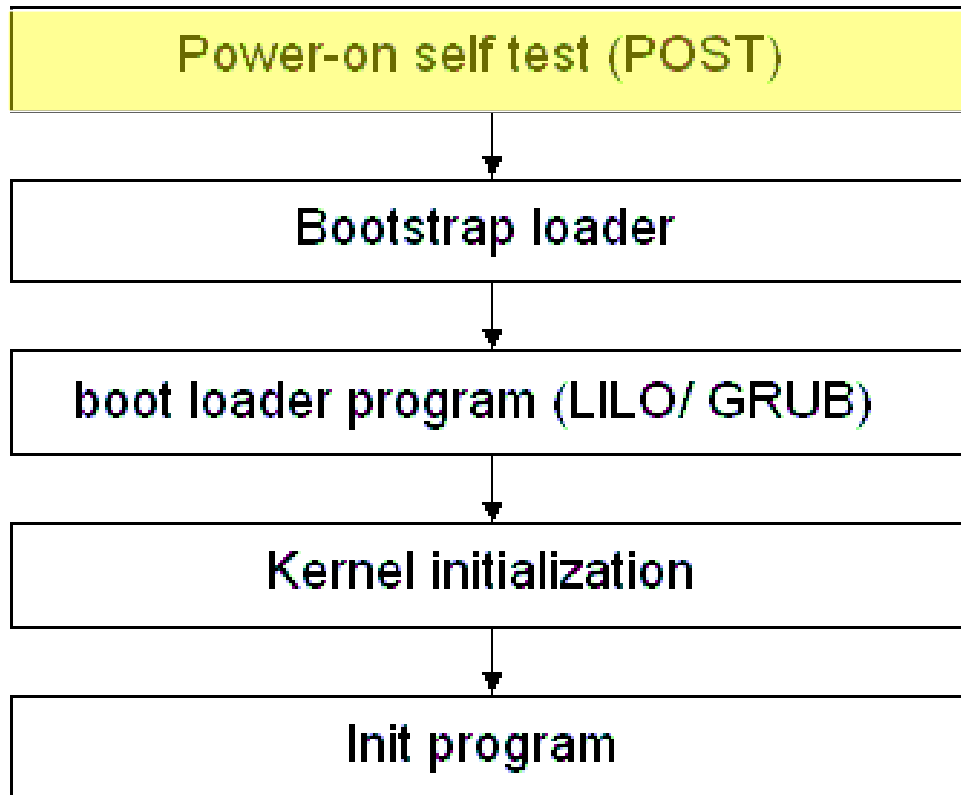
# Naloge ob zagonu

- Diagnostika, ki določi stanje stroja. Če je diagnostika uspešna, se zagon nadaljuje.
- Izvedba “Power-On Self Test” (*POST*), ki preveri, ali naprave, ki sestavljajo računalnik, delujejo.
- BIOS gre preko predkonfiguriranih naprav in poišče tisto, ki je zagonška (bootable). Če take naprave ne najde, sledi obvestilo o napaki in zagon se zaustavi.
- Inicializacija registrov CPE, krmilnikov naprav in vsebine pomnilnika. Zatem sledi nalaganje operacijskega sistema.





# POST



Preveri prisotnost pričakovanih naprav  
(le osnovno delovanje, avtomatsko preverjanje)

# Zagonski postopek (boot procedure)

```
PU Clock      : 2000MHz          L1 Cache Size  : 128K
                                           L2 Cache Size  : 256K

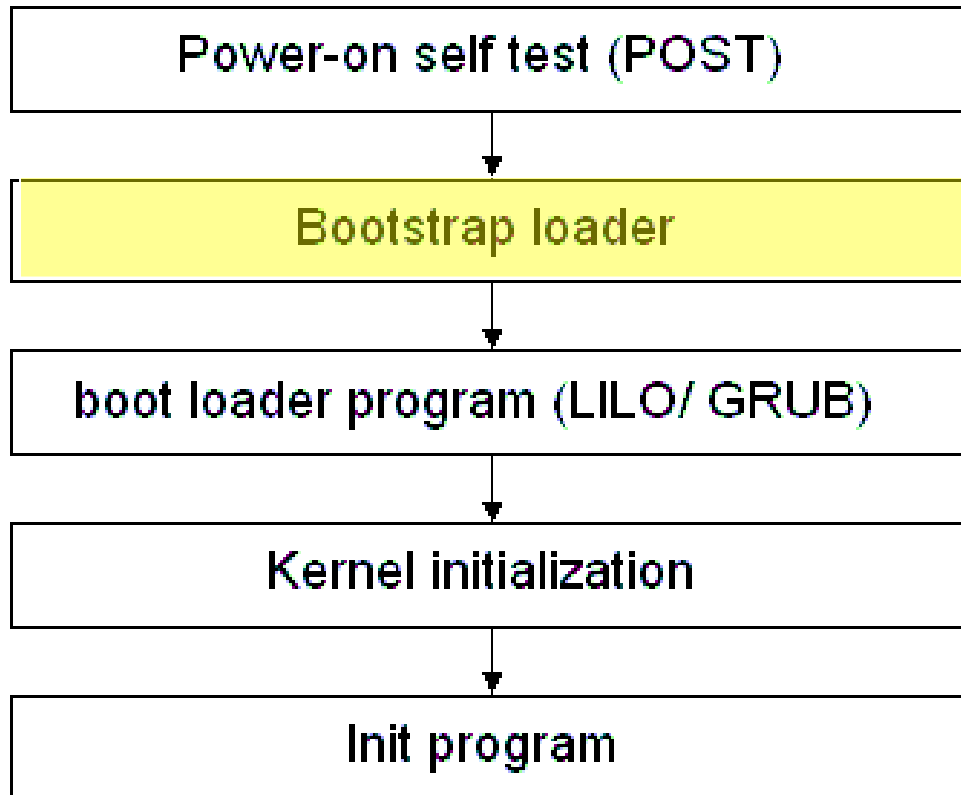
iskette Drive A : 1.44M, 3.5"    Display Type   : EGA/UGA
iskette Drive B : None          Serial Port(s) : 3F8 2F8
ri. Master Disk : None          Parallel Port(s) : 378
ri. Slave Disk  : None          DDR SDRAM at Bank : 0 1
ec. Master Disk : None
ec. Slave Disk  : None
```

## Devices Listing ...

Dev	Fun	Vendor	Device	SUID	SSID	Class	Device Class	IRQ
16	0	1106	3038	1458	5004	0C83	USB 1.1 Host Cntrlr	10
16	1	1106	3038	1458	5004	0C83	USB 1.1 Host Cntrlr	10
16	2	1106	3038	1458	5004	0C83	USB 1.1 Host Cntrlr	11
16	3	1106	3104	1458	5004	0C83	USB 2.0 Host Cntrlr	11
17	1	1106	0571	1458	5002	0101	IDE Cntrlr	14
17	5	1106	3059	1458	A002	0401	Multimedia Device	11
19	0	100C	0139	100C	0139	0200	Network Cntrlr	11
0	0	100E	002D	0000	0000	0300	Display Cntrlr	10
							ACPI Controller	9

```
ifying DMI Pool Data ..... Update Success
K BOOT FAILURE, INSERT SYSTEM DISK AND PRESS ENTER
```

# Bootstrap Loader



# Terminologija zagonskega mehanizma

## ● Loader

- Naloži kodo (običajno) z diska v pomnilnik in sproži izvajanje te kode)

## ● Bootloader / Bootstrap

- Program, ki naloži “prvi program” (jedro oziroma kernel)

## ● Boot PROM / PROM Monitor / BIOS

- Fiksna koda, ki je ob vklopu računalnika “že naložena”

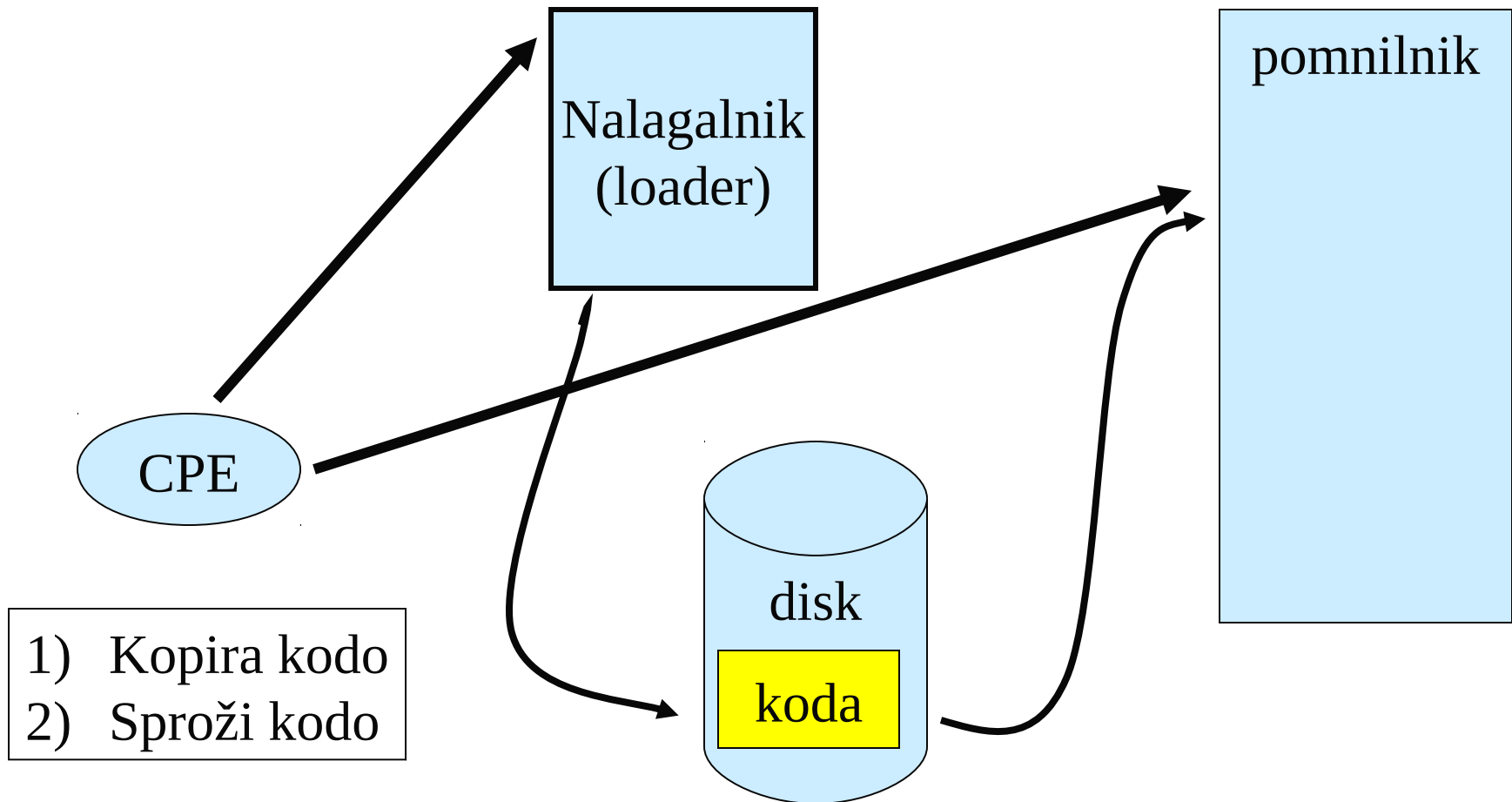
## ● Boot Manager

- Program, ki nam omogoča izbrati “prvi program”, ki naj bo naložen



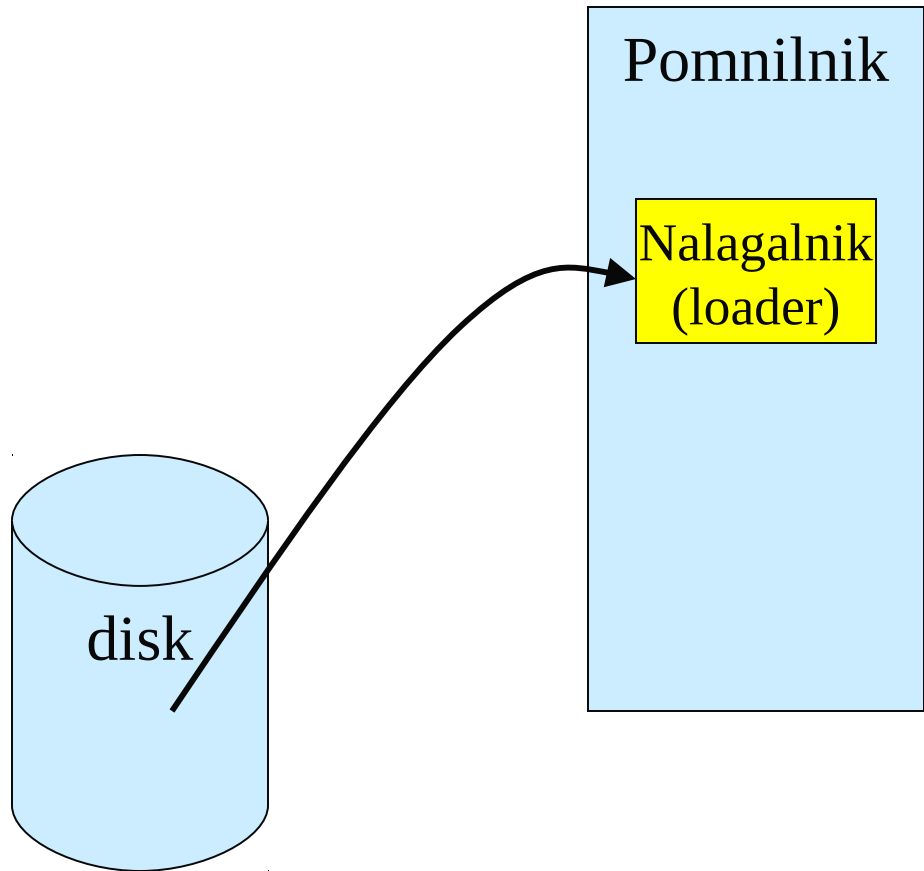
# Kaj je nalagalnik (loader)?

Program, ki kopira kodo (običajno z diska) v pomnilnik in sproži izvajanje te kode.



# Kdo pa naloži nalagalnik?

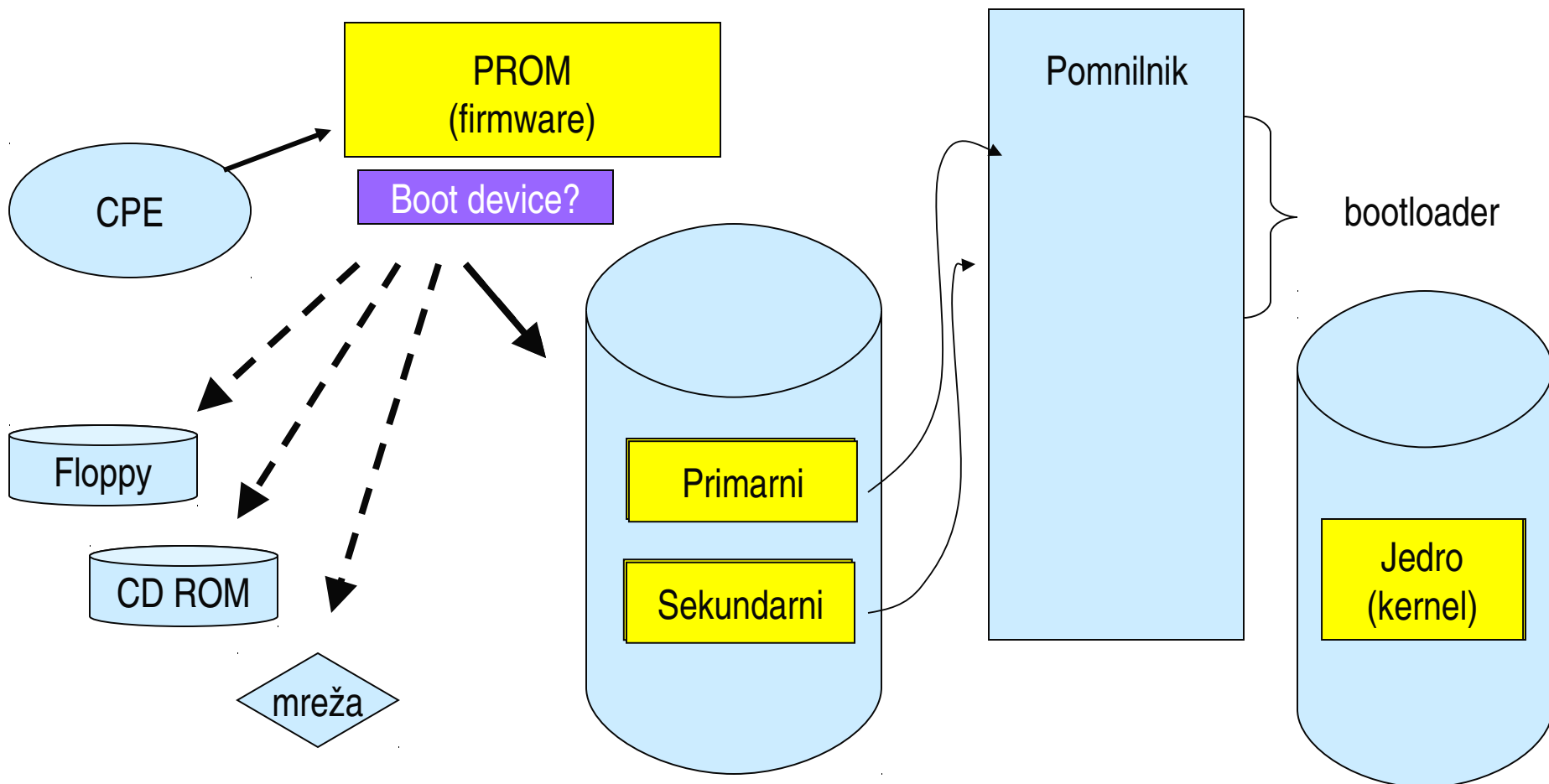
- Seveda je tudi nalagalnik le program, ki leži v pomnilniku. Kako pa je tja prišel?



- Potrebujemo “nalagalnik nalagalnika” ...

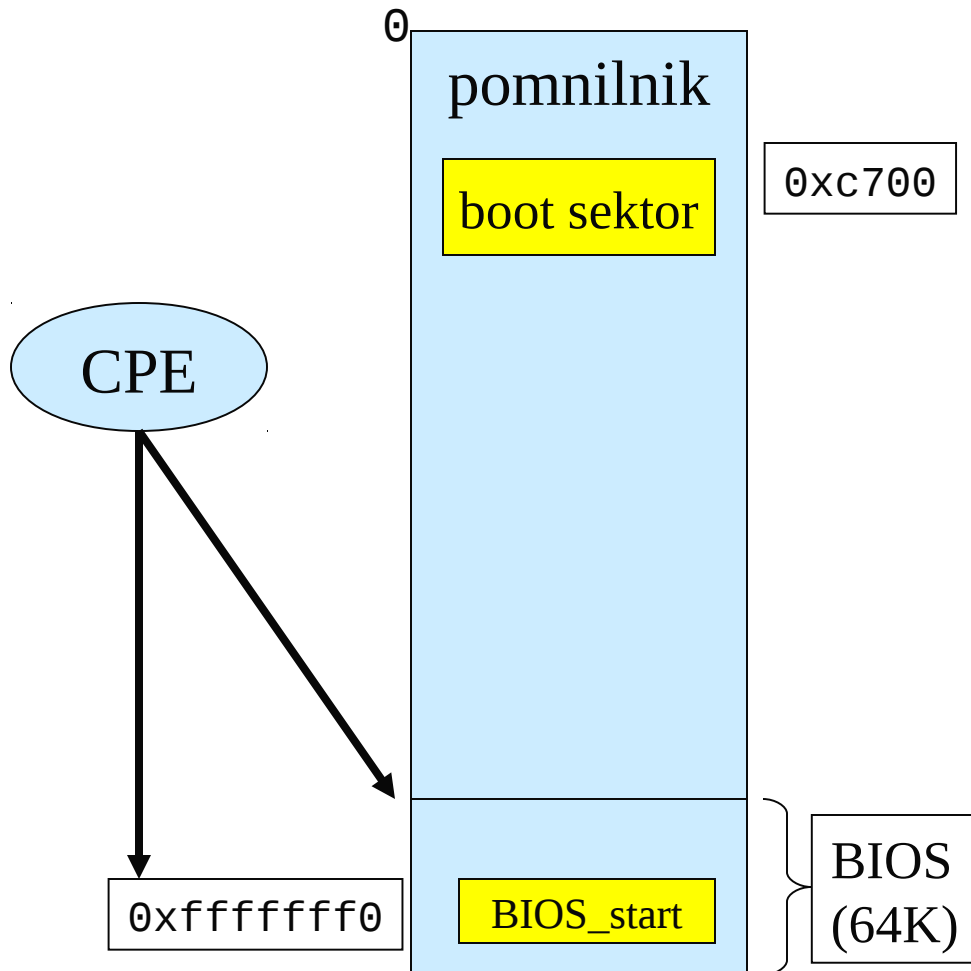
# Bootstrap Loader (Bootloader)

- Program, ki naloži “prvi program”
- Pogosto “večstopenjski”: primarni, sekundarni
- Terja podporo firmware (“hardware bootstrap”)



# Zagon PC

- Intel X86 firmware naloži "boot sector" na 0x7C00 in nanj prenese nadzor v realnem režimu (limita 640K )



1. Power On Self Test (POST)
2. Proženje INT 19h (bootstrap)
3. Izbira zagonske naprave (boot device)
4. Nalaganje zagonskega sektorja (boot sector)
  1. floppy: prvi sektor
  2. trdi disk: MBR (mboot) ali "partition boot block" (pboot)
5. Verifikacija "magic number"
6. Izvajanje boot sektorja (primary bootloader)

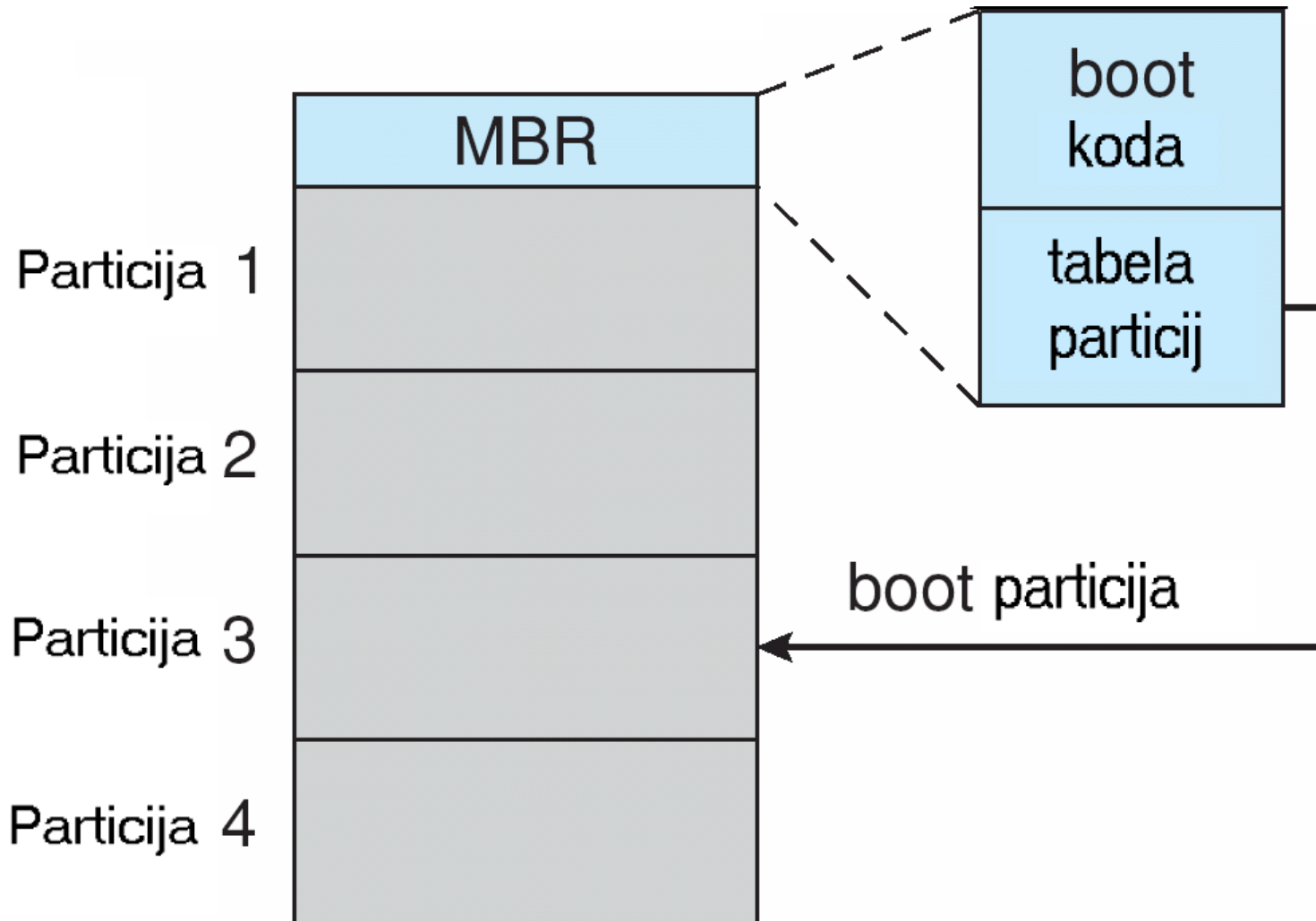


# Naloge ob zagonu (nadaljevanje)

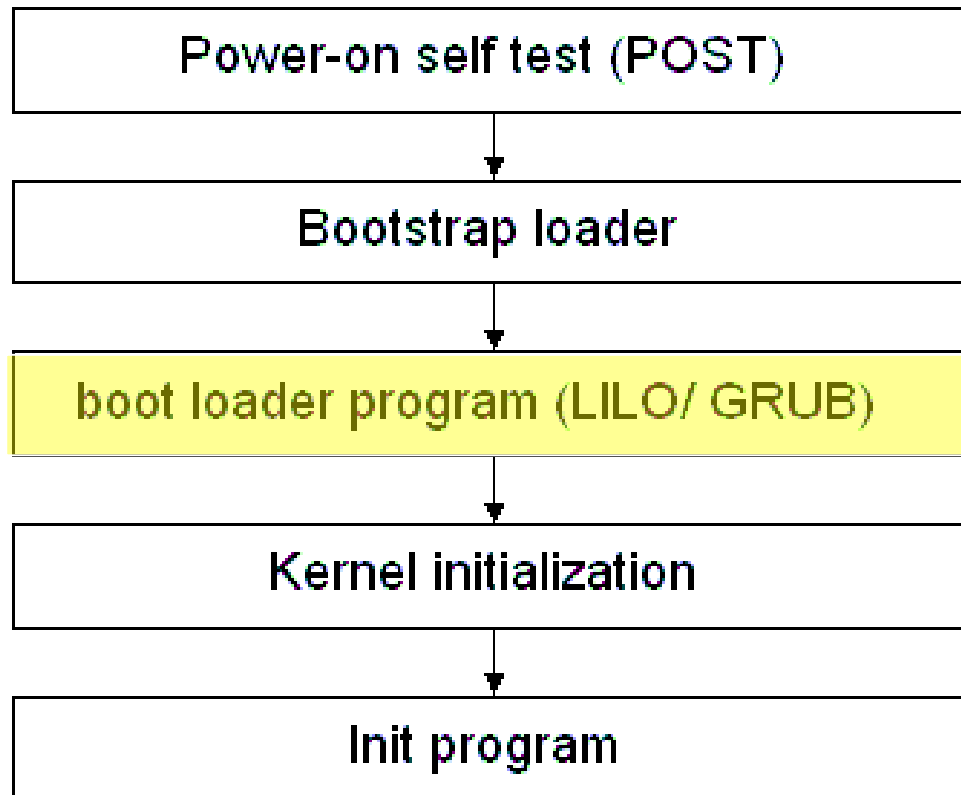
- Ko najde zagonsko napravo, naloži BIOS njen zagonski sektor (boot sector) in ga sproži. V primeru trdega diska je to na MBR ( master boot record ) in pogosto ni specifično za določen operacijski sistem.
- Koda MBR preveri tabelo particij (partition table) in išče v njej aktivno particijo. Če jo najde, naloži koda MBR zagonski sektor (boot sector ) te particije in sproži njegovo izvajanje.
- Zagonski sektor (boot sector) je pogosto specifičen za dani operacijski sistem. V mnogih operacijskih sistemih je glavna naloga kode zagonskega sektorja nalaganje in proženje jedra (kernel), ki nato nadaljuje zagonske postopke (startup).

MBR	Partition table	Particija	Particija	Particija
-----	-----------------	-----------	-----------	-----------

# Zagon z diska pri Windows 2000

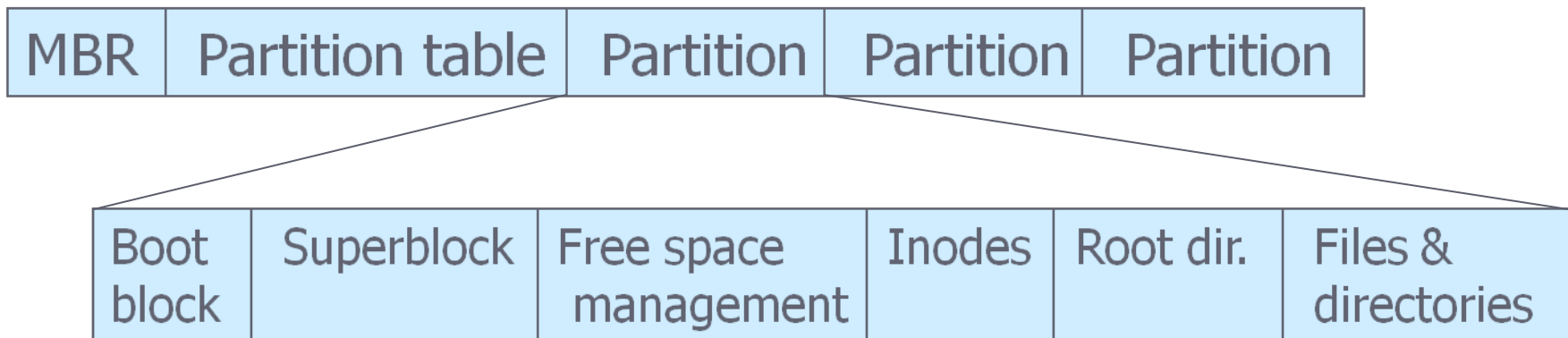


# Sekundarni zagonski nalagalenik



# Sekundarni zagonski nalagalnik

- Če ni aktivne particije ali če izbere particijo (lahko s pomočjo vnosa uporabnika), naloži njen zagonski sektor.
- Primeri sekundarnih zagonskih nalagalnikov:
  - GRUB – GRand Unified Bootloader
  - LILO – LInux LOader
  - NTLDR – NT Loader
- Sekundarnim zagonskim nalagalnikom pravimo tudi zagonski upravniki (boot managers)



# LILO: Linux LOader

## ● Prilagodljiv zagonski upravnik, ki omogoča:

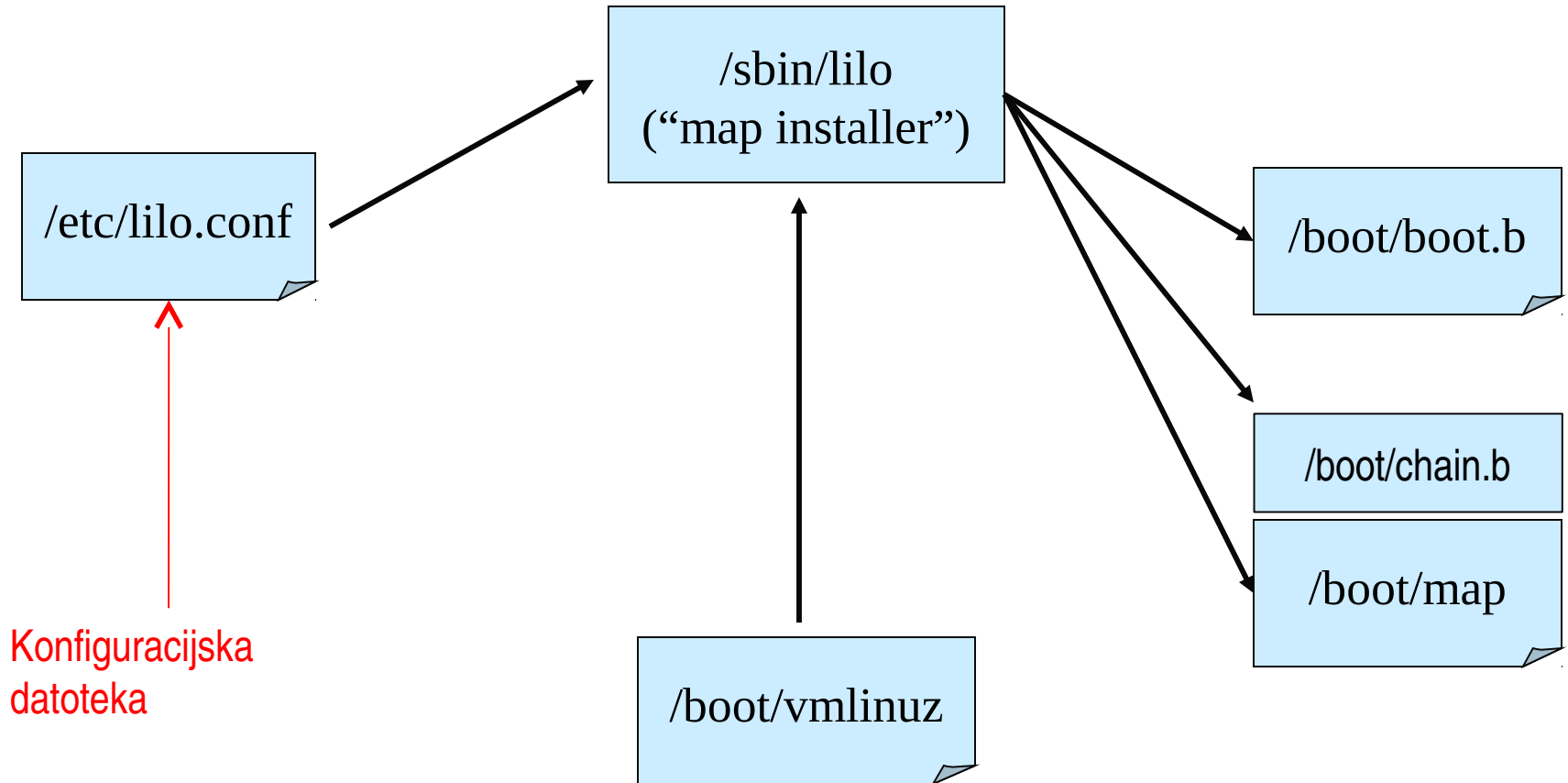
- Izbiro jeder Linux
- Nastavitev časovnih parametrov zagona
- Zagon jeder, ki niso Linux
- Nastavljanje konfiguracij

## ● Značilnosti:

- Leži v MBR ali zagonskem sektorju (boot sector) particije
- Ne ve nič o datotečnem sistemu



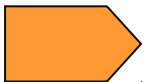
# Komponente LILO



# Primer konfiguracijske datoteke lilo.conf

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
default=linux

image=/boot/vmlinuz-2.2.12-20
    label=linux
    initrd=/boot/initrd-2.2.12-20.img
    read-only
    root=/dev/hda1
```



# Primer konfiguracijske datoteke grub.conf

```
default=0
timeout=10
splashimage=(hd1,2)/grub/splash.xpm.gz
password --md5 $1$opeVt0$Y.br.18LyAasRsGdSKLYlp1
title Red Hat Linux
    password --md5 $1$0peVt0$Y.br.18LyAasRsGdSKLYlp1
    root (hd1,2)
    kernel /vmlinuz-2.4.18-14 ro root=LABEL=/
    initrd /initrd-2.4.18-14.img
title Windows XP
    password --md5 $1$0peVt0$Y.br.18LyAasRsGdSKLYlp1
    rootnoverify (hd0,0)
    chainloader +1
```



# GRUB

- “Lupina za čas zaganjanja (Boot time shell)
- GRUB interaktivni ukazi
- Sproti (on the fly) izvaja novo konfiguracijo
- Dinamično privzeto konfiguriranje
- Lahko zaganjamo druge OS

# Primerjava LILO in GRUB

## LILO

- Tradicionalna
- Če zamenjamo “kernel” ali spremenimo zagonsko konfiguracijo, moramo reinstalirati v “master boot record”

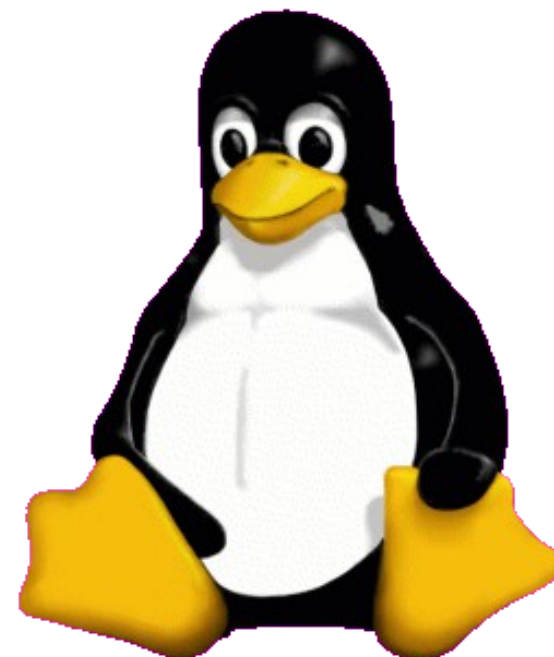
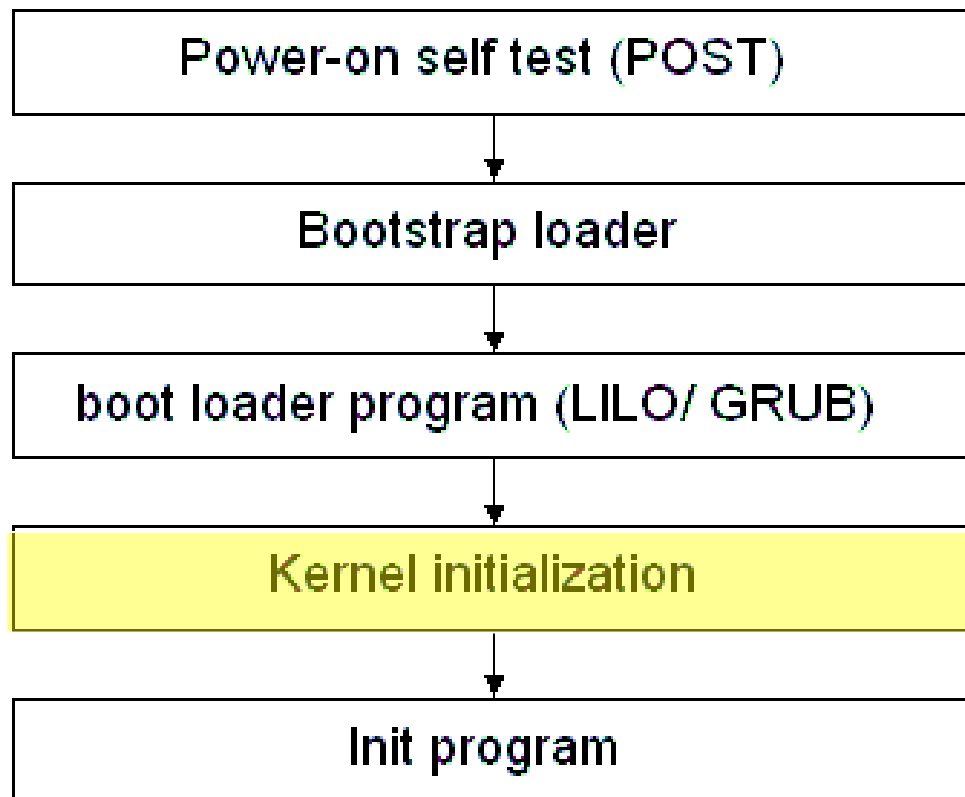
## GRUB

- Nova
- Fleksibilna
- Interaktivni ukazi

# Zagon z diska s pomočjo LILO

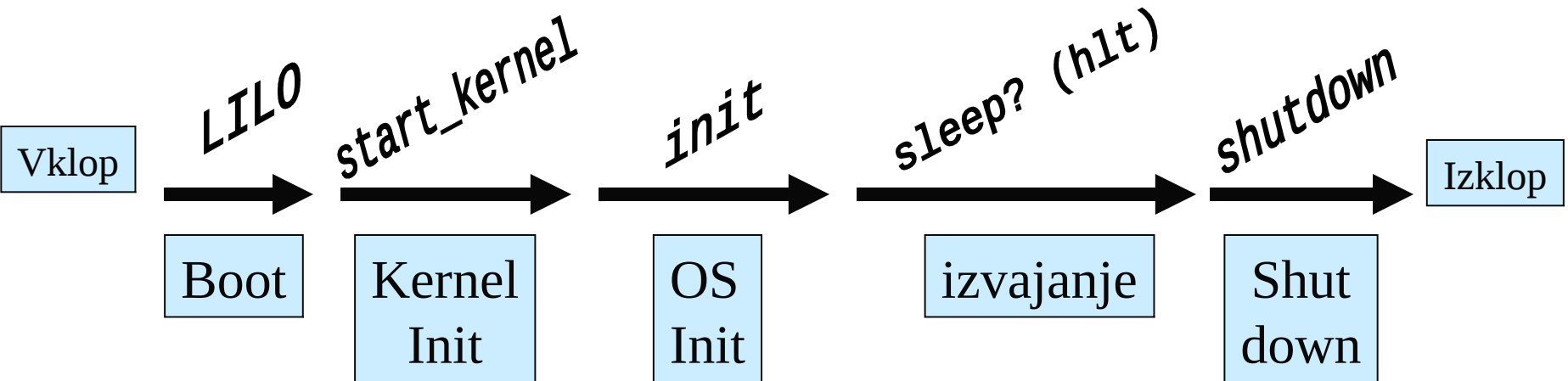
- LILO izpisuje niz “LILO boot:”
  1. BIOS naloži zagonski sektor na 0x7c00  
ter se sam prestavi na 0x9a00
  2. Nastavi sklad  
naloži sekundarni nalagalnik na 0x9b00
  3. Nadzor prevzame sekundarni nalagalnik
  4. naloži “block map” na 0x9d200  
Izpiše komandno vrstico
  5. In čaka na vnos uporabnika ali iztek časa “timeout”

# LINUX: od zagona do zaustavitve

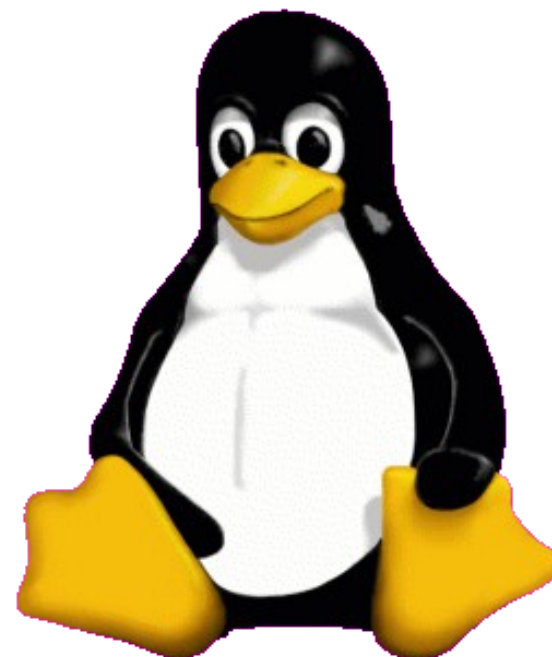
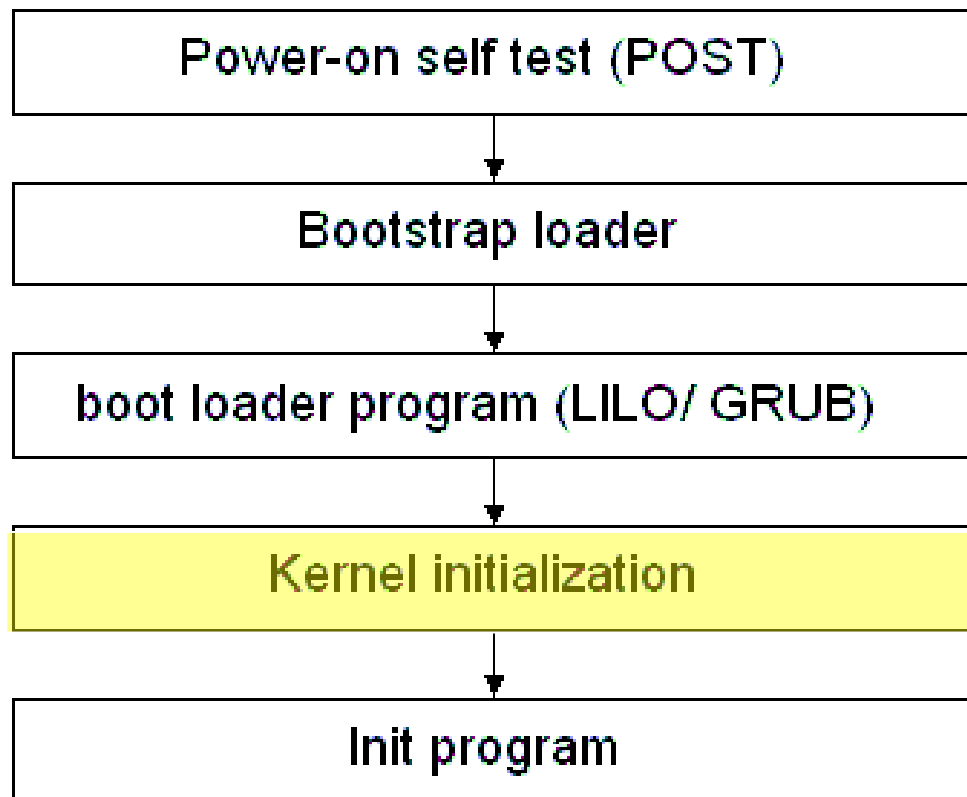


# Življenski cikel sistema: gor in dol

- Zagon (Booting)
- Inicializacija jedra (Kernel Initialization)
- Prvi proces: init
- Zaustavitev sistema (shutdown)



# LINUX: od zagona do zaustavitve



# Inicializacija jedra

- Preveri sistemske naprave
- Identifikacija specifičnih naprav
- Jedro (Kernel)
  - Zagotavlja, da bo strojna oprema delala to, kar hočejo programi

# Inicializacija jedra

## ● Preskus bistvenih naprav

- CPE, konzola, pomnilnik

## ● Preskus strojnih podsistemov

- I/O vodila, omrežni vmesniki, trdi diski, CD-ROM pogoni, disketni pogoni, pomnilne naprave



# Inicializacija jedra

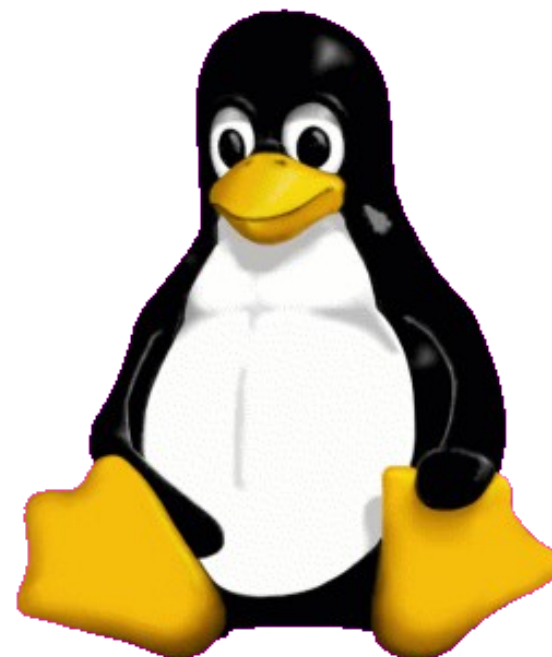
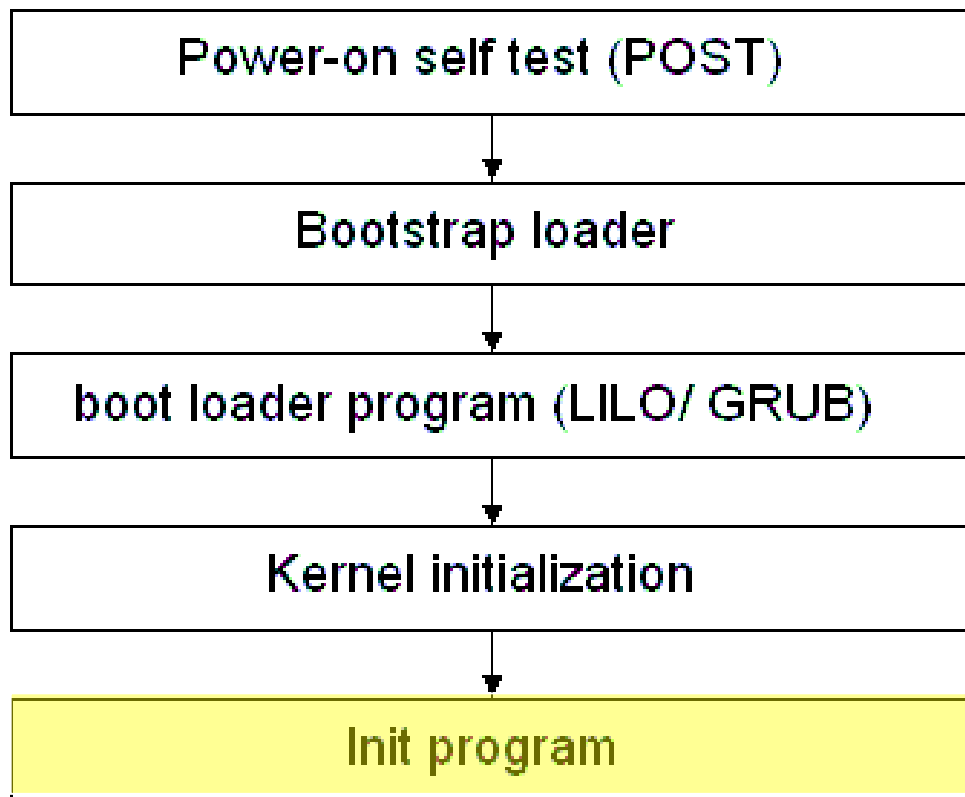
## ● Inicializacija datotečnega sistema

- Logical volume manager subsystem
- RAID
- SCSI naprave
- Particije na trdem disku

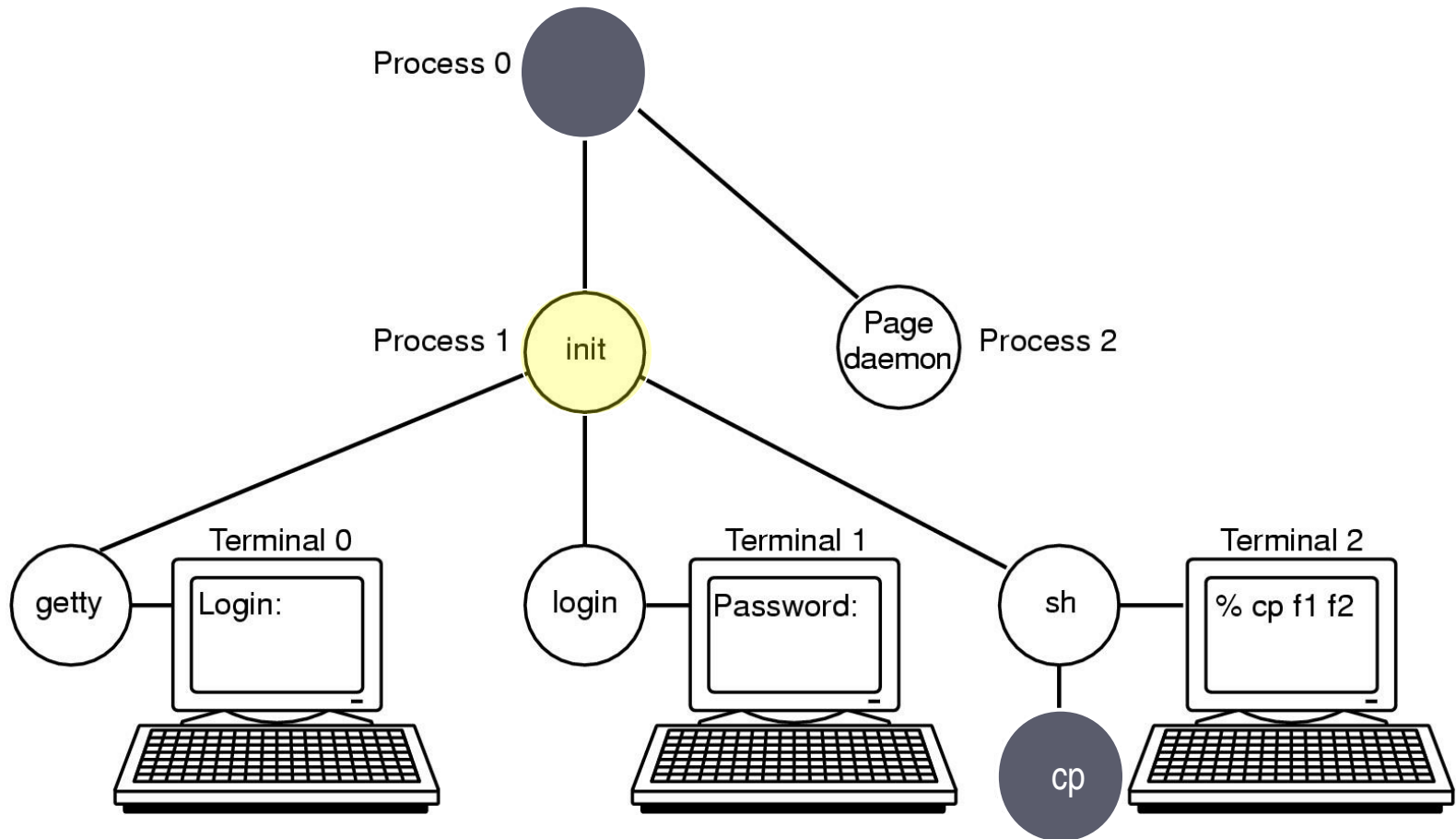
## ● Spreminjanje konfiguracije jedra

- `/usr/src/linux/make menuconfig` or `xconfig`
- `rdev`
- Boot loader parameter

# LINUX: od zagona do zaustavitve



# Zagon UNIX (Linux)




Zaporedje procesov, ki zaživijo pri zagonu nekaterih sistemov

# init()

● `init()` začne življenje kot nit jedra in konča kot proces na uporabniškem nivoju (`/sbin/init`)

## ● `init/main.c:init`

- 
- acquire “the big kernel lock” on a multiprocessor (MP)
  - perform high-level initialization – `do_basic_setup()`
  - free `__init` memory
  - release lock
  - try to exec (in user space) the `init` process
  - panic if unsuccessful

# Init?

- Init je predhodnik vseh procesov (vendar brezposeln) “seje” otroke
- Lokacija: `/sbin/init`
- Teče v uporabniškem načinu (user mode) (do jedra dostopa preko sistemskih klicev)

# Kaj naredi Init ob zagonu?

- Pregleda datoteko `/etc/inittab`
- Preko nje glede na **nivo izvajanja** požene skripte, ki se nahajajo v datotekah v imeniku `/etc/rc.d`
  - servisi za beleženje sistemskih obvestil
  - vzpostavitev mreže
  - mrežni strežniki (splet, pošta...)
  - ...
- Požene procese, ki omogočijo prijavo na sistem na
  - tekstovnih terminalih
  - lahko tudi grafični uporabniški vmesnik

# Nivoji izvajanja

## ● UNIX Run Levels – nivoji izvajanja

- pove na kakšnem nivoju (načinu) je sistem
  - ▶ single user (vzdrževanje) in
  - ▶ multi-user (brez in z omrežnimi servisi)
- parameter pri poganjanju procesa init pove v kakšen način naj se računalnik postavi

## ● Tudi Windows imajo nivoje:

- Multi-user
- Safe Mode
- Safe mode with networking

# Nivoji izvajanja v init

- **0:** Ustavljen sistem (pripravljen na izklop)
- **1:** Enouporabniški režim
- **2:** Večuporabniški režim brez mrežnih datotečnih sistemov
- **3:** Večuporabniški režim z mrežo
- **4:** navadno ni v uporabi
- **5:** Večuporabniški sistem z GUI
- **6:** Reboot režim
- **S,s:** Single user mode (brez /etc/inittab)



# Init

- Katere servise init zaganja in njihov vrstni red glede na nivo izvajanja določa
  - datoteka `/etc/inittab`
  - programi v lupini (skripte) v imenikih `/etc/rc.d`, ki so urejeni po nivojih izvajanja
- Ena od prednosti uporabe skript pri zagonu je, da jih lahko preverjamo in spreminjamo
  - skripte lahko ročno pokličemo z argumenti `stop` in `start` in preverjamo, ali delujejo pravilno
  - tak postopek je priporočljiv, saj lahko odkrijemo napake, ki bi prekinile zagon in povzročile, da bi bil sistem neuporaben.

# Primer iniciacijskega skripta

```
RETVAL=0

umask 077

start() {
    echo -n "Starting system logger: "
    daemon syslogd $SYSLOGD_OPTIONS
    RETVAL=$?
    echo
    echo -n "Starting kernel logger: "
    daemon klogd $KLOGD_OPTIONS
    echo
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/syslog
    return $RETVAL
}

stop() {
    echo -n "Shutting down kernel logger: "
    killproc klogd
    echo
    echo -n "Shutting down system logger: "
    killproc syslogd
    RETVAL=$?
    echo
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/syslog
    return $RETVAL
}
```

# Init

- Lokacija: /sbin/init
- Uporablja funkcije iz knjižnjic v jeziku C
- Preveri in montira datotečni sistem
- Požene demone za beleženje sistemskih obvestil
- Požene procese getty, ki na virtualne terminale izpišejo najavko login
- Omreženje
- Strežniki WEB strani
- Poslušanje miši

# Spreminjanje nivojev izvajanja

- Naslednji ukazi so tipično rezervirani spreminjanje nivojev:
  - UNIX:
    - ▶ Ukazi shutdown, telinit oz. init
  - Windows:
    - ▶ Ukaz shutdown

# Najava uporabnikov (login)

Ko je jedro naloženo, smo še v privilegiranem režimu. Če bi se sedaj lahko prijavi (login) kot uporabnik, bi imeli sistemske privilegije.

Vendar program "login" po preverjanju gesla vrne lupino kot drug proces z nižjimi privilegiji. Jedro procesu "login" zaupa. Če bi ga nadomestili oziroma vanj vdrli, bi lahko dobili administratorsko lupino (root shell) za vsako prijavo.

**Account Login**

Username:

Password:

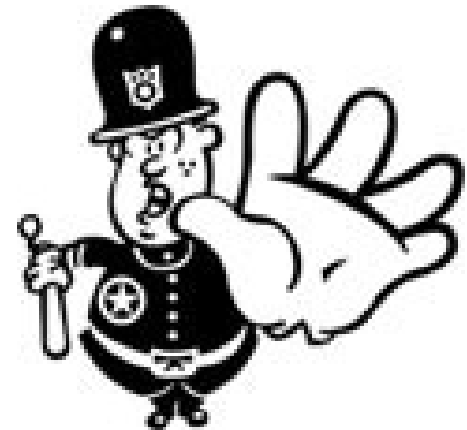
# Zaustavitev (shutdown)

- Zapis “bufferjev”: da ne bi izgubili podatkov in okvarili datotečni sistem, uporabimo `/bin/shutdown`
- shutdown onemogoči logiranje, zahteva od “init”, da pošlje vsem procesom signale SIGTERM in SIGKILL (jih torej konča).



# Zaporedje zaustavitve

- Shutdown [-h/r]
- Blokirano je ponovno prijavljanje
- Vsem procesom pošljemo signal SIGTERM in jim tako povemo, da se sistem ugaša
- Procesi se čisto zaključijo
- Procesu init pošljemo signal, da naj spremeni nivo izvajanja
  - Privzeto: 1, -h flag 0, -r flag 6



# Zaustavljanje

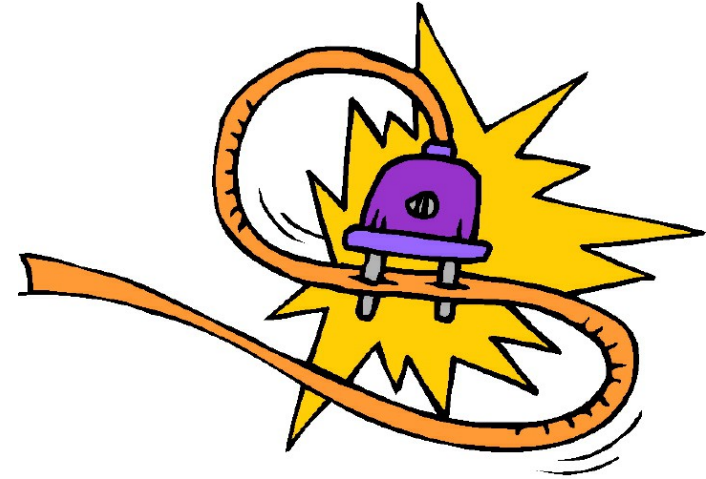
- Odmontiranje particije
- Strukturne informacije datotečnega sistema
- Shutdown -F (fsck)
- “Journaling” datotečnega sistema
  - shranjenje zapisa transakcij o spremembah datotečnega sistema
  - Odgovor pri ponovnem zagonu sistema





# Upravljanje s pomnilnikom

- Halting in the idle process
  - idle process executes hlt on Intel
  - low-power consumption mode
- Suspending the system
  - patches for suspending to disk
- APM: Advance Power Management
  - laptop standard power management
- ACPI: Advanced Configuration and Power Interface
  - new comprehensive standard from Intel-Microsoft
- Power-management is essential for mobile systems



# Dodatek

Ali imam na sistemu tudi sekundarni zagonski nalagalnik, če imam en sam operacijski sistem ?

*Stari sistemi tega morda nimajo. Danes pa so celo Windows nameščeni s privzetim sekundarnim zagonskim nalagalnikom (NTLDR). Tudi Linux je običajno nameščen z LILO ali GRUB kot privzetim zagonskim nalagalnikom.*

*Tako se nam lahko zgodi, da Windows ne moremo zagnati in se pojavi obvestilo "NTLDR missing". To se zgodi, če primarni zagonski nalagalnik ne more prenesti nadzora na NTLDR, ki je morda pokvarjen ali pomotoma zbrisan.*

# Dodatek

Kakšen je vpliv na zagonski sektor (boot sector) in zagonski nalagalnik (boot loader), ko namestimo dva operacijska sistema, na primer Windows in Linux, v dve ločeni particiji ?

*Predpostavimo, da smo najprej namestili Windows. Privzeti zagonski nalagalnik, nameščen v MBR, je NTLDR in vsebuje podatke o aktivni particiji z Windows. Ko na ta sistem namestimo še Linux, namestitev zahteva prekritje sekundarnega zagonskega nalagalnika, ki identificira aktivni particiji tako z Windows kot z Linux. To nam daje možnost izbire operacijskega sistema ob zagonu.*

*Če pa smo najprej namestili Linux in šele nato Windows, bo "Windows Installer" prekril MBR s svojim lastnim zagonskim nalagalnikom, ki ne spozna aktivne particije Linux. To pa je problem, ki ga rešimo s ponovno namestitvijo sekundarnega nalagalnika, ki zna ugotoviti oba operacijska sistema in ponuditi izbiro.*

# O tabeli particij

- When installing an OS on a computer from scratch, here is how the partition table is created.
- The hard disk is denoted as “hda” where hd=hard disk, and the third letter could mean the hard-disk on the system. For e.g. the first hard disk is “hda”, the second is “hdb”.
- When the partitioning is done, “hda0” is the place of MBR. “hda1” is the primary partition. Then a secondary partition may be created which is further subdivided into logical drives. Another OS could be installed on any of these logical drives.
- hda0 – MBR
  - hda1 – Primary Partition e.g. Windows XP
  - hda2 – Secondary Partition
    - hda3 – Logical Drive 1 (FAT32 or NTFS partition)
    - hda4 – Logical Drive 2 (FAT32 or NTFS partition)
    - hda5 – Logical Drive 3 (Swap for Linux Partition)
    - hda6 – Logical Drive 4 (Root for Linux Partition)

The above example is a simple example. Specific cases can be different.