

# *Linux Szerver Alapok*

Órai jegyzet  
Változat: 2009-03-31



# Linux Szerver Alapok

- Rigó Ernő
    - E-mail: [rigo@sztaki.hu](mailto:rigo@sztaki.hu)
    - Tel: +3612796222 (munkaidőben)
  - Tantárgyi oldal: <http://gdf.tricon.hu/linux>
  - Segédanyagok:
    - órai jegyzet (kivetített diák)
    - parancssori gyakorlófelület
    - óra végi teszt felület
    - hallgatói Linux operációs rendszer környezet
- 
-

# Licensz



Ezt a dokumentumot szerzői jog védi a  
Creative Commons by-nc-sa licenc szerint!

<http://creativecommons.org/licenses/by-nc-sa/2.0/deed.hu>



# Mottó

*Mondd el és elfelejtem; Mutasd meg és megjegyzem;  
Engedd, hogy csináljam és megértem.*

*/Kung Fu-Ce/*

---

---

# Tematika

- Történeti áttekintés, alapfogalmak
  - Alapvető shell használat, általános parancsok
  - Felhasználók, jogosultságok
  - Shell scriptek
  - Linux kernel
  - Linux grafikus környezet (X)
  - Irodai alkalmazások
- 
-

# *Az órák tervezett menete*

- Elméleti és gyakorlati blokkok váltakoznak
- Hangsúly a gyakorlati ismereteken
  - Hallgatói Linux munkakörnyezet
- Óra második felében, végén, önálló munka (teszt)
  - regisztrációt igényel!
  - <http://gdf.tricon.hu/tcexam>
- Utolsó gyakorlaton megajánlott jegy szerzés
  - gyakorlatok teszt-eredményei beszámítanak!

# *Első alkalom*

## Első alkalom

1.1 Történeti áttekintés

1.2 Ismerkedés a rendszerrel

1.3 Alapvető navigációs parancsok, tömörítés



# *UNIX Történeti áttekintés 1.*

- Bővebben: <http://en.wikipedia.org/wiki/Unix>
  - 60'as évek: Multics (MIT, AT&T Bell labs, GE)
    - Single Level Store (vö: mmap, vfs)
    - Dinamikus Linker (vö: dll, so)
    - On-line Rekonfiguráció (vö: multiprocessing)
    - Biztonsági szintek („gyűrűk”) fogalma
    - Hierarchikus fájlrendszer
- 
-

# UNIX Történeti áttekintés 2.

- 70'es évek: „nyílt” UNIX (AT&T Bell labs)
  - Brian Kernighan
    - UNIX keresztapa, „Hello World”, AWK, K&R C, m4
  - Ken Thompson
    - B programnyelv, reguláris kifejezések, UTF-8, ma: Google
  - Dennis Ritchie
    - C programnyelv (ASM → C UNIX port)
  - A Bell Labs teljes, forráskóddal kiegészített licenszket ad az egyetemeknek és a kormánynak
  - UNIX v5-6: forkok nagy tömege jelenik meg
    - PWB/UNIX – az első kereskedelmi változat

# *UNIX Történeti áttekintés 3.*

- 80'as évek: kereskedelmi és nyílt UNIX
    - BSD (Berkeley Software Distribution) – nyílt irány
      - C shell
      - Job control
      - TCP/IP támogatás, Socket API
        - Windows korai verzióiban is
    - System III (Bell Labs) – üzleti irány
      - Fájlok zárolása
      - Rendszer adminisztráció
      - STREAMS (a Socket API konkurensa)
      - IPC (processzek közti kommunikáció)
    - Trusted UNIX verziók (katonai célok)
- 
-

# UNIX Történeti áttekintés 4.

- 90'es évek: UNIX „háború”
    - Open Software Foundation (OSF)
      - SystemV+BSD: OSF/1, később: Digital Unix
      - 386BSD
    - UNIX International (UI)
      - Bell Labs SystemV R4 → Novell
      - UnixWare (Windows NT konkurencia is)
      - '93: UI → X/Open Consortium (Novell döntés)
    - '96: X/Open + OSF → Open Group
      - Single UNIX Specification (SUS), POSIX
    - '97: Apple: NeXT (BSD alapú) → Mac OS X (Darwin)
- 
-

# *UNIX Történeti áttekintés 5.*

- 2000 óta: Nyílt forráskód
  - '03: SCO szerint övé a UNIX, elkezd jogdíjat gyűjteni
  - '03: SCO vs. Novell per a UNIX jogairól
  - '05: Sun: OpenSolaris (System V R4 alapú)
  - '07: Novell: „We don't believe there is Unix in Linux”

# Linux Történeti áttekintés

- „UNIX-like” operációs rendszer
    - '84: Richard Stallman: GNU project
    - '85: Free Software Foundation
    - '85: General Public License (GPL - „copyleft”)
    - '87: Andrew S. Tannenbaum – MINIX (üzleti, oktatási)
    - '91: Linus Torvalds – (GNU/?)Linux („GPL MINIX”)
    - '92: Tannenbaum – Torvalds vita
      - Nem UNIX-származék
      - Nem POSIX-konform
      - Monolitikus kernelt használ
    - 2000 óta: egyre elterjedtebb
      - beágyazott rendszerek
      - desktop
      - szerverek
- 
-

# *Linux alapfogalmak*

- Linux (kernel)
  - Linux (operációs rendszer)
  - Disztribúció
  - Linux Standard Base, LSB
    - (POSIX-szerű és/vagy bővebb)
    - Standard függvénykönyvtárak, API-k
    - Parancsok és alkalmazások
    - Fájrendszer szerkezete
    - Futási szintek (runlevel)
    - Nyomtató alrendszer, spooler, filter
    - Grafikus alkalmazási felületek (X Window System)
    - Csomagkezelés (?)
- 
-

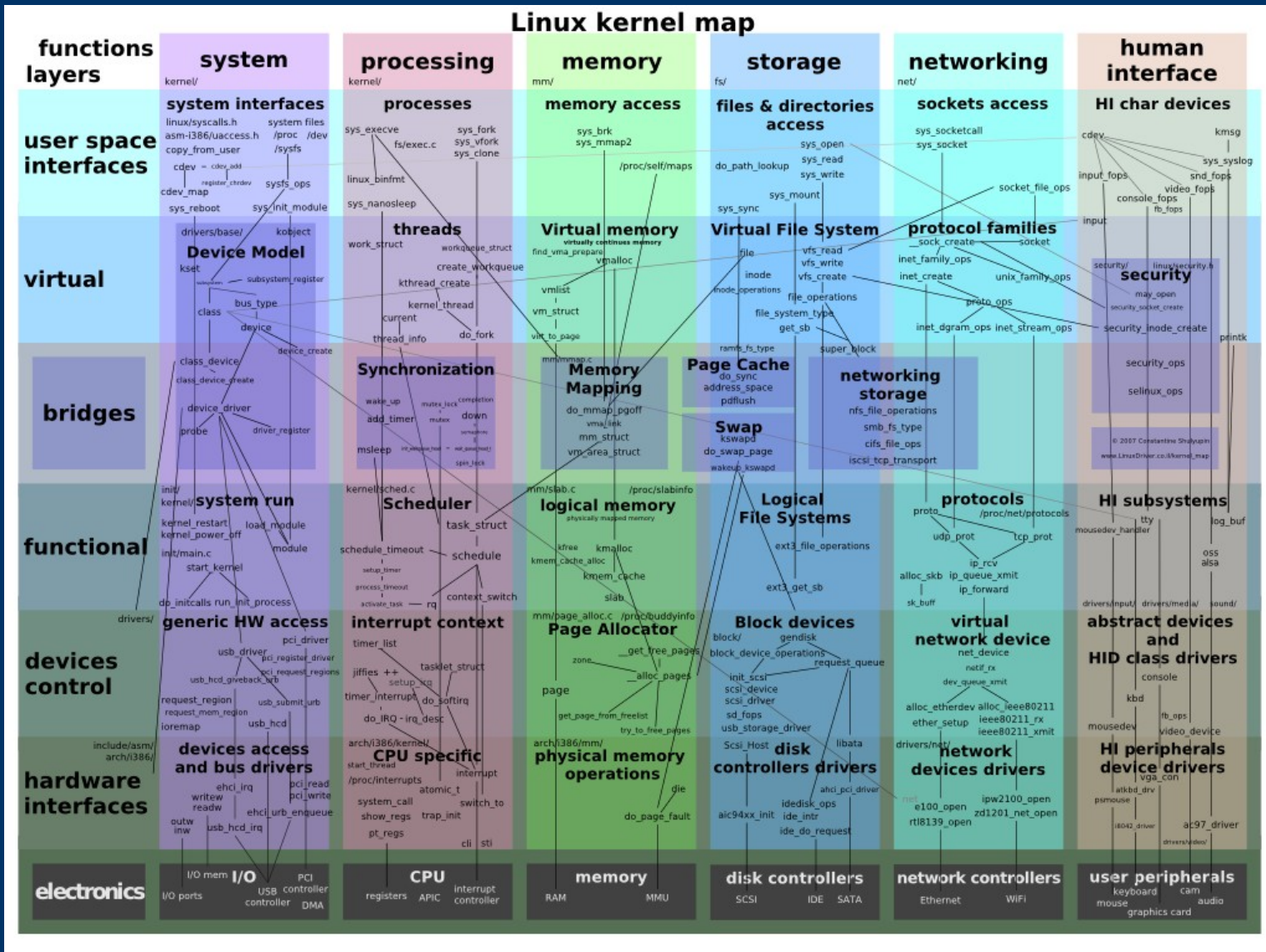
# Linux kernel történelem

- Fejlődése:
    - '91 – 0.01 – 10 239 programsor – MINIX függő
    - '94 – 1.0.0 – 176 250 programsor
    - '99 – 2.2.0 – 1 800 847 programsor
    - '01 – 2.4.0 – 3 377 902 programsor
    - '03 – 2.6.0 – 5 929 913 programsor
    - '08 – 2.6.28 – 10 195 402 programsor
    - EU kutatás szerint a 2.6.8 munkaértéke 882 millió euro
  - Verziószámozás A.B.C[.D]
    - A: kernel verzió (nagyon ritkán változik)
    - B: fő verzió (régebben: páros: stabil, páratlan: instabil)
    - C: főbb javítás és fejlesztés verziója
    - D: kisebb javítás, gyors hibajavítás verziója
- 
-

# Linux kernel felépítése

- Futás közben
    - Monolitikus bináris állomány
      - Több rétegű, összetett (lásd következő ábra!)
      - Egy kódtér („kernel space”) az összes funkciónak
    - Loadable Kernel Module (LKM) támogatás
      - GPL sértési lehetőségek?
    - A rendszer két szintje: Kernel Space vs. User Space
  - Fordítás esetén
    - C és ASM forráskód (hierarchikus szerkezet)
    - Kernel Makefile (fordítási instrukciók)
    - Kernel konfiguráció (hierarchikus)
    - C header fájlok az utólagos LKM fordításhoz
- 
-

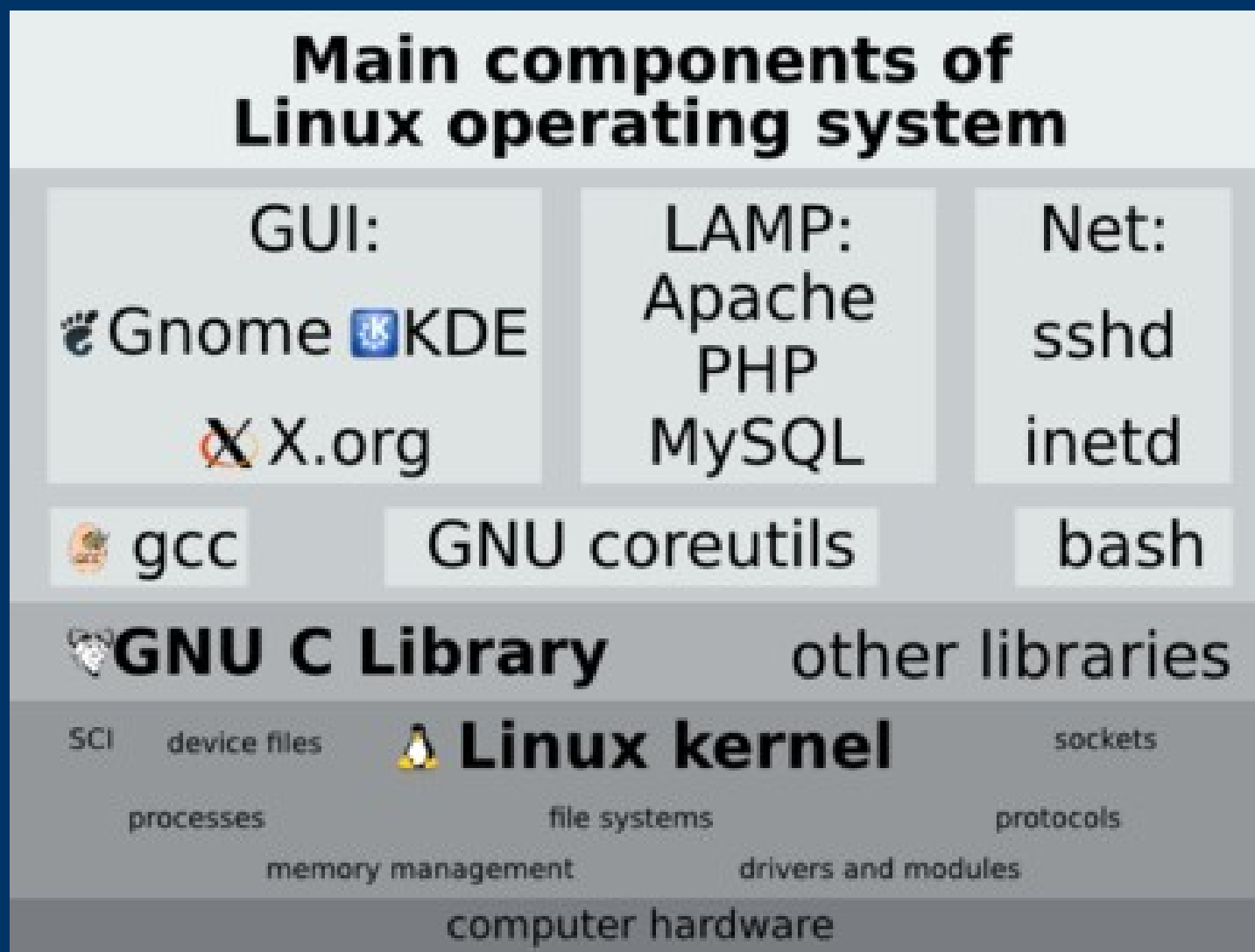
# Linux kernel térkép



# *Linux operációs rendszer*

- Kezdetben: főként GNU szoftverek
    - GCC (GNU C fordító)
    - GLIBC (GNU C függvénykönyvtárak)
    - GNU Core Utils (alap parancsok)
  - Napjainkban: GNU alapok, plusz hobbi és üzlet
    - Mozilla alapítvány termékek
      - Firefox, Thunderbird, stb...
    - Sun Microsystems
      - Java, Netbeans, VirtualBox, stb...
    - Openoffice.org
    - X.org
    - stb...
- 
-

# Linux operációs rendszer térkép



# Linux disztribúciók

- Ad-hoc definíció: „Alkalmazások többé-kevésbé integrált gyűjteménye, Linux kernel alapon”
  - Különböző „szerkesztők”
    - Üzleti: Fedora (*Red Hat*), openSUSE (Novell), Ubuntu (Canonical), Mandriva, stb...
    - Közösségi: *Debian*, Gentoo, *Slackware*, stb...
  - Különböző célokra
    - szerver, desktop, beágyazott, hálózati eszköz, bootolható DVD, biztonsági eszköztár, oktatóanyag, stb...
  - Különböző minőségben...
- 
-

# *Disztribúciók főbb jellemzői*

- Az alábbiak többé-kevésbé (vagy egyáltalán nem) érvényesek minden disztribúcióra:
    - Előre fordított és konfigurált kernel és alkalmazások
    - Rendszeres biztonsági és egyéb frissítések, update
    - Csomag- és függőségkezelés (.deb, .rpm, .tar.gz)
    - Szoftver adatbázis
    - Beépített (grafikus) konfigurációs eszközök
    - Integrált telepítő és együttműködő alkalmazások
    - Támogatás szoftver és tanácsadási szempontból is
    - LSB kompatibilitás
    - Saját téma, „branding”
    - Különböző felhasználási célra minta-konfigurációk
    - Tudásbázis, fórumok, aktív közösség
- 
-



# LSB

- Alap
    - ISO POSIX (legutóbbi „stabil” változat)
  - Célok
    - Bináris kompatibilitás a disztribúciók közt
    - Fejlesztők munkájának összehangolása
  - Eszközök
    - Hivatalos „LSB kompatibilis” tanúsítvány
    - Automatizált teszt eszközök
    - Visszafele kompatibilitás (3 fő verzió, de min. 6 év)
  - Verziók
    - 2001: 1.0, 2004: 2.0, 2005: 3.0
    - 2008: 3.2 (ISO/IEC 23360 DRAFT)
- 
-

# *LSB modulok*

- Core (generic, architectural)
- C++ (generic, architectural)
- Desktop (generic, architectural)
- Interpreted Languages (generic)
- Printing (generic)
- ... plusz trial modulok és egyéb bővítmények



# *LSB generic core*

- I. Introduction of Elements (scope, stb...)
  - II. Executable and Linking Format (ELF)
  - III. Base Libraries
    - libc, libpthread, libgcc, libdl, libpam, libcrypt, librt
  - IV. Utility Libraries
    - libz, libncurses, libutil
  - V. Commands and Utilities
  - VI. Execution Environment
    - /dev, /etc, /sbin, pam, stb...
  - VII. System Initialization
  - VIII. Users and Groups
  - IX. Package Format and Installation
- 
-

# LSB III. - Base Libraries

- Alapvető programozási és működési környezet
  - Főbb tulajdonságok felhasználói szemszögből:
    - kis/NAGY betű érzékenység
    - 3x3+3 jogosultság rendszer
    - 32(64) bites egészként tárolt rendszeridő
    - speciális eszközeirő fájlok (blokkos, karakteres)
    - processzkezelés Process ID (PID) alapon
    - hálózati (és helyi) kommunikáció Socket alapon
    - reguláris kifejezések támogatása
    - fájlnev kiterjesztések (globbing) támogatása
    - környezeti változók, futáskörnyezet (environment)
    - stb...
- 
-

# *LSB/V. - Commands and Utilities*

- kb. 140 szabványosított parancs (alkalmazás)
  - Processzkezelés, időzítés
    - at, crontab, kill, killall, ps, renice, time, wait
  - Szövegmanipuláció
    - awk, ed, cat, sort, split, ed, egrep, head, tail, sed, tr
  - Fájrendszer
    - chmod, chown, cp, dd, df, du, file, find, gzip, ln, rm, ls
  - Rendszer
    - date, dmesg, env, fuser, useradd, groupadd, groups, hostname, man, su, mount, sendmail
  - Scripting, programozás
    - [, test, echo, expr, false, true, make, strip, pwd, xargs
- 
-

# LSB/VI. - Execution Environment

- File System Hierarchy Standard (FHS)
    - külön szabvány: <http://www.pathname.com/fhs/>
    - az LSB-ben csak a kivételek vannak
      - egyes szimbolikus linkek nem kötelezőek
      - /dev bővebb lehet, major, minor nem annyira szigorú
      - /etc tartalma bővebb: cron.d, cron.\*, init.d, profile.d
      - több fenntartott név: <http://www.lanana.org/>
      - user accounting fájlok nem kötelezőek
      - rendszerparancsok helye: /sbin, /usr/sbin, /usr/local/sbin
  - Additional Recommendations (biztonság)
  - Additional Behaviors (POSIX konformancia mód)
  - Localization
    - locale támogatás (collation, dátumforma, multibyte, ...)
- 
-

# Önálló feladat – FHS

Kövesse a továbbiakban elmondottakat saját számítógépén!



# FHS – / (*root*)

- / - gyökér fájlrendszer – általában nem osztható
    - /bin – alapvető parancsok mindenki számára
    - /boot – boot loader kellékek
    - /dev – speciális és eszközfájlok
    - /etc – rendszerkonfiguráció
    - /home – felhasználók munkakönyvtárai (opc.)
    - /lib – alapvető függvénykönyvtárak, modulok
    - /media – cserélhető adattárolók csatolási pontja
    - /mnt – állandó adattárolók csatolási pontja
    - /opt – önálló bővítőcsomagok
    - /root – adminisztrátor munkakönyvtára (opc.)
    - /sbin – rendszer parancsok
    - /srv – rendszerszolgáltatások adatai (pl. webroot)
    - /tmp – átmeneti állományok
- 
-

# FHS – /usr

- /usr – alkalmazási terület – általában osztható
    - /usr/X11 – grafikus környezet (opc.)
    - /usr/bin – általános felhasználói parancsok
    - /usr/include – C függvénykönyvtárak header fájljai
    - /usr/lib – általános függvénykönyvtárak
    - /usr/sbin – általános rendszerparancsok
    - /usr/share – architektúrafüggetlen adatok
    - /usr/share/man – kézikönyv oldalak
    - /usr/src – forráskódok
- 
-

# FHS – /var

- /var – változó adatfájlok – részben osztható
    - /var/account – process accounting naplók (opc.)
    - /var/cache – applikációk gyorsítótára
    - /var/crash – összeomlási naplók (opc.)
    - /var/lib – változó állapotinformációk
    - /var/lock – zárolási állományok
    - /var/log – naplófájlok
    - /var/mail – felhasználói postafiókok (opc.)
    - /var/run – futásidejű változó adatok
    - /var/spool – várakozási sorok
    - /var/tmp – rebootok közt megőrzött átmeneti fájlok
- 
-

# *FHS – a /etc tartalma (opcionális)*

- exports – NFS exportált kötetek
  - fstab – statikus fájlrendszer információk
  - group – rendszer csoportok
  - hosts – hosztnév és -cím információk
  - hosts.allow, hosts.deny – TCP korlátozások
  - inittab – az init program konfigurációja
  - mtab – dinamikus fájlrendszer információk
  - passwd – rendszer felhasználók
  - profile – alapértelmezett rendszerprofilok
  - resolv.conf – névfeloldás konfigurációja
  - services – szolgáltatások jegyzéke
- 
-

# LSB/VII. - System Initialization [1]

- cron – ütemezett futtatások
    - /etc/cron\* fájlok és könyvtárak
    - min hour day month weekday [username] command
    - pl.: *\* / 5 0 \* 1 1 root cat /etc/passwd*
      - jelentése: 5 percenként; 0 órakor; minden nap; januárban; hétfőn; a root felhasználó; futtassa a *cat /etc/passwd*-t
  - init scriptek – rendszerszolgáltatások
    - általában az /etc/init.d könyvtár fájljai (opc)
    - szabványos argumentumok: start, stop, restart, stb...
    - szabványos visszatérési értékek: 0-OK, 1-nem fut, ...
    - szabványos függőségi információk
- 
-

# LSB/VII. - System Initialization [2]

- Futási szintek (Run Levels)
    - 0 – halt
    - 1 – single user mode
    - 2 – multi user mode (no network services)
    - 3 – normal, full multiuser
    - 4 – reserved (default: 3)
    - 5 – normal, multiuser with display manager (grafikus)
    - 6 – reboot
  - Minden init script indítása és leállítása futási szintekhez rendelve (/etc/rcX.d könyvtárak)
  - A futási szintek felelőse az *init* processz
    - a rendszer első szintű processze, PID:0
- 
-

# LSB/VIII. - Users and Groups

- Minden felhasználóhoz tartozik
    - 1 darab UID (user ID)
    - 1 darab elsődleges GID (primary group ID)
    - tetszőleges számú járulékos GID
  - Speciális felhasználók (név/csoport)
    - root/root (UID:0, GID:0) – rendszergazda
    - nobody/nogroup (UID:65534, GID:65534) – senki
    - LSB ajánlás szerint továbbá:
      - adm/adm – adminisztrátor
      - lp/lp – nyomtató alrendszer
      - man/man – kézikönyvek kezelése
      - halt/halt – rendszer leállítás, stb...
  - GID 0-99 – rendszer célra fenntartva
- 
-

# LSB/IX. - Package Format and Inst.

- Az LSB ellenzők egyik fő támadási területe
  - LSB: Korlátozott RPM csomagformátum
    - A nem RPM alapú disztribúciók valószínűleg nem fognak átállni
    - Debian: *alien* program a csomagkonvertáláshoz
  - Szabványos tartalom
    - név, rövid leírás, verziószám, ellenőrzőkódok, aláírás
    - becsomagolt állományok („payload”)
    - telepítést elősegítő szabványos felületű scriptek
    - függőségkezelés
    - névtér-kezelés
- 
-

# *A Linux boot folyamata*

- A rendszer üzembeállása (helyi alapeset)
    - számítógép bekapcsolása, BIOS, POST
    - boot loader indulása, esetleg menü (GRUB, LILO)
    - kernel image betöltése
    - initial ram disk image (initrd) betöltése (opcionális)
    - kernel indítása, beépített driverek inicializálása
    - initrd indítása, initrd driverek, egyéb inicializáció (opc)
    - root (/) fájlrendszer felcsatolása
    - init processz indítása (konfigurációja: /etc/inittab)
    - a beállított runlevel elérése (alprocesszek indítása)
    - login prompt (karakteres, vagy grafikus)
    - shell vagy grafikus környezet indítása
- 
-

# Alapvető shell használat

- A Linux nem grafikus operációs rendszer
    - (DE létezik hozzá több grafikus környezet)
  - Shell – karakteres parancskörnyezet
    - a rendszer konfigurációjának alapvető felülete
  - általános shell: *BASH* (Bourne Again SHell)
    - beépített parancsok (cd, pwd, kill)
    - külső parancsok elérhetők (*PATH* környezeti változó)
    - history támogatás (fel-le gombok, ctrl-r)
    - „job” (shell alprocessz) kezelés (bg, fg, jobs, ctrl-z, &)
    - *alias* támogatás
    - beépített *help* parancs
    - konfigurációja: `[.]bashrc` és/vagy `[.]bash_profile` fájlok
- 
-

# Kézikönyvek és súgók

- Helyileg:
    - *man* parancs (kézikönyv oldalak)
      - *apropos* – keresés a leírásokban és címekben
      - *whatis* – listázás kulcsszó szerint
    - *info* rendszer (hierarchikus)
    - beépített bash *help* parancs
    - szabványos *--help* kapcsoló sok parancsnál
  - Internet:
    - *howto* (hogyan) – összetettebb téma, konkrét leírás
    - *faq* (gyik) – gyakran ismételt kérdések
    - <http://www.tldp.org>
    - <http://tldp.fsf.hu>
- 
-

# Önálló gyakorlás – *man, help, info*

Próbáljuk ki a *man, help* és *info* parancsok működését!



# Általános parancsok

- ls, pwd, tree, cd, cp, mv, rm, ln, mkdir, rmdir, cat, head, tail, more, less
  - vi, nano, mcedit, joe, emacs, mc
  - tar, gzip, gunzip, bzip2, rsync
  - find, sed, awk, xargs, read
  - ps, top, &, bg, fg, jobs, kill, killall
  - mount, umount, df, du
  - write, talk, ytalk, mesg, w, who, whoami, finger, last, lastlog, adduser, userdel, usermod, groupadd, groupdel, groupmod, groups,
  - at, atq, atrm, cron, crontab, date
- 
-

# Az „ls -l” értelmezése

- *-rw-r--r-- 1 root adm 6 2009-02-12 20:54 mailrc*
    - első karakter: fájl típus (-,d,l,c,b,p,s,?, stb...)
    - következők: 3x3 jogosultság (user, group, others)
      - r: read (1,4,7) – olvasás (d: könyvtárlista olvasása)
      - w: write (2,5,8) – írás (d: fájlbejegyzés létrehozás, módosítás, törlés)
      - x: execute/search (3,6) – futtatás (d: könyvtárváltás)
      - S: setuid (3) vagy setgid (6) – tulajdonos vagy csoport öröklése (d: futtatás vagy fájlbejegyzés létrehozása esetén)
      - T: sticky/restricted (9) – nem használt (d: csak a tulajdonos törölheti vagy nevezheti át a könyvtárbejegyzéseket)
      - s: S+x, t: T+x
    - további információk: linkek száma, tulaj, csoport, blokkszám, módosítás dátuma, fájlnev (linkeírás)
- 
-

# Az „ls -l” értelmezése – példa

```
mcree@csatt:/opt/teszt$ ls -l
total 1380
srw-rw-rw- 1 root root          0 2009-02-15 13:15 acpid.socket
crw-rw---- 1 root audio 14, 12 2009-02-15 13:14 adsp
-rwxr-xr-x 1 root root 702160 2008-05-12 20:33 bash
-rw-r--r-- 1 root root          0 2009-02-15 13:16 emptyfile
lrwxrwxrwx 1 root root          13 2009-02-15 13:14 MKDV -> /MAKEDEV
-rw-r--r-- 1 root root       1473 2009-02-12 20:54 passwd
brw-rw---- 1 root disk    8,  0 2009-02-15 13:15 sda
drwx--x--x 2 root root          6 2009-02-15 13:14 subdir01
drwxr-xr-x 2 root root         19 2009-02-15 13:16 subdir02
-rwsr-xr-x 1 root root 702160 2008-05-12 20:33 suidbash
prw-r--r-- 1 root root          0 2009-02-15 13:15 teszt.fifo
drwxrwxrwt 2 root root          6 2009-02-15 13:14 tmp
```

# 01. lecke

Alapvető listázó parancsok (pwd, ls, tree)



# *02. lecke*

Alapvető navigáció (cd, pwd)



# 03. lecke

Fájl manipuláció (cp, mv, rm)



# 04. lecke

Könyvtár manipuláció (mkdir, rmdir, rm -r)



# *05. lecke*

Archiválás, tömörítés (zcat, zless, gzip, tar)



# *Második alkalom*

## Második alkalom

2.1 fájlok manipulációja

2.2 I/O átirányítás

2.3 reguláris kifejezések



# 06. lecke

Fájlok tartalmának kezelése  
(cat, head, tail, more, less)



# 07. lecke

Karakteres szövegszerkesztő programok  
(vi, nano, mcedit, joe, emacs)



# Linkkezelés

- Hard linkek
    - egy fájl vagy könyvtár több néven érhető el
    - csak egy fájlrendszeren (blokk eszközön) belül
    - a fájl akkor törlődik, ha minden linket törölünk
  - Szimbolikus linkek (symlink)
    - a symlink egy speciális „szövegfájl”
    - tartalma egy elérési útvonal (lehet relatív is)
    - ha az elérés útvonal nem valós, a symlink „broken”
    - nem átlátszó a kezelése a több parancs számára sem
- 
-

# 08. lecke

Linkek (ln)



# Több fájl együttes kezelése

- más néven: „globbing”, „wildcard matching”
  - A speciális karaktereket a fájlnevekben kibontják
    - ? - tetszőleges karakter (1 darab karakter helye)
    - \* - tetszőleges karaktersorozat (üres is, hosszabb is)
    - [...] - karakter osztály (1 darab karakter helye)
      - *[abcd]* – 1 darab karakter, a vagy b vagy c vagy d
      - *[a-z0-9]* – 1 darab karakter, a-z között vagy 0-9 között
      - *[!ab]* – 1 darab karakter, nem a és nem b
    - a rejtett fájlok (. kezdetűek) alapból nem illeszkednek!
    - POSIX szerint az illeszkedés nélküli wildcard önmaga!
    - az illeszkedés lokalitás-függő (LC\_COLLATE)!
    - lásd: *man glob*
- 
-

# 09. lecke

Fájlok együttes kezelése (\*, ?, [])

---

---

# I/O csatornák (bash)

- Linuxban minden processzhez tartozik:
    - *stdin* – standard bemenet - /dev/stdin, /proc/self/fd/0
    - *stdout* – standard kimenet - /dev/stdout, /proc/self/fd/1
    - *stderr* – standard hibakim. - /dev/stderr, /proc/self/fd/2
  - Átírányítások:
    - *parancs < fájl* – fájl a parancs stdin csatornájára
    - *parancs > fájl* – parancs stdout csatornája fájlba
    - *parancs >> fájl* – mint >, de hozzáír, nem felülír
    - *parancs1 | parancs2* – parancs1 stdout továbbküldésre parancs2 stdin-re (parancs láncolás)
    - *parancs &> fájl* – parancs stdout és stderr fájlba
    - *parancs 2>&1* – parancs stderr stdoutra
- 
-

# 10. lecke

I/O csatornák



# Reguláris kifejezések – grep 1

- A cél: illeszkedő mintákat keresni
  - Bővített POSIX reguláris kifejezések (egrep):
    - Karakterosztály: [...]
      - Felsorolás: [abcd1234]
      - Intervallum: [a-zA-Z0-9]
      - Negálás: [^abcd1234]
      - Szimbolikus: [[:alnum:][:print:][:punct:][:space:]...]
      - Speciális jelentésű karakterek illesztése: [^\_-]
    - Tetszőleges karakter: .
    - Horgonyok:
      - Sor eleje és vége: ^ és \$
      - Szó eleje és vége: \< és \>
    - Speciális karakterek érvénytelenítése: \
- 
-

# Reguláris kifejezések – grep 2

- Bővített POSIX reguláris kifejezések (fojt.):
  - Csoportosítás (rész-kifejezések):
    - Definiálása: *(kifejezés)*
    - Alternálása: *(kifejezés|kifejezés2|...)*
    - Hivatkozása (n-edik részkifejezés értéke):  $\backslash n$
  - Ismétlődések (az előző elem vagy rész-kifejezés...):
    - Opcionális: ? pl.:  $a?$
    - Opcionálisan ismétlődik: \* pl.:  $a^*$
    - Legalább egyszer ismétlődik: + pl.:  $a^+$
    - Pontosan n-szer ismétlődik:  $\{n\}$  pl.:  $a\{4\}$
    - Legalább n-szer ismétlődik:  $\{n,\}$  pl.:  $a\{4,\}$
    - Legfeljebb m-szer ismétlődik:  $\{,m\}$
    - Legalább n-szer, legfeljebb m-szer ismétlődik:  $\{n,m\}$

# Reguláris kifejezések – grep 3

- Összetett példák:
  - $^[ab]^+$
    - Azon teljes sorok, melyek legalább egy a, vagy b karakterből állnak$
  - $^{\{4,10\}}$
    - A minimum 4, maximum 10 karakter hosszúságú teljes sorok$
  - $[0-9]{2,}.*\.[jpe?g]$
    - A legalább 2 számjegyű számot tartalmazó .jpg és jpeg kiterjesztésű fájlnevek$
  - $[a-zA-Z0-9.-]^+@[a-zA-Z0-9.-]^+\.[a-zA-Z0-9.-]^+$
    - Szintaktikailag helyes email cím (valami@valami.valami)$

# Reguláris kifejezések – grep 4

- A *grep* parancs használata:
    - *grep opciók minta fájlnev fájlnev2 ...*
  - A *grep* parancs szabványos paramétereit:
    - Bővített reguláris kifejezések használata: *-E*
    - Illeszkedési minták olvasása fájlból: *-f fájlnev*
    - Kis-nagybetű érzéketlenség: *-i*
    - A nem illeszkedő sorok megjelenítése: *-v*
    - Illeszkedések megszámlálása: *-c*
    - Illeszkedő fájlok listázása: *-l* (nem illeszkedők: *-L*)
    - Rekurzív illeszkedés alkönyvtárakban is: *-r*
- 
-

# 11. lecke

Reguláris kifejezések (grep)



# *Harmadik alkalom*

## Harmadik alkalom

3.1 fájlok keresése

3.2 környezet

3.3 programozott szövegfeldolgozás

3.4 processzkezelés

3.5 blokk eszközök



# 12. lecke

Fájlok keresése (find)



# 13. lecke

Programozott szövegfeldolgozás (sed, xargs)



# Környezet (*environment*)

- Linux alatt minden processzhez tartozik egy környezet (kulcs-érték párok és függvények)
  - Környezet bővítése: értékadással (=)
    - pl: „*PATH=.*”
  - *set* parancs (bash beépített): paraméter nélkül a környezet listája
  - *export* parancs (bash beépített): környezeti változó kiterjesztése az alprocesszekre is
    - pl: „*export PATH=.*” (értékadás és exportálás)
    - vagy: „*export PATH*” (már definiált változó)
    - vagy: „*export PATH=*” (törlés)
- 
-

# Az elérési útvonal (*PATH*)

- Az elérési útvonal a *PATH* környezeti változóban
  - : karakterekkel elválasztott könyvtár útvonalak
  - lekérdezése: „*echo \$PATH*” parancs
  - az alapértelmezett *PATH*-ban általában nincs benne az aktuális könyvtár (.)
- *which* parancs: az elérési útvonalban keresve melyik alkalmazást indítaná a rendszer

# Önálló gyakorlás – *env*, *PATH*

Egy új terminál ablakban (*bash*) kérdezze le az aktuális *PATH* értékét, majd törölje ki! Próbálja ki a megváltozott *PATH* mellett pár alapvető parancs futtatását! Állítsa be, hogy a *PATH* csak az aktuális munkakönyvtárat tartalmazza (.)! Próbálja ki ezt a működési módot is, miközben különböző munkakönyvtárakban tartózkodik (pl.: */usr/bin* vagy */bin*)! Próbálja ki az „*export*” parancs működését: a módosított *PATH* érték mellett indítson egy alárendelt shellt („*/bin/bash*” parancs) és figyelje meg, milyen *PATH* értéket örökölt az új parancsértelmező különböző beállítások mellett!

---

---

# Processzek kezelése 1

- Processz lista megtekintése
    - processzek listája: *ps* (több fajta paraméterezést ismer)
      - összes processz: *a*, *-A*, *-e*
      - terminál nélküli processzek is: *x*
      - felhasználó információk: *u*, *l*, *-L*, *j*
      - teljes parancssor: *w*
      - processz hierarchia: *f* (lásd még: *pstree* parancs)
    - interaktív, rendezett lista: *top* (súgó: *h* billentyű)
  - *bash* job control:
    - Jobok listája: *jobs*
    - Aktuális processz felfüggeszt: *ctrl-z*, leállít: *ctrl-c*
    - Job háttérbe: *bg jobspec*, *parancs &*
    - Job előtérbe: *fg jobspec*
- 
-

# Processzek kezelése 2

- Jelzés (signal) küldése processznek:
    - *kill -signal PID*
      - Lásd: *man signal*
      - 9, *KILL* – terminálás feltétel nélkül
      - 15, *TERM* – leállítás (ctrl-c)
      - 17, *STOP* – ctrl-z
      - 19, *CONT* – fg
    - futó parancsok PID-je: *pgrep parancs*
    - jelzés küldése futó parancsoknak: *killall -signal parancs*
  - Processz prioritásának beállítása (-20 – 19)
    - futtatáskor: *nice -n érték parancs*
    - futás során: *renice érték PID*
    - 0: alap prioritás, 1-től: alacsony pri., -1-től: magas pri.
- 
-

# Önálló gyakorlás - processzkezelés

Futtasson egy „*sleep 1m*” parancsot (hatása: 1 perc tétlenség). Szüneteltesse (*ctrl-z*) a futását segítségével, majd listázza a jobokat és küldje a háttérbe a parancsot (*bg*), majd várja meg, hogy befejeződjön! Indítson egy újabb *sleep* parancsot, szüneteltesse, majd keresse meg a processz azonosítóját (PID) a *ps* parancs segítségével. A kiderített PID számára küldjön egy TERM jelzést! Az *&* operátor segítségével indítson több hosszabb *sleep* parancsot, majd állítsa le őket *killall* segítségével (a teszteléshez használjon *pgrep* parancsot!)

---

---

# Blokk eszközök és a szabad hely

- Szabad hely a lemezen: *df*
    - Könnyebben értelmezhető kimenet: *-h*
  - Egy könyvtár/fájl helyfoglalása: *du*
    - Könnyebben értelmezhető kimenet: *-h*
    - Ne lépjen alkönyvtárakba: *-s*
  - Felcsatolt blokk eszközök listája: *mount*
  - Blokk eszköz
    - felcsatolása: *mount eszköz könyvtár*
    - lecsatolása: *umount könyvtár*
  - Fájrendszer
    - létrehozása: *mkfs*
    - ellenőrzése: *fsck*
- 
-

# Önálló gyakorlás – blokk eszközök

Előkészítés:

```
dd if=/dev/zero of=/tmp/fajl bs=1k count=10k  
mkfs.ext2 /tmp/fajl  
sudo mount -o loop /tmp/fajl /tmp/konyvtar
```

Próbáljuk ki az ismertetett parancsok működését!

Tipp: a *mount* és *umount* parancsok használatához a legtöbb esetben root jogosultság kell!  
(lásd még: *man sudo*)

---

---

# *Negyedik alkalom*

## Negyedik alkalom

- 4.1 Felhasználók kezelése
- 4.2 Jogosultságok
- 4.3 Ütemezett parancsindítás
- 4.4 Rendszeridő



# Felhasználók kezelése

- Az érvényes felhasználó neve: *whoami*
  - A legutóbbi belépések listája: *last, lastlog*
  - Felhasználók kezelése: *adduser, useradd, deluser, userdel, usermod*
  - Csoportok kezelése: *groupadd, groupdel, groupmod, groups*
  - Aktuális felhasználó váltása: *su, sudo*
  - Jelszó változtatás: *passwd*
  - Érvényes felhasználók listája: */etc/passwd* fájl
  - Csoportok listája: */etc/group* fájl
- 
-

# Felhasználók közti kommunikáció

- Üzenetküldéssel kapcsolatos parancsok:
  - napjainkban nagyon ritkán használt funkciók
  - *write, talk, ytalk, mesg, wall*
- Felhasználó adatainak lekérdezése (ritka): *finger*
- Belépett felhasználók listája:
  - *w, who*

# Önálló gyakorlás – felhasználók

Hozzon létre egy új felhasználót, egy új csoportot!  
Adja a felhasználót a csoporthoz! Váltson át az új felhasználóra és próbálja meg írni, olvasni a „hallgato” felhasználó fájljait, könyvtárait! Hozzon létre új fájlakat a /tmp könyvtárban több felhasználó nevében.

Próbálja ki az `ls -l` parancsot és figyelje meg a különböző tulajdonosokat, próbálja ki jogaikat!

Próbálja ki a kommunikációs parancsokat!

---

---

# Linux jogosultságkezelés

- 3x3(+3) rendszer (alapvető, általános)
    - 3 halmaz: *User*, *Group*, *Others* (betűk: *ugo*)
    - 3 jog: *Read*, *Write*, *eXecute* (betűk: *rwX*)
    - 3 további jelzőbit: *Setuid*, *Setgid*, *sTicky* (betűk: *STst*)
    - minden fájlhoz tartozik: *owner* (*UID*) és *group* (*GID*)
    - *ls* és *stat* parancsokkal kérdezhetők le
    - *chmod* és *chown* parancsokkal manipulálhatók
  - A Linux jogosultságkezelés bővítései:
    - POSIX ACL-ek (*getfacl*, *setfacl* parancsok)
    - Kiterjesztett attribútumok (*lsattr*, *chattr* parancsok)
    - Linux Capabilities (2.2 kernel óta)
    - Kernel bővítmények (Grsecurity RSBAC, SELinux)
- 
-

# A „*chmod*” parancs

- A 3x3+3 jogosultság megadása többféleképp
  - szimbolikus: [ugoa...][+ -=][rwxXst...],...
    - ugoa: 'U'ser, 'G'roup, 'O'thers, 'A'll(=ugo)
    - '+': hozzáad, '-': elvesz, '=': beállít
    - rwxXst: 'R'ead, 'W'rite, e'X'ecute (X: dir only), 'S'etUID/GID, s'T'icky/restricted
    - [ugo...][+ -=][ugo...],... - jogosultság másolása
  - oktális (nyolcas szrendszer): 0SUGO
    - 'S'peciális: 1: sticky, 2: setGID, 4: setUID
    - 'U', 'G', 'O': 1: execute, 2: write, 4: read
  - például: *u=rw,g=wx,o=rt* → *01634*
- 
-

# *umask*

- Alapértelmezett létrehozási jogosultságok
  - a *chmod* jogosultságaihoz hasonló
  - bitenkénti ÉS kapcsolattal működik
    - 00022 → u=rwx,g=rx,o=rx
  - *umask* parancs
    - paraméter nélkül az aktuális érték
    - paraméter megadása: érték megváltoztatása
    - *-S* kapcsoló: szimbolikus üzemmód
  - az 'x' maszk (jog) alapértelmezésben csak könyvtárra érvényesül
  - az *umask* nem korlátoz, csak alapértelmezés!
- 
-

# Önálló gyakorlás – jogosultságok

Hozzon létre 3 felhasználót ( $a, b, c$ ) és 2 csoportot ( $aa, bb$ )! Az  $aa$  csoport tagja legyen  $a$  és  $b$ , a  $bb$  csoport tagja legyen  $b$  és  $c$ ! Minden felhasználóhoz tartozzon egy saját elsődleges csoport is, azonos néven! Hozzon létre az  $aa$  és  $bb$  csoportnak is egy-egy munkakönyvtárat, amiben csak az adott csoport tagjai rendelkeznek írási és olvasási jogosultsággal! Adja meg a felhasználók *umask* értékeit úgy, hogy harmadik fél számára ne legyenek hozzáférhetőek az alapértelmezetten létrehozott fájlok! Csak az  $a$  felhasználó férjen hozzá a  $c$  felhasználó home könyvtárában levő fájlokhoz olvasásra!

---

---

# Ütemezett parancsindítás

- Késleltetett futtatás (ritkán használt):
    - *at, atq, atrm*
  - Időzített futtatás:
    - Lásd: LSB/VII.!
    - a felhasználó saját ütemező állománya:
      - Formátuma ugyanaz, mint a rendszerszintűé, - felhasználónév nélkül
      - Fájl érvényesítése: *crontab fájl*
      - Aktuális crontab listázása: *crontab -l*
      - Crontab törlése: *crontab -r*
- 
-

# Önálló gyakorlás – ütemezés

Hozzon létre egy felhasználói szintű cron ütemezési konfigurációt, mely egy fájlhoz minden percben hozzáfűz egy új sort!



# Rendszeridő

- Rendszeridő lekérdezése és beállítása: *date*
    - paraméterek nélkül: aktuális idő
    - idő beállítása: *-s idő*
    - adott időpont megjelenítése: *-d idő*
    - formátum változtatása: *date paraméterek +formátum*
    - a *date* nagy biztonsággal megérti az angol nyelvi formátumú időmegadásokat (pl.: „*now +2 days*”)
    - lásd még: *man date*
  - Fizikai óra programozása (ritka): *hwclock*
  - Internetes idő szolgáltatások: *ntp, chrony, rdate*
- 
-

# Önálló gyakorlás – rendszeridő

A *date* parancs segítségével derítse ki, az Ön születésnapja a hét melyik napjára esett!

# *Ötödik alkalom*

Ötödik alkalom

5. Shell Scriptek



# Shell Scriptek 1

- „batch fájlok”, „kötegetelt feldolgozás”
  - futtatásuk (bash használata esetén):
    - aktuális shell: *. fájlnev*
    - subshell: *bash fájlnev*
    - subshell: *#!/bin/bash* az első sorban és eXecute jog
  - komment kezdő karakter: *#*
  - shell változók (karakter sorozatok tárolására)
    - létrehozása: *valtozo=ertek*
    - felhasználása (olvasása): *\$valtozo*
    - a futtatott script neve: *\$0*
    - parancssori argumentumok: *\$1 - \$9*
    - környezeti változó definiálása: *export \$változó*
- 
-

# Shell Scriptek 2

- Linuxban a parancs visszatérési értéke
    - 0: „OK”, IGAZ érték
    - nem 0: „NEM OK”, HAMIS érték
  - parancsok végrehajtása:
    - sorban: *parancs1; parancs2 ...* (új sor = ;)
    - ÉS kapcsolat: *parancs1 && parancs2 ...*
    - VAGY kapcsolat: *parancs1 || parancs2 ...*
  - parancsok csoportosítása:
    - aktuális shell: *{ parancs1; parancs2; stb.... ; }*
    - subshell: *( parancs1; parancs2; stb.... )*
  - háttérben futtatás: *parancs &*
- 
-

# Shell Scriptek 3

- aritmetikai és logikai shell kifejezések:
  - aritmetikai kiértékelés (egész számok): *(( kifejezés ))*
    - egyoperandusú operátorok: *++vált*, *--vált*, *vált++*, *vált--*
    - matematikai operátorok: *+*, *-*, *\**, */*, *%* (marad.), *\*\** (hatv.)
    - bitművelet operátorok: *<<*, *>>*, *~*, *|*, *&*, *^* (xor)
    - logikai operátorok: *!*, *&&*, *||*
    - értékadó operátorok: *=*, *\*=*, */=*, *%=*, *+=*, *-=*, *<<=*, stb...
  - feltétel kiértékelés: *[[ kifejezés operátor minta ]]*
    - string mintaillesztés operátorok: *==*, *!=*
    - regex mintaillesztés operátor: *=~*
    - numerikus mintaillesztés operátorok: *<*, *>*, *<=*, *>=*

# Shell Scriptek 4

- Vezérlési szerkezetek
  - *for név in string do parancsok ; done*
    - *\$név* a *string* **globbing** paraméter(ek) értékén fut végig
  - *for (( kif1 ; kif2 ; kif3 )) ; do parancsok ; done*
    - klasszikus C for ciklus (( init ; test ; increment ))
  - *select név in string do parancsok ; done*
    - *\$név* a *string* paraméter(ek) értékei közül választható
  - *case string in minta[|minta...] ) parancsok ;; .... esac*
    - többes elágazás a *string* és *mint(a/ák)* megfeleltetése alapján
  - *if parancsok; then parancsok; fi*
    - *if ... then ... elif ... else ... fi*
  - *while parancsok; do parancsok; done*
  - *until parancsok; do parancsok; done*
  - *break, continue*

# Shell Scriptek 5

- Shell függvények
  - *[function] név() { parancsok; }*
  - nem felüldefiniálhatók
  - \$0-\$9 újraértelmeződnek
- Metakarakterek
  - bash metakarakter: \
  - sorfolytonosság megtartása: \<sortörés>
  - spec: \\$ \” \` \a \b \e \f \n \r \t \v \\ \' \nnn \xHH \cx
  - spec használata: \$'string' vagy echo -e

# Shell Scriptek 6

- Stringen belüli értelmezés
    - ” jeleken belül értelmezett: \$, `, \
    - ' jeleken belül nincs belső értelmezés
    - multilanguage support: \$"string"
  - Speciális változók, paraméterek
    - \$0 - script neve
    - \$1 - \$9 - parancssori (vagy függvény) argumentumok
    - \$\* - a teljes parancssor \$IFS karakterekkel elválasztva
    - "\$@" - a teljes parancssor megfelelően idézőjelezve
    - \$# - a parancssori paraméterek száma
    - \$? - a legutóbbi parancs visszatérési értéke
    - \$\$ - az aktuális shell process ID (PID) értéke
    - \$\_ - a legutóbbi parancs paraméterei
- 
-

# Shell Scriptek 7

- A bash 7 fajta behelyettesítést végez
  - kapcsos zárójel behelyettesítés
    - $a\{b,c,d..f\}g = abg\ acg\ adg\ aeg\ afg$
  - tilde (~) behelyettesítés
    - ~/ - aktuális user home könyvtára
    - ~user/ - adott user home könyvtára
  - paraméter behelyettesítés
    - $\$n\acute{e}v$  vagy  $\${n\acute{e}v}$  – shell változó értékének behelyettesítése
    - sok egyéb (alapértékek, műveletek) – lásd: *man bash*
  - parancs behelyettesítés
    - $\$(parancs)$  vagy  $\`parancs\`$  - a *parancs* stdout tartalma
- 
-

# Shell Scriptek 8

- aritmetikai behelyettesítés
    - $\$(kifejezés)$  – a kifejezés értéke
  - processz behelyettesítés
    - $<(parancs)$  vagy  $>(parancs)$  FIFO-t hoz létre
  - „word splitting” behelyettesítés
    - egyéb behelyettesítések után az  $\$IFS$  alapján (!!!)
  - path behelyettesítés (globbing)
    - \*, ? és [...] (karakterosztály) behelyettesítés
  - idézet (quote) behelyettesítés
    - nem egyéb behelyettesítésből származó ', " és \ karakterek eltávolítása
  - további info a behelyettesítésekről: *man bash*
- 
-

# Shell Scriptek - irodalom

- további hasznos parancsok
    - *true, false, read, alias, printf, command, test, xargs, pushd, popd, shopt, set, eval, let, trap, exec, source, caller, compgen, complete, exit, mktemp, hash, history, local, return, shopt, unset, wait*
  - Irodalom
    - *man bash*
    - *man test*
    - *help*
    - <http://www.faqs.org/docs/bashman/>
    - BASH FAQ: <http://woledge.org:8000/>
- 
-

# Példafeladat 1

- Készítsen shell scriptet, mely a képernyőn megjeleníti az első három parancssori paraméterét futtatáskor!
  - Példa futtatás: *./scriptem alma 2 korte*
  - Példa eredmény: *alma 2 korte*

# *Példafeladat 1 - megoldás*

```
#!/bin/bash  
echo $1 $2 $2
```



## *Példafeladat 2*

- Készítsen shell scriptet, mely létrehozza a paraméteréül megadott fájlt!
  - Példa futtatás: *./scriptem fajlom*

# *Példafeladat 2 - megoldás*

```
#!/bin/bash
touch $1
# vagy...
echo "testing" > $1
# vagy...
> $1
# vagy...
ls > $1
```

## *Példafeladat 3*

- Készítsen shell scriptet, mely létrehoz egy fájlt a mai nap dátuma alapján, a dátum formátuma legyen *éééé-hh-nn.dat*
  - Példa eredmény: *2009-03-13-dat*

# *Példafeladat 3 – megoldás*

```
#!/bin/bash
touch `date +%Y-%m-%d`.dat
# vagy....
FILENAME=`date +%Y-%m-%d`.dat
Touch $FILENAME
# vagy stb....
```

## *Példafeladat 4*

- Készítsen shell scriptet, mely megkérdez a felhasználótól egy fájlnevet, majd létrehozza a fájlt és értesíti a felhasználót, hogy a fájl létrehozása sikerült



# *Példafeladat 4 - megoldás*

```
#!/bin/bash  
clear  
echo 'add meg a fájl nevét: '  
read FILENAME  
touch $FILENAME  
echo "$FILENAME létrehozva"
```



## *Példafeladat 5*

- Készítsen shell scriptet, mely megkérdez a felhasználótól egy fájlnevet, majd létrehozza a fájlt és értesíti a felhasználót, hogy a fájl létrehozása sikerült. A script figyelmeztessen, ha a megadott fájlnev üres.

# *Példafeladat 5 - megoldás*

```
#!/bin/bash
clear
echo 'Add meg a fájlnevet: '
read FILENAME
if [ ! -z $FILENAME ] ; then
    touch $FILENAME
    echo "$FILENAME fájl létrehozva"
else
    echo "Nem adtál meg fájlnevet."
fi
```

# Önálló script feladat 6

- Készítsen shell scriptet, mely megkérdezi a felhasználó kedvenc színét. Ha a válasz nem „piros”, jelenítsen meg hibaüzenetet, várjon 3 másodpercet, törölje le a képernyőt és tegye fel újra a kérdést.
- Tippek:
  - read, test, sleep, clear, while

# Önálló script feladat 7

- Kérdezze meg a felhasználótól egymás után a nevét, korát, nemét és kedvenc színét

# Önálló script feladat 8

- Kérdezze meg a felhasználótól egymás után a nevét, korát, nemét és kedvenc színét, majd ezeket tárolja egy, a mai dátum alapján elnevezett éééé-hh-nn.dat nevű fájlba

# Önálló script feladat 9

- Készítsen shell scriptet, mely az aktuális könyvtárban levő összes *.html* végződésű fájlból biztonsági másolatot készít *.html.old* végződéssel.

# *Hatodik alkalom*

Hatodik alkalom

6.1 Linux Kernel

6.2 Grafikus Felület

6.3 Irodai alkalmazások

Gyakorlási lehetőség



# Linux Kernel - modulok

- Monolitikus, betölthető kernel modulokkal (LKM)
  - modulok kezelése (root jogosultsággal):
    - *lsmod* – betöltött modulok listázása
    - *insmod* – egy modul betöltése
    - *modprobe* – modul betöltése függőségekkel
    - *rmmmod* – modul eltávolítása
    - *depmod* – modul függőségi lista frissítése (fordításkor)
    - *modconf* – modulok fordítása (nem mindenhol van)
  - */etc/modules* fájl: rendszerindításkor betöltendő modulok listája
  - <http://kernelnewbies.org/FAQ>
- 
-

# Linux Kernel - fordítás

- Szükséges: GNU C fordító környezet (gcc, make)
- Lépései (2.6 sorozat esetén):
  - kernel forráskód letöltése (<http://kernel.org>)
  - kernel forráskód kicsomagolása
    - jellemzően az */usr/src/* könyvtárban
    - tipikus parancs: *tar xjvf linux-2.6.0.tar.bz2*
  - kernel konfigurálása (plusz könyvtárakat igényelhet!)
    - karakteres: *make config, make menuconfig*
    - grafikus: *make gconfig, make xconfig*
  - kernel fordítása
    - *make*
  - kernel image telepítése, bootloader konfigurálása
    - *make modules\_install, make install* + grub vagy lilo
    - egyéb kézi vagy automatizált procedúra

# X.org

- A Linux általánosan használt grafikus felülete: X
  - x.org Foundation: <http://www.x.org/>
  - Futtatása:
    - Karakteres login után: X, startx
    - Grafikus login: xdm, kdm, gdm
  - Régebben sok kézi állítgatás: `/etc/X11/xorg.conf`
  - Napjainkban plug and play rendszer
  - 3D: OpenGL támogatás, DirectX értelemszerűen nincs
  - Az X felépítése:
    - Szinte már archaikus API és szerkezet, folyamatos fejlesztés
    - Moduláris szerkezet (XRandR, DRI2, Render, GLX, stb...)
  - Zárt forrású grafikus driverek léteznek: nVidia, ATI

# *Irodai alkalmazások*

- Egyéni felfedezés...
  - KDE
  - GNOME
  - [openoffice.org](http://openoffice.org)
  - mozilla suite
  - CUPS



# *Hetedik alkalom*

Hetedik alkalom

Megajánlott jegy szerzési lehetőség

