

Dr. Mileff Péter

UNIX/LINUX OPERÁCIÓS RENDSZER ÜZEMELTETÉSE

5. ELŐADÁS

Miskolci Egyetem
Általános Informatikai Tanszék

A rendszerindítás...

Általános áttekintés

- ◉ **Rendszerindítás (boot):** az a folyamat, amely a számítógép bekapcsolása után az operációs rendszer betöltését végzi.
 - A "boot" (csizma) név egy olyan képből származik, mely szerint a számítógép ilyenkor felhúzza a csizmáját (**bootstrap**).
- ◉ **Az indítás:**
 - a BIOS különféle tesztek végzésére annak ellenőrzésére, hogy minden rendben van-e
 - power on self test, röviden POST
 - Ezután indul a tényleges rendszerindítás

Általános áttekintés

- ◉ Először egy lemezmeghajtó kerül kiválasztásra
 - az ebben levő lemez legelső szektorát, a boot szektort olvassa be a rendszer.
- ◉ Merevlemezekenél a **master boot record (MBR)** kerül beolvasásra,
 - Mert egy merevlemez több partíciót is tartalmazhat, mindegyiken saját boot szektorral
- ◉ A boot szektor egy kis (egy szektorba elférő) programot tartalmaz (ún. **bootstrap loader**)
 - feladata az aktuális operációs rendszer beolvasása és elindítása.

Általános áttekintés

- Ha merevlemezről boot-olunk: a master boot record-beli kód megvizsgálja a partíciós táblát
 - hogyan azonosítsa az aktív partíciót (amelyik boot-olhatóvá lett téve),
 - beolvassa annak boot szektorát, és elindítja az itteni kódot.
 - A partíció boot szektorában található kód beolvassa a kernelt és elindítja.
- Ennél kicsit több:
 - a boot szektorban lévő kód nem olvashatja egyszerűen sorban a lemez blokkjait,
 - hanem meg kell találni azokat a blokkokat, ahova a fájlrendszer lerakta a kernel képmását.
 - Több megoldás is létezik, de a leggyakoribb a **boot menedzserek** használata (LILO, GRUB).

5

Általános áttekintés

- Miután a Linux kernel bekerült a memóriába, és valóban elindult:
 - a kernel a lemezen tömörítve van, ezért először kicsomagolja önmagát.
 - A kernel képmás eleje egy kis programot tartalmaz e célból.
 - a kernel ellenőrzi, milyen hardver elemek léteznek a gépben, és megpróbálja ezek eszközmeghajtóit megfelelően konfigurálni.
 - A hardverelemek megtalálásáról üzenetet is ad.
 - a kernel csatlakoztatja (mount) a gyökér fájlrendszert.
 - A fájlrendszer típusát automatikus felismeri a rendszer.
 - Ha a gyökér fájlrendszer csatlakoztatása nem sikerül, a kernel "pánikba esik" (**kernel panic**), és megállítja a boot-olás folyamatát.

6

Általános áttekintés

- 4. Ha minden rendben ment, akkor a kernel elindítja az **init** programot (/sbin/init) a háttérben.
 - Értelemszerűen az init processz-azonosítója (PID) mindenképpen 1 lesz.
- 5. Az init ezután több felhasználós üzemmódba kapcsol
 - elindítja a **getty** programot a virtuális konzolokra és a soros vonalakra,
 - így lehetővé teszi a bejelentkezést a felhasználók számára.
 - Az init elindíthat még más programokat is a konfigurációjától függően.

Ezzel a rendszerindítás kész, és a rendszer normálisan fut tovább.

7

Egy boot folyamat

```

Configuring ISA PNP
Setting system time from the hardware clock (localtime).
Using rtcwake to suspend to initialize <dev>rtcwake.
Initializing basic system settings ...
Updating shared libraries
Setting hostname: engine21.mindorch.edu.au
INIT: Entering runlevel: 4
rc.M -> Going to multiuser ...
Starting system logger ... [ OK ]
Initializing advanced hardware [ OK ]
Setting up modules ...
Initializing network
Setting up localhost ... [ OK ]
Setting up inetd ... [ OK ]
Setting up udev ... [ OK ]
Setting up fancy console and GUI
Loading fc-cache ... [ OK ]
rolling init -> Going to runlevel 4
Starting services of runlevel 4
Starting dbus ... [ OK ]
-> rc.X Going to multiuser GUI mode ...
XFree86 Display Manager
Framebuffer: /dev/fb0 is 307200 bytes.
Grabbing 640x480 ...
  
```

8

Az init, mint első folyamat...

9

Az init folyamat

- ◉ Az **init** alapvető egy Unix/Linux rendszer működése szempontjából
 - többnyire nem veszünk róla tudomást.
- ◉ **Helye:** általában /sbin/init.
 - Ha a kernel nem találja az init-et, akkor a rendszerindítás sikertelen.
- ◉ A rendszerindítás folyamatát zárja le néhány adminisztratív feladat elvégzésével
 - pl. a fájlrendszerek ellenőrzése, a /tmp kitakarítása,
 - különféle szolgáltatások elindítása és a getty indítása minden terminálra illetve virtuális konzolra,
 - ahonnan a felhasználók bejelentkezhetnek
 - a legelső processz, az azonosítója mindig 1.

10

Az init folyamat

- ◉ Az init örökbe fogadja az árva processzeket:
 - amikor egy processz gyermekprocesszt indít, és gyermeke előtt meghal:
 - a gyermek azonnal az init gyermekévé válik.
 - Ez többféle technikai okból fontos, és jó tudni róla.
 - könnyebben érthetővé teszi a processzek listáját és fadiagramjait.
- ◉ Az init-nek több változata érhető el:
 - A legtöbb Linux terjesztés a **sysvinit**-et használja,
 - a System V rendszer init-jén alapul.
 - UNIX BSD változatainak más az init programja.
 - különbség a futásszintekben van.

11

Futási szintek...

12

A futási szintek

- A Unix/Linux rendszerek különféle üzemmódokat, **futási szinteket** (*run level*) különböztetnek meg.
- **A futási szintek:** az operációs rendszer állapotai
 - Minden állapot egy-egy külön célt szolgál.
 - A rendszergazda állítja a feladatnak megfelelően.
- A futási szintek viselkedése operációs rendszer függő, beállító állományok határozzák meg.
 - Minden disztribúció saját konfigurációt használ.
 - Dokumentálva van az /etc/inittab -ban

13

Red Hat Linux alapú szintek

- **0 – Leállítás:** ez a futási szint a számítógép leállítását, kikapcsolását végzi.
 - Amelyik számítógép felveszi ezt a futási szintet, az rövid időn belül leáll.
- **1 – Egy felhasználós üzemmód:** ez az üzemmód szervizcélokra, helyreállításra használható.
 - Szokás S betűvel is jelölni.
 - Ha egy számítógép felveszi ezt a futási szintet, csak a rendszergazda számára engedélyezi a rendszer használatát.
- **3 – Több felhasználós üzemmód:** a számítógép normális használat közben használt futási szintje.

14

Red Hat Linux alapú szintek

- **5 – Több felhasználós üzemmód grafikus felülettel:** ezt a futási szintet használva a felhasználók számára grafikus bejelentkezési felületet biztosíthatunk.
- **6 – Újraindítás:** az a számítógép, amely felveszi ezt a futási szintet, rövidesen újraindul.
- Az, hogy rendszerünk éppen milyen futási szintben van a **runlevel** paranccsal kérdezhetjük le.

15

Futási szintek konfigurációja...

16

Konfiguráció

- A futási szintek konfigurációja az úgynevezett **indítótáblázat**ban található.
 - Ennek helye: `/etc/inittab`.
- **Jellemzője:**
 - egy egyszerű szöveges állomány.
 - az `init` processz olvassa be.
 - különlegesen fontos, mivel az `init` által a különféle futási szinteken indítandó szolgáltatások leírását tartalmazza.
 - Ha a fájl sérült, vagy nem létezik, akkor a számítógép csak egyfelhasználós üzemmódban indítható.

17

Inittab

- A `/etc/inittab` sorai négy, kettősponttal határolt mezőre oszlanak:

id:runlevels:action:process
- A mezőket a fájlban lejjebb részletezzük ki.
 - A megjegyzéseket a `#`-val kezdődő sorok jelentik.
- **Jelentések:**
 - **id:** azonosítja a fájlban belüli sort. A getty vonalakra ez mondja meg a terminált, amelyen fut, pontosabban a `/dev/tty` utáni karaktereket az eszközfájl nevében.
 - Más sorokra tartalma lényegtelen (de nem lehet túl hosszú), és egyedinek kell lenni.

18

Inittab

- **runlevels:** A futásszintek, melyekre a sor vonatkozik.
 - A szinteket egyetlen számként kell megadni elválasztójel nélkül.
- **action:** : Mit kell tenni a megadott sorral.
 - Pl. mindig újraindítani (`respawn`), vagy csak egyszer futtatni (`once`).
- **process:** A futtatandó parancs.

Példa:

```
16:6:wait:/etc/init.d/rc 6
```
- A rendszer futása közben a futási szint a **telinit** paranccsal módosítható, a kívánt szint számát paraméterként megadva.
 - Erre csak a rendszergazda jogosult.
 - Váltás 3-as futási szintre:


```
# telinit 3
```

19

Alrendszerek indítása és leállítása

- Nem minden szolgáltatást indít az `init` közvetlenül.
 - azon programokat, amelyek a rendszer működése szempontjából nem elengedhetetlenek.
- **Oka:** hatékonyság, biztonság növelése.
 - Veszélyeztetné a működőképesség szempontjából létfontosságú alapszolgáltatások meglétét
 - a kevésbé fontos szolgáltatások telepítése, törlése, beállítása.
 - Mindig módosítani kellene az `/etc/inittab`-ot.
- A magasabb szintű szolgáltatások (alrendszerek) saját indítóprogramokkal rendelkeznek
 - általában BASH programként megvalósított

20

Alrendszerek indítása és leállítása

- Az init az alrendszerek indítását és leállítását **közvetve** végzi.
- Minden futási szint-váltásnál elindítja a /etc/rc.d/rc programot
 - amely eldönti, hogy mely alrendszereket kell leállítani és elindítani az új futási szinten,
 - és a megfelelő indítóprogramok segítségével el is végzi ezt a munkát.
- Az egyes futási szintekhez tartozó elindítandó „programok” az etc/inittab fájlban megtalálhatók.

21

Példa

```
> cr0:0:wait:/etc/rc.d/rc 0
> cr1:1:wait:/etc/rc.d/rc 1
> cr2:2:wait:/etc/rc.d/rc 2
> cr3:3:wait:/etc/rc.d/rc 3
> cr4:4:wait:/etc/rc.d/rc 4
> cr5:5:wait:/etc/rc.d/rc 5
> cr6:6:wait:/etc/rc.d/rc 6
```

- Az rc 0, rc 1, stb valójában egy-egy jegyzék az etc jegyzékben (tehát pl. /etc/rc 1).
- Ezekben található meg azon alrendszerek indító scriptjei amelyeket az adott futási szinten indítani szeretnénk.
 - Ezt a folyamatot az /etc/rc.d/rc BASH program végzi el.

22

Felhasználók nyilvántartása...

23

Felhasználói azonosítók

- A felhasználói adatbázis: az operációs rendszer által kezelt nyilvántartás.
 - Ahol minden felhasználói azonosítóhoz felhasználói nevet (username) és más, fontos kiegészítő információkat rendelünk.
- A Unix rendszerek:
 - minden felhasználó rendelkezik egy **egyedi azonosító** számmal (uid = user identifier),
 - és **egy vagy több csoport azonosító** számmal (gid = group identifier).
 - A műveletek elvégzése előtt az operációs rendszer ellenőrzi, hogy az adott folyamatnak van-e joga a műveletet végrehajtani.

24

Felhasználói azonosítók

- A felhasználók azonosítása a rendszerben az **UID** szám alapján történik.
- Bejelentkezéskor a rendszer a név alapján kikeresi az **/etc/passwd** állományból a felhasználó azonosítóját.
 - A későbbiekben ezt használja a rendszerben folytatott tevékenységünk során.
- A csoportokba sorolás fontossága:
 - segítségével a felhasználók egy csoportját összefoghatjuk
 - s a rendszer egyes részeihez több jogot biztosíthatunk, mint a többieknek.

25

Felhasználói azonosítók

- Egy felhasználó legalább egy (**elsődleges csoport**), de több csoport tagja is lehet egyszerre.
 - Ebből az egyik csoport szerepe megkülönböztetett.
 - Ez az alapértelmezett, amelyet az állományok létrehozásakor használ a rendszer.
- A felhasználó azonosítója és az alapértelmezett csoport azonosítója az **/etc/passwd** állományban van eltárolva.
 - A többi csoport azonosító az **/etc/group** állomány bejegyzései alapján rendelődik hozzá a felhasználóhoz.

26

Felhasználói azonosítók

- Létezik egy kitüntetett felhasználó, a rendszergazda (super-user, a login neve "**root**").
- **Jellemzői:**
 - Rá nem vonatkoznak a felhasználókra beállított hozzáférési jogok,
 - a rendszer még az állományok tulajdonosnál is bővebb lehetőségeket engedélyez számára.
 - A rendszer annak alapján ismeri fel, hogy az uid-je 0.
 - Jogai gyakorlatilag korlátlanok.

27

Felhasználók létrehozása...

28

Felhasználói létrehozása

- ◉ Új felhasználó létrehozása: elegendő, ha egy rá vonatkozó bejegyzést írunk az `/etc/passwd` állományba.
 - Továbbá létre kell hoznunk neki egy **home** jegyzéket, amelyet a tulajdonába adunk,
 - és bejegyzünk szintén a **passwd** állományba.
- ◉ Léteznek segédprogramok is, amelyek ezeket a műveleteket elvégzik nekünk.
 - Parancssoros programok közül ilyen a mindenhol megtalálható **useradd**.

29

Felhasználói létrehozása

- ◉ Két működési módját különböztetjük meg.
 - 1. Az egyik feladata, hogy új felhasználókat hozzon létre.
 - 2. A másik módja a konfigurációs mód, amikor az alapértelmezett értékeket állítjuk be.
- ◉ A legegyszerűbb, és leggyakrabban használt formája:


```
useradd -c <valódi név> <login név>
```
- ◉ Alapértelmezett esetben:
 - bejegyzí a felhasználót a **passwd** állományba
 - majd létrehoz egy home könyvtárat (-m), amelybe bemásolja az `/etc/skel` könyvtárban található állományokat.

30

Felhasználói létrehozása

- ◉ A jelszót utólag rendszergazdaként adhatunk a **passwd** paranccsal:


```
passwd <login név>
```
- ◉ Nem csak valódi felhasználókat hozhatunk létre, hanem úgy nevezett **rendszer-felhasználókat** is.
 - Ezek jelszó nélküli, blokkolt felhasználók home jegyzék nélkül.
- ◉ Létrehozásuk célja:
 - Azért szükségesek, hogy egyes szolgáltatások ezeknek a virtuális felhasználóknak a nevében fussanak,
 - ezáltal a rendszer a hozzáférési jogosultságait szabályozhassa.

31

Felhasználói létrehozása

- ◉ A rendszer-felhasználók azonosítója általában kisebb.
- ◉ Létrehozás: a **useradd** programmal az **-r** opcióval.
 - Az alapértelmezett paramétereket módosíthatjuk is a **-D** opció használatával:


```
useradd -D <opció és alapértelmezett érték>
```
- ◉ A rendszer az alapértelmezett értékeket az **/etc/default/useradd** állományban tárolja.

32

Felhasználói tiltása

- A felhasználó hozzáférését számos okból letilthatjuk.
- Egyik szokásos eset, amikor csak egy bizonyos időszakra adjuk a hozzáférést.
- Ilyenkor már a felhasználó létrehozásánál
 - a `useradd` parancs `-e` opciójával megadhatjuk a lejáratí dátumot, amikortól nem férhet hozzá a rendszerhez.
- A másik szokásos eset az automatikus letiltásra
 - amikor kötelező érvényűvé tesszük, hogy a felhasználók bizonyos idő elteltével cseréljék a jelszavukat.
 - Ezt a `passwd` parancs `-x` opciójával adhatjuk meg napokban.

33

Köszönöm a figyelmet!

34