

Linux rendszergazda tanfolyam

II. Parancsuralom



Bekapcsolás – boot folyamat

- A számítógép bekapcsolása (Power ON)
- Lefut a BIOS teszt – POST (Power On Self Test)
- Betöltésre kerül az MBR
- Linuxok esetén az MBR-ben lévő betöltő program egy úgynevezett *boot-manager*, ami általában vagy a *Lilo* vagy a *Grub*.
- A *boot manager* lehetővé teszi, hogy a felhasználó menüből válassza ki az indítani kívánt operációs rendszert.
- Betöltésre kerül az operációs rendszer (kernel), elindulnak az alapvető rendszerfolyamatok, szolgáltatások.
- Megjelenik a *login prompt*, a bejelentkezési képernyő innentől lehet bejelentkezni a rendszerbe!



Bejelentkezés, kijelentkezés

- A *login prompt* után közvetlenül megadhatjuk a felhasználói nevünket
- A rendszer ezután automatikusan megkérdezi a felhasználóhoz tartozó jelszót (*Password:*).
- Helyes jelszó megadás után a beállított *shell* felületen el lehet kezdeni dolgozni.
- A *root* prompt-ja (készletlégi jel, ahova a parancsokat írhatja) rendszerint a „#”. A felhasználók prompt-ja pedig a „\$” jel.
- Kijelentkezni a rendszerből a „*logout*” vagy az „*exit*” paranccsal, vagy a Ctrl+D billentyű kombinációval lehet!
- **root felhasználóként csak akkor jelentkezzünk be, ha nagyon muszáj!**



Kikapcsolás

- **A rendszert csak root jogokkal rendelkező felhasználó állíthatja le!!!**
- Ha be van jelentkezve, akkor a következő parancsok használhatók:
 - *halt*
 - *shutdown -h*
 - *init 0*
- Konzolról bejelentkezés nélkül is újraindítható a rendszer az Alt+Ctrl+Del billentyűzet kombinációval.
- A kikapcsolási parancs hatására sorra leállnak a rendszer funkciói, biztonságos állapotba kerülnek a rendszer legfontosabb adatai, s csak ezek után lehet fizikailag is kikapcsolni a gépet.
- **Egy Linuxos rendszert nem szabad a hálózati kapcsolóval egyszerűen kikapcsolni!**



Felhasználó kezelés 1 – a látszat

- Új felhasználó felvitele:

adduser [név] / *useradd [név]*

Az *adduser* hatására:

- Létrehozásra kerül a felhasználó a következő szabad UID-al, a megadott jelszóval és egyéb adatokkal
- Létrehozásra kerül egy azonos nevű csoport a következő szabad GID-al
- Létrehozásra kerül a felhasználó saját könyvtára (home könyvtár) a /home könyvtárban
- A /etc/skel tartalma bemásolódik a felhasználó könyvtárába

- Felhasználó adatainak módosítása:

usermod [név]

- Felhasználó törlése:

deluser [név] / *userdel [név]*

Alapesetben a home könyvtára még megmarad!!!



Felhasználó kezelés 2 – a háttér

- A felhasználói adatok a `/etc/passwd` fájlban vannak, alapesetben a jelszó is itt található, igaz kódolva.
- Az `/etc/passwd` fájl mindenki olvashatja -> biztonsági probléma! Ezért ma már úgynevezett árnyék (*shadow*) jelszó fájlt is alkalmaznak. Ebben az esetben az `/etc/passwd` fájl már nem tartalmaz jelszavakat, a jelszavak az `/etc/shadow` fájlban vannak kódolva elhelyezve. A *shadow* fájl már csak root jogokkal olvasható! Példa: `cp`!
- A felhasználói csoportok információi az `/etc/group` fájlban kerülnek eltárolásra. Ennek is van *shadow* fájlja, az `/etc/gshadow`.
- Ezek az állományok ugyan hagyományos fájlok, kézi szerkesztésük azonban nem ajánlott, ha mégis szükséges, vannak olyan szerkesztők, amelyeket kifejezetten ezeknek a fájloknak a szerkesztésére készítették fel (*vipw*, *vigr*)!
- Ha a fenti fájlok megsérülnek, akkor a rendszerbe lehet, hogy *root*-ként se lehet bejelentkezni!
- `/etc/adduser.conf`, `/etc/deluser.conf`



Alapvető rendszer parancsok

Parancs	Szerepe
<u>man</u>	Manual oldalak minden parancshoz
<u>apropos</u>	Ha funkcióhoz keresünk parancsot
<u>date</u>	Dátum és idő megjelenítése, módosítása
<u>whoami</u>	Ki vagyok én?
<i>w, who</i>	Bejelentkezett felhasználók információi
<i>free</i>	Memória kihasználtság
<i>df</i>	Partíciók információi
<i>ps</i>	Process információk
<i>kill, killall</i>	Adott processek „megölése”
<i>lspci</i>	PCI eszközök listázása
<i>dmesg</i>	A rendszerindulás információi
<i>su</i>	Indít egy shell-t más user és group ID-vel
<i>passwd</i>	Felhasználó jelszavának módosítása



Alapvető fájlkezelő parancsok

Parancs	Szerepe
<i>pwd</i>	Aktuális útvonal megjelenítése
<i>ls</i>	Fájlok listázása
<i>cd</i>	Könyvtár váltás
<i>mkdir</i>	Könyvtár létrehozás
<i>rmdir</i>	Könyvtár törlése
<i>touch</i>	Állományok létrehozása
<i>vi, nano</i>	Állományok szerkesztése
<i>cat</i>	Állományok tartalmának megjelenítése
<i>cp</i>	Állományok másolása
<i>mv</i>	Állományok átnevezése, mozgatása
<i>rm</i>	Állományok törlése
<i>ln</i>	Linkek létrehozása



Fájlkezelés feladatok 1

1. Listázzuk ki a `/etc` könyvtár tartalmát!
2. Hozzuk létre a `home` könyvtárunkban a következő alkönyvtárszerkezetet: példa. Próbáljuk ki az `„ls -R”` parancsot!
3. Töröljük le az előző feladatban létrehozott alkönyvtárszerkezetet!
4. Hozzuk létre a `home` könyvtárunkban egy `„gyumolcsok”` nevű és egy `„ujfajl”` nevű üres állományt!
5. Szerkesszük editor segítségével mindkét fájlt! A `„gyumolcsok”` fájlban soroljunk fel legalább öt gyümölcsöt külön sorokban. Az `„ujfajl”`-ban tetszőleges adatokat írhatunk, de legalább 10 sor legyen benne!
6. Ellenőrizzük mindkét állomány tartalmát a képernyőn!
7. Olvassuk el a `/usr/share/doc/adduser/examples` könyvtárban lévő `„INSTALL”` és `„README”` fájlokat!
8. Készítsünk a `home` könyvtárunkban lévő `„gyumolcsok”` fájlról egy `„gyumi.pelda”` nevű másolatot!



Fájlkezelés feladatok 2

9. Hozzunk létre egy „*masolat*” nevű könyvtárat a home könyvtárunkban, majd másoljuk bele mindhárom állományt!
10. Másoljuk le a home könyvtárunkba a */usr/share/doc/adduser/examples/adduser.conf* fájlt!
11. Nézzük végig a tartalmát, majd nevezzük át „*adduser.pelda*”-ra!
12. Hozzunk létre egy „*pelda*” nevű könyvtárat a home könyvtárunkban és mozgassuk át oda az összes „*pelda*” kiterjesztésű fájlt!
13. Hozzunk létre egy „*pelda_masolat*” nevű könyvtárat a home könyvtárunkban, és másoljuk bele a komplett „*pelda*” könyvtárat teljes tartalmával! (A „*pelda_masolat*”-on belül legyen egy „*pelda*” könyvtár!)
14. Töröljük le a „*pelda_masolat*” nevű könyvtárat!
15. Készítsünk hard és soft linket is az „*ujfajl*”-hoz, „*ujfajl_hard*” és „*ujfajl_soft*” névvel! Ellenőrizzük is a hivatkozások működését!
16. Próbáljunk készíteni a */var/www* könyvtárhoz hard és soft linket, *whard* és *wsoft* névvel! Ellenőrizzük a működését!



A jogok kezelése

- Ismétlés
- Ha egy felhasználó létrehoz vagy lemásol egy fájlt vagy könyvtárat, akkor a létrejött fájlnak vagy könyvtárnak az adott felhasználó lesz a tulajdonosa, valamint a beállított alapértelmezett jogosultságok lesznek beállítva! Az aktuális beállítás az umask paranccsal kérdezhető le, illetve módosítható.
- Tulajdonos módosítása a chown paranccsal történik.
- Csoport módosítása a chgrp paranccsal is történhet.
- Az engedélyek (jogosultságok) megváltoztatása a chmod paranccsal történik.



Feladatok a jogokkal kapcsolatban 1

1. Állítsuk be a home könyvtárunk hozzáférésének jogát egy biztonságos értékre, (rwxr-xr-x) az ajánlott.
2. Állítsuk be a „gyumolcsok” fájl jogait úgy, hogy csak mi tudjuk olvasni és írni, de senki másnak ne legyen semmilyen joga rá! Ellenőrizzük le egy másik konzolról egy másik felhasználóval!
3. Módosítsuk a home könyvtárunk hozzáférését úgy, hogy csak saját magunk férhessünk hozzá! Ellenőrizzük másik felhasználóval!
4. Állítsuk úgy be az engedélyeket, hogy a „pelda” könyvtárban lévő fájlokat mindenki tudja olvasni, de a home könyvtárunkban lévő egyéb fájlokhoz és könyvtárakhoz nem fér hozzá senki! Ellenőrizzük is!
5. Ellenőrizzük a `/bin/bash` fájl jogait és tulajdonosait! Próbáljuk módosítani a jogokat!
6. Szedjük el minden jogot a „gyumolcsok” fájlról, még magunktól is! Ellenőrizzük a hatását, próbáljuk meg olvasni és írni! Majd állítsuk vissza a jogokat!



Feladatok a jogokkal kapcsolatban 2

7. Vegyünk fel 3 felhasználót a rendszerbe *teszt1*, *teszt2* és *teszt3* azonosítóval! Majd jelentkezünk is be a nevükben ellenőrzésképpen, és hozzunk létre tetszőleges tartalommal egy „*adatok*” nevű fájlt a home könyvtárunkban!
8. Módosítsuk a home könyvtárak jogait úgy, hogy az egyes felhasználók egymás állományaihoz ne férjenek hozzá! Ellenőrizzük is!
9. Hozzunk létre a */home* könyvtárban egy „*kozos_r*” nevű könyvtárat, helyezzünk el benne legalább egy fájlt. Állítsuk be a jogokat úgy, hogy minden felhasználó csak olvasni tudja a fájlt! Se új fájlt se új könyvtárat ne tudjon létrehozni! Ellenőrizzük is!
10. Hozzunk létre a */home* könyvtárban egy „*kozos_w*” nevű könyvtárat, helyezzünk el benne legalább egy fájlt. Állítsuk be a jogokat úgy, hogy minden felhasználó olvasni és írni is tudja a fájlt, de új fájlt és könyvtárat most se hozhasson létre! Ellenőrizzük is!
11. Módosítsuk a „*kozos_w*” jogait úgy, hogy mindenki tudjon új fájlt és könyvtárat is létrehozni benne! Ellenőrizzük!
12. Vegyük el mindenkitől az írási jogot a „*passwd*” fájlban, majd próbáljuk meg törölni!



Feladatok a jogokkal kapcsolatban 3

13. Hozzunk létre a */home-n* belül egy „*kozos_g*” könyvtárat állítsuk be úgy a csoportokat és jogokat, hogy csak a *teszt2* és *teszt3* felhasználók tudjanak belépni a könyvtárba! (Célszerű ehhez egy új csoportot létrehozni - *tesztg*!) Ellenőrizzük!
14. Hozzunk létre a „*kozos_g*”-n belül egy „*elso*” nevű fájlt, és tegyük írhatóvá a csoport tagjai számára! Ellenőrizzük mindenkivel!
15. Hozzunk létre a „*kozos_g*”-n belül egy „*munka*” nevű könyvtárat, és adjunk teljes hozzáférést a csoport tagjainak! Teszteljük! Hozzunk létre az egyik felhasználóval egy fájlt, majd próbáljuk módosítani a másikkal!
16. A probléma ott van, hogy az újonnan létrehozott fájlnak a tulajdonosa és a csoportja is a létrehozó felhasználó. Korigáljuk, majd ellenőrizzük másik felhasználóval! Vigyázzunk arra is, hogy alapesetben a csoport is csak olvasási jogot kap, ezért azt is módosítani kell!



Feladatok a jogokkal kapcsolatban 4

17. Az előző feladat napi használat esetén nem megkönnyíti, hanem megnehezíti a munkát. Ezért egyszerűbb megoldás is van, mégpedig az „s” bit, amit tulajdonosra (*suid bit*) és csoportra (*sgid bit*) is be lehet állítani. Állítsuk be a „munka” könyvtáron az „sgid”-et, valamint módosítsuk 002-re az *umask*-ot. Most teszteljünk! Hozzunk létre az egyik felhasználóval egy fájlt és módosítsuk a másikkal és vizsgáljuk meg a jogosultságokat!
18. Próbáljuk ki az előzőeket könyvtárak esetén is!
19. Módosítsuk úgy a jogosultságokat, hogy a „munka” könyvtár, és benne minden létrehozott fájl és könyvtár a *teszt2* felhasználó tulajdonában legyen! (*chmod* parancs -R paraméterrel!) Ellenőrizzük is minden felhasználóval, hogy ugyanúgy használható-e minden fájl és könyvtár!
20. Módosítsuk a *teszt1* felhasználó *shell* felületét */bin/false*-ra, majd próbáljunk meg belépni vele!



Fájlkezelés egyszerűbben

Left	File	Command	Options	Right		
<- /bin				< ~		
	Name	Size	MTime	Name	Size	MTime
	/..	4096	Aug 16 2004	/..	4096	May 6 2005
	*arch	2684	Dec 24 2002	/k1	4096	Mar 4 14:58
	*bash	511400	Apr 8 2002	also	0	May 6 2005
	*cat	9308	Mar 7 2002	harmadik	60	Mar 4 14:17
	*chgrp	15976	Mar 18 2002			
	*chmod	15816	Mar 18 2002			
	*chown	17896	Mar 18 2002			
	*cp	34664	Mar 18 2002			
	*cpio	49760	Jan 31 2005			
	*date	25820	Jul 26 2001			
	*dd	27336	Mar 18 2002			
	*df	25576	Mar 18 2002			
	*dir	43784	Mar 18 2002			
	*dmesg	4032	Dec 24 2002			
	*dnsdomainname	9784	Jul 8 2001			
	*echo	10844	Jul 26 2001			
	*ed	43292	Nov 27 2000			
/..				also		

teszt@debian:~\$

1 Help 2 Menu 3 View 4 Edit 5 Copy 6 RenMov 7 Mkdir 8 Delete 9 PullDn 10 Quit

A *Midnight Commander*-el (*mc*) mindaz elvégezhető, amiről az előzőekben szó volt, de azokon felül rengeteg hasznos további funkcióval is rendelkezik. Az *mc* egy különálló program, amit külön fel kell telepíteni!



További hasznos parancsok 1

Parancs	Szerepe
<i>more</i>	Nagyméretű fájlok oldalakra tördelése
<i>less</i>	Mint a more parancs, de használhatóbb
<i>tac</i>	A bemenet sorait fordított sorrendben írja ki.
<i>grep</i>	Mintaillesztésre használható szűrő parancs
<i>wc</i>	Sorok, szavak, karakterek számát adja meg
<i>cut</i>	Megadott részt vág ki egy sorból.
<i>finger</i>	Információk a felhasználókról
<i>find</i>	Keresés mindennek felett (find / -name at)
<i>sort</i>	Rendezett megjelenítés
<i>uniq</i>	Rendezés után az egyedi sorokat adja
<i>alias</i>	Másodneveket rendelhetünk parancsokhoz
<i>whereis</i>	Parancsok helyének meghatározása
<i>file</i>	Fájlok típusát próbálja meghatározni
<i>du</i>	Összegzi a lemezfoglalást
<i>awk</i>	Nagy tudású szövegfeldolgozó program



További hasznos parancsok 2

Parancs	Szerepe
<i>tee</i>	Csövek elágaztatása
<i>head</i>	A fájlok első x sorát jeleníti meg
<i>tail</i>	A fájlok utolsó x sorát jeleníti meg
<i>od</i>	Állománylistázás oktális formában
<i>dd</i>	Állománylistázás konverziókkal
<i>tr</i>	Karakterkonverziós program
<i>comm</i>	Állományok összehasonlítása
<i>cmp</i>	Állományok összehasonlítása, akár binárisokat is
<i>diff</i>	Állományok összehasonlítása
<i>read</i>	standard inputról olvas be egy sort, és a paraméterében felsorolt változóknak adja a kapott értéket
<i>line</i>	sorvégéig veszi a standard inputról jövő karaktereket, és visszaírja őket a standard kimenetre
<i>test</i>	Objektum tulajdonság ellenőrzése
<i>expr</i>	Aritmetikai és logikai kifejezés kiértékelés
<i>tar</i>	Tömörítő program



További lehetőségek

- Átirányítás, „>” jellel a kimenetet, „<” jellel a bemenetet lehet átirányítani. Ne felejtsük, hogy a programok számára a bemenet is fájl és a kimenet is! (Pl: „*cat gyumolcsok > gyumolcs*”, hatására egy új fájl készül „*gyumolcs*” névvel)

- Szabványos be- és kimenetek:

Név	Azonosító	ID
Szabványos bemenet	STDIN	0
Szabványos kimenet	STDOUT	1
Szabványos hibakimenet	STDERR	2

„*rmdir 2>&1 >log*”
hatására a hibakimenet
átirányításra kerül a
kimenetre, majd minden
egy „log” fájlba kerül!

- Hozzáfűzés, a „>>” jellel egy már létező fájlhoz fűzi hozzá a kimenetet!
- Láncolás, a „|” (AltGr+W) jellel egymás után tetszőleges parancsot lehet összefűzni. Az egyik parancs kimenete, a másik parancs bemenete lesz! (Pl.: „*cat gyumolcsok | grep alma*” hatására csak azok a sorok jelennek meg, amelyekben szerepel az „*alma*” kifejezés!)



Attribútumok

- Linux esetén az attribútumoknak nincs igazán nagy jelentősége, alig használt lehetőségről van szó. Az attribútumok listázása az *lsattr*, módosítása a *chattr* paranccsal történhet (+ hozzáad, - elvesz, = csak az lesz beállítva).
- Néhány fontosabb attribútum:

Jele	Leírás
a	Nem frissíti a fájlok utolsó módosításának időpontját
c	Tömörítés bekapcsolása
d	Nem készül biztonsági másolat a dump paranccsal
s	Törlés esetén a fájl tartalma fizikailag nullázódik
S	A változás azonnal kiíródik a lemezre (nincs cache)
u	Törlés után egy paranccsal visszaállítható lenne !!!

- ~~Rendszerünk nem lehet törölni, állítni, egyáltalán módosítani a fájljára az „i” bitet, majd próbáljuk meg az adott fájl tulajdonosaként módosítani vagy törölni!~~

