

Linux Boot folyamat – Egyszerusített jegyzet

1. BIOS / UEFI rendszerindítás kezdete

A számítógép bekapcsolásakor a BIOS (regi gépeken) vagy az UEFI (ujabbakon) indul el. Lefuttat egy POST tesztet, ami a memória, CPU és hattertarak állapotát ellenorzi. Ha minden rendben, kiválasztja a boot eszközt (pl. HDD, SSD, USB) és betölti a bootloadert.

2. Bootloader MBR vagy GPT szerint

BIOS rendszeren a bootloader az MBR első 512 bajtjában található. UEFI esetén a bootloader egy .efi fájl, amit az EFI partíció tárol a rendszer. Ez indítja a GRUB-ot vagy más betöltőt.

3. GRUB indítomenu

A GRUB lehetővé teszi, hogy válasszunk több operációs rendszer vagy kernel verzió közül. A `/etc/default/grub` és `grub.cfg` fájlok segítségével konfigurálható.

4. Kernel és initramfs

A Linux kernel betöltődik memóriába, és elindítja az initramfs-t, ami egy ideiglenes fájlrendszer. Ezután eléri a fő (gyöker) fájlrendszert, és elindítja a systemd vagy init folyamatot (PID=1).

5. Systemd vagy Init szolgáltatások kezelése

A PID=1 folyamat (pl. systemd) elindítja a rendszer szolgáltatásait: hálózat, login, grafikus felület stb. A rendszer célállapotait (runlevel/target) is kezeli.

6. Runlevel és programindítás

A rendszer a megfelelő célállapothoz tartozó szkripteket indítja el az `/etc/rc?.d` könyvtárakból.

Miert fontos mindez?

Ha nem indul a VM vagy hiányzik a GRUB menü, tudod, hol keresd a hibát. Megerted, miért a systemd a PID=1, és hogyan kapcsolodnak hozzá a többi folyamatok.

UEFI BIOS utoda

Linux Boot folyamat – Egyszerusített jegyzet

Az UEFI (Unified Extensible Firmware Interface) a modern számítógépek firmware rendszere, amely leváltja a hagyományos BIOS-t. Az UEFI-t ipari konzorcium (UEFI Forum) fejleszti, és támogatja a BIOS-visszakapcsolási modot (CSM), ha szükséges.

UEFI előnyei BIOS-szal szemben

- Támogatja a 2 TB-nál nagyobb lemezeket a GPT partíciós táblával.
- Gyorsabb rendszerindítás az optimalizált hardver inicializáció miatt.
- Secure Boot: csak aláírt rendszerindító kódokat enged.
- Modularis: UEFI alkalmazások (pl. shell, diagnosztika) futhatnak, OS-től függetlenül.

Használat és követelmények

- Modern PC-k többsége már csak UEFI modot támogat (pl. Intel 2020-tól).
- Windows 11 és modern Linux disztrók igénylik az UEFI-t és Secure Boot-ot.
- Virtuális gépek is támogatják: pl. VirtualBox, VMware, QEMU/KVM.

BIOS vs. UEFI Összehasonlító táblázat

Jellemző | BIOS | UEFI

-----|-----|-----

Partíció | MBR (max ~2 TB) | GPT (nagyobb lemezek)

Boot | Egyszerű, lassabb | Gyorsabb, grafikus

Biztonság | Nincs aláírás | Secure Boot támogatás

Bővíthetőség | Korlátozott | UEFI alkalmazások

Kompatibilitás | Régi rendszerek | Modern hardverekkel

Mi az a GRUB?

A GRUB (GNU GRand Unified Bootloader) egy nyílt forráskódú bootloader, amit Unix-szerű rendszereken (pl. Linux) használnak. Lehetővé teszi több operációs rendszer közötti választást az indítás során.

GRUB működési szakaszai

Linux Boot folyamat – Egyszerusített jegyzet

- Stage 1: Az MBR-ben található, elindítja a következő szakaszt.
- Stage 1.5 (core.img): Tartalmaz alap drivereket és fájlrendszer hozzáférést biztosít.
- Stage 2: Betölti a grub.cfg fájlt, megjeleníti a menüt, és elindítja a választott kernelt.

GRUB főbb jellemzői

- Különböző fájlrendszerekből tud olvasni (ext4, FAT, NTFS stb.).
- Interaktív parancssorral rendelkezik hibakereséshez.
- Támogatja a chainload-ot (pl. Windows indítása saját loaderrel).
- Több platformon működik (x86, ARM stb.).

GRUB Legacy vs GRUB 2

- GRUB Legacy: régi verzió, egyszerűbb, kevésbé modularis.
- GRUB 2: modern, modularis rendszer, dinamikus modulokkal, szkriptekkel és lokalizációval.

Konfiguráció és gyakorlat

- A grub.cfg fájl szabályozza a GRUB menüt.
- A /etc/default/grub fájlban beállítható a menüidőzítő (GRUB_TIMEOUT), stílus (GRUB_TIMEOUT_STYLE).
- Módosítás után használd: `sudo update-grub`

GRUB menü billentyűk

- Shift (BIOS rendszeren): lenyomva tartva megjelenik a GRUB menü.
- Esc (UEFI rendszeren): indításkor nyomogatva előhívja a GRUB menüt.
- Nyílak: menüpont választás.
- e: bejegyzés szerkesztése.
- c: GRUB parancssor.
- Esc: vissza a főmenübe szerkesztés/parancssor után.

Folyamatkezelés célja

Megtanulod, hogyan figyeld meg és kezeld a futó folyamatokat Linux alatt. Cél a PID=1

Linux Boot folyamat – Egyszerusített jegyzet

folyamat és gyermekfolyamatainak azonosítása, valamint a háttér/előter-kezelés elsajátítása.

ps pillanatkép a folyamatokról

- `ps -e` vagy `ps -A`: összes folyamat listázása.
- `ps -ef`: részletes folyamatlista.
- `ps aux --sort=-%cpu | head -n 5`: top 5 CPU-használó folyamat.
- `ps -e --forest`: fástrukturában mutatja a folyamatokat.
- Példa PID=1 keresésére:
`ps -e -o pid,ppid,cmd | grep " 1 "`

top valós idejű rendszerfigyelés

- Interaktív, élő frissülő folyamatlista.
- Információk: CPU/memória használat, load average, swap.
- Hasznos billentyűk:
 - P CPU szerint rendez
 - M memória szerint rendez
 - k folyamat kilövés
 - c teljes parancssor megjelenítése
 - q kilepés

htop színes, interaktív top

- Jobban kezelhető, görgethető felület.
- Nyílakkal navigálhatsz, F9 kilövés, F10 kilepés.
- PID, felhasználó, CPU/MEM% jól látható oszlopokkal.

Folyamatkezelés parancsai háttérrel és előterrel

- `sleep 600 &` : parancs háttérben fut.
- `Ctrl+Z` : folyamat szüneteltetése (SIGTSTP).
- `bg` : háttérben való folytatás (SIGCONT).
- `fg` : visszahozás előterbe.

Linux Boot folyamat – Egyszerusített jegyzet

- jobs -l : hatterben futó folyamatok listázása.
- kill %1 vagy kill PID : folyamat megszakítása.

Extra parancsok: disown, nohup, screen

- disown : levalasztja a folyamatot a shell-ről, így az nem áll le kilepeskor.
- nohup : parancs futtatása shell-függetlenül, pl. nohup parancs &
- screen/tmux : külön terminal session, amihez később is visszatérhetsz.

Mi az a subshell?

A subshell egy új bash folyamat, amit a szülő shell indít el. Örökli a környezeti változókat, de a módosításai nem hatnak vissza a szülő shellre.

Subshell létrehozása

- Zárójelek (): minden zárójelbe tett parancs új shellben fut:
(cd /tmp; echo \$PWD)
- Parancs behelyettesítés: \$(...) vagy `...`:
echo "Today is \$(date)"
- bash -c "...": külön bash folyamat indítása explicit módon.

Változók és hatókör

A subshell látja az exportált környezeti változókat, de az ott módosított értékek nem kerülnek vissza a szülő shellbe.

Példa:

x=5

(x=10; echo "subshell x=\$x")

echo "parent x=\$x"

Kimenet:

subshell x=10

parent x=5

Linux Boot folyamat – Egyszerusített jegyzet

`$BASH_SUBSHELL` változó

- Megmutatja a subshell mélységet (0 = fő shell).
- Többszintű zároléssel is az értéke:

```
echo $BASH_SUBSHELL
```

```
(echo $BASH_SUBSHELL; (echo $BASH_SUBSHELL))
```

Gyakorlati példák

- PID/PPID azonosítása:

```
(echo "Subshell $$ PPID $PPID"; sleep 5) &
```

- `ps faux` vagy `ps -ef` kimenetében megfigyelhető a hierarchia.
- `$BASHPID` is megmutatja az aktuális shell PID-jét.

Osszefoglaló Subshell

Fogalom: Zárolékekkel vagy `$()` hozunk létre új bash folyamatot.

Hatkor: Környezeti változókat látja, de a módosításokat nem viszi vissza.

`$BASH_SUBSHELL`: Megmutatja a shell mélységet.

Hasznalat: Kitérő, izolált környezet, párhuzamos futtatás.