Számítógépes alapismeretek

6. előadás

Dr. Illés Zoltán

ELTE IK Média és Oktatásinformatika Tanszék

zoltan.illes@elte.hu

Ami eddig volt…

- Számítógépek architektúrája
- Alapvető alkotóelemek
  - Processzor
  - Memória
  - Perifériák
- Hardver elemek,
- Hálózatok
  - Arpanet
  - …

zoltan.illes@elte.hu

Ami ezután következik…

- Szoftver
  - Gépi kódtól az operációs rendszerig
- Operációs rendszerek és programozási lehetőségei
  - UNIX (Linux) lehetőségek
    - Nyelvi eszközök használata (gépi kód,C++, Java,stb.)
    - Shell script
  - Windows
    - Batch, nyelvi eszközök (gépi kód, C++, Java,stb.)
    - PowerShell

zoltan.illes@elte.hu

Irodalom

- Brian W. Kernighan, Rob Pike: A UNIX operációs rendszer
- Bartók Nagy János, Laufer Judit: Unix felhasználói ismeretek
- http://www.szabilinux.hu/ufi/main.htm
- Internet…..

zoltan.illes@elte.hu

Szoftveres lehetőségek – Gépi kód

- Gépi kód
  - Alacsony szintű, közvetlen erőforrás elérés, kezelés
  - Assembly – Assembler
    - Makró assembler (masm, tasm,…)
  - Z80, C64 Motorola, x86, …
  - PIC (Microchip Technology, RISC forma)
    - Processzor, memória, A/D ki-bemenetek egyben
    - Széles körben elérhető fejlesztőeszközök

zoltan.illes@elte.hu

Szoftveres lehetőségek – Operációs rendszerek

- Operációs rendszer fogalma
  - AT&T Bell labs, Unix születés , 1969
  - 1974 egyetemeken, Bell laborban elterjed
  - C programozási nyelv kialakulása
  - Unix – Linux
- IBM DOS – MS DOS
  - Microsoft kliens operációs rendszerek
    - Windows 3.1, Windows 95, 98, 2000,XP,Vista
  - Microsoft szerver operációs rendszerek
    - Windows NT, 2000, 2003, 2008 szerver

zoltan.illes@elte.hu

Unix kapcsolat

- Unix X-terminál kapcsolat
  - Jellemzően helyi konzol használat esetén.
  - Távoli X kapcsolat.
  - Mi nem fogjuk használni.
- Normál terminál (getty) kapcsolat
  - telnet (Arpanet szabvány)
    - Nyílt, nem titkosított.
    - Már csak ritkán engedélyezett!
  - Ssh (putty, stb.)
    - RSA, nem tisztán (jéggel…:)

zoltan.illes@elte.hu

Fontosabb Shell típusok

- Sh (Bourne shell)
- Ksh (Korn shell)
- Csh (C shell)
- Sh (Posix shell, a korábbi Bourne néven)
- Bash (Bourne again shell)
  - Minden felhasználó alapértelmezett shellje!
  - Parancs history, sorszerkesztés, fájlnév befejezés, alias kezelés

zoltan.illes@elte.hu

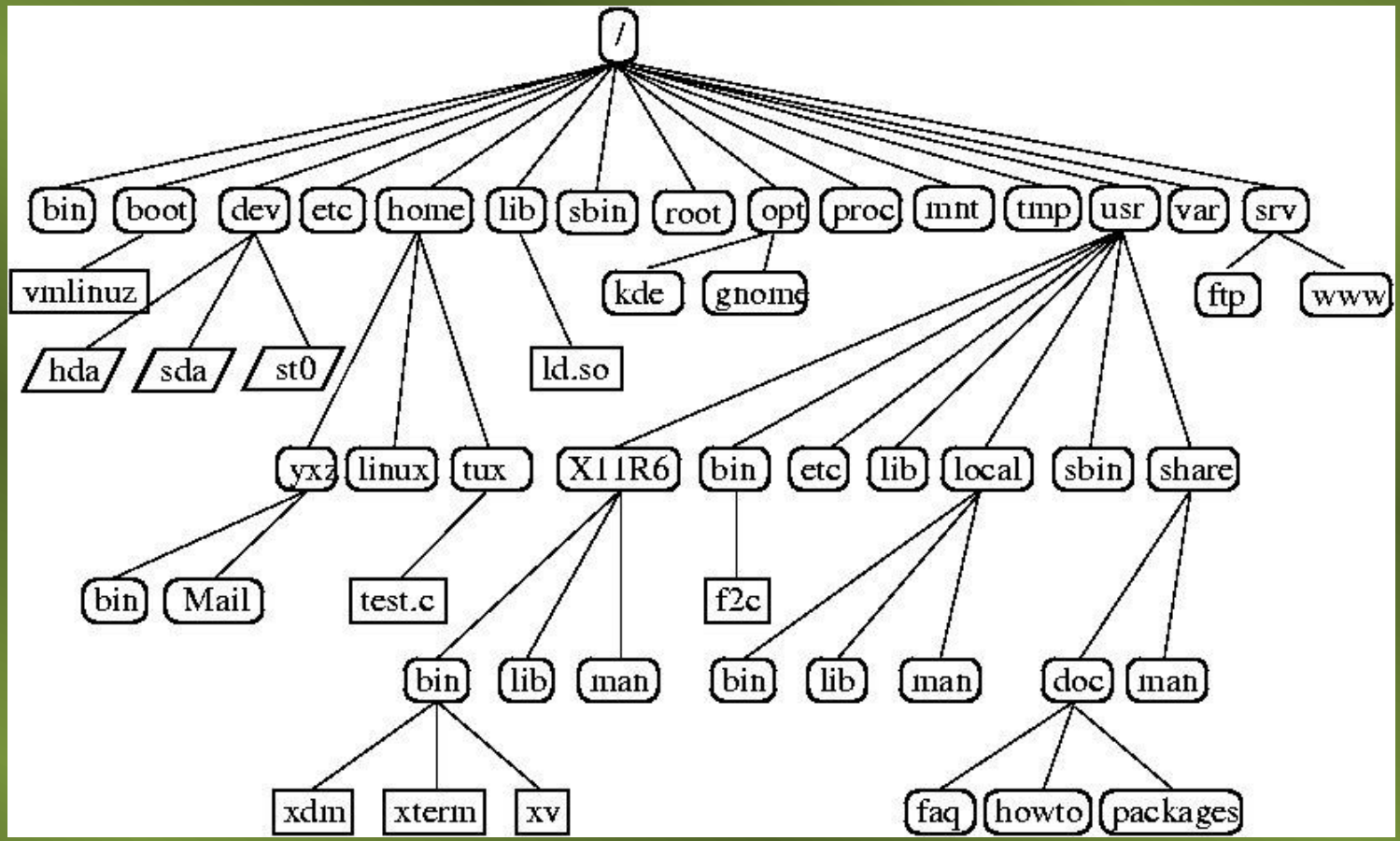
Unix fájlszerkezet I.

- Fastruktúra
  - / ,egy gyökér van, ez a: /
  - /dev/… az eszközök közös könyvtára
    - pl: /dev/fd0h1440, 1.4 floppy, /dev/null: szemétkosár
  - /etc/… konfigurációs állományok könyvtára
    - pl: /etc/passwd, felhasználók felsorolása
  - /home, /h, felhasználók könyvtára
    - pl:/h/i/illes
  - /usr/…/usr/local/…, rendszer(helyi)könyvtárak
    - pl: /usr/bin/cc, /bin/sh <-> /usr/bin/sh
  - /var/…, működési segéd, pl. logok

zoltan.illes@elte.hu

Unix fájlszerkezet II.

- Példa: OpenSuse jellemző filerendszer



zoltan.illes@elte.hu

Bash fontosabb jellemzők I.

- Fő kapcsolódási pont(mindent ebben végzünk)
- Parancssor szerkesztés, kiegészítés(tab)
  - Ha nem egyértelmű kiírja a választékot.
- Előző parancs(ok) használata (fel-, lenyil)
  - history n (az előző n parancs kiírása)
- Álnév használat
  - Alias név=szöveg
  - PL: alias dir=”ls –l”
  - dir a*

zoltan.illes@elte.hu

Bash fontosabb jellemzők II.

- Parancs szerkezet
  - Elsődleges, másodlagos prompt: PS1,PS2
  - Parancs alakja: PS1 név paraméter(ek) (enter)
  - Ha úgy érzi nincs vége a parancsnak, kapjuk a másodlagos promptot!
  - Egy sorba két (több) utasítás: ;
  - Megjegyzés: #
- Login folyamat: /etc/profile, ~/.profile végrehajtása
  - Helyi utasítások gyűjtőhelye: .profile, pl: PATH

zoltan.illes@elte.hu

Fontosabb parancsok I.

- ls, ls –l, ls –al #könyvtár tartalom
- pwd, cd, mkdir, rmdir #könyvtár műveletek
- chmod, chown, chgrp, umask # jogosultság
- passwd, kpasswd # jelszó állítás
- cp, mv, rm, ln # fájl műveletek
  - ln –s #soft link
- mail, telnet(ssh), ftp, nfs(mount) #arpanet
  - ssh név@host

zoltan.illes@elte.hu

Fontosabb parancsok II.

- find
  - find . –name alma.fa
- tar (tape archive) –kulcs [f file] fájlok
  - fontosabb kulcsok:
    - c, create, archive létrehozás
    - x, eXtract, kivesz,visszatölt archivumból
    - t, tartalom kiírása
    - v, képernyőre írja a file neveket
  - Példa: tar –cvf alma.tar *.txt
    - tar –xvf alma.tar *

zoltan.illes@elte.hu

Fontosabb parancsok III.

- cat, head, tail # fájl tartalom megnézés
- read a # a nevű változóba olvas be enterig
  - read a b # a és b változóba olvas be, a-ba az első helyközig olvas, majd a többi elem b-be kerül
- diff file1 file2 # fájlok összehasonlítása
- zip, unzip, gzip, tömörítés
  - zip alma.zip *.txt # alma.zip-be tömöríti az összes txt kiterjesztésű fájlt.
- …és még sok minden….

zoltan.illes@elte.hu

Jogosultság állítás

- Alapértelmezett jogosultság: 644
- umask, azon bitek megadása, melyekhez nem adunk jogot
  - példa: umask 111 # az új file rw-rw-rw- jogú
  - default: umask 022
- Kiegészítő jogok: Példa: chmod 6644 alma
  - setuid, parancs a fájl jogaival fut, nem a futtató jogaival (x helyett S)
  - setgid, parancs a fájl csoport jogaival fut
  - sticky bit, könyvtárban csak saját fájl módosítható

zoltan.illes@elte.hu

Változók

- Környezeti változók, mind a környezetben, mind az abból indított parancsokban láthatók.
  - env (összes környezeti változó)
  - PATH (.profile), elválasztó :
  - PS1, LOGNAME
- Shell változó, csak a shellben látható!
  - Pl: csapat=fradi
  - set (összes változó)
- Változónév betű lehet, majd szám és aláhúzás használható.

zoltan.illes@elte.hu

Változó tartalma, hatóköre

- Tartalom: $név
  - alma=fradi
  - echo $alma
  - echo Hajrá $alma !
- export név, környezeti változó lesz
  - export alma
- unset név, törli a változót
  - unset alma
  - adjuk ki a set parancsot ezután: _=alma

zoltan.illes@elte.hu

Idézőjelek

- Idézőjelek megszüntetnek spec. hatásokat.
  - \ megszünteti a következő karakter hatását
    - pl: fa=virág; echo alma\$fa #alma$fa
    - echo alma$fa #almavirág
  - ‘ belül minden hatás megszünt ‘
    - pl: echo ‘alma\$fa’ #alma\$fa
  - ” belül a $, a \ a ` és a ‘ karakterek hatása megmarad”
    - pl: a=fradi; echo ”hajrá $a !”

zoltan.illes@elte.hu

Változó behelyettesítés

- csapat=Fradi; echo $csapatalegjobb!; # ??
- echo $csapat a legjobb!;
- echo ${csapat}a legjobb!
- unset csapat; x=${csapat-Újpest}
  - inicializálás ha nem létezik
  - x=${csapat=Újpest} # csapat is változik!
- echo $x a legjobb! # ???
- y=${csapat?Szia} # ha csapat nem definiált, a Szia kírásra kerül, majd kilép a shell, nem kap y új értéket!
- y=${csapat+Újpest} # ha csapat definiált, y=Újpest

zoltan.illes@elte.hu

Parancs behelyettesítés

- ` parancs ` vagy $(parancs)
  - ki_vagyok=`whoami`
  - a=`date` ; b='date' #
  - echo $a # ???
  - echo $b # ???
    - A date szó lesz az eredmény.
  - eval $b #man eval
    - Ugyanaz, mintha a parancs $b lenne!
- x="date"; y='date' #Különbség? semmi
  - z='ls –al'; $z # működik

zoltan.illes@elte.hu

Fájlnév megadások

- Ajánlott karakterek fájlnévben:
  - betűk(nem ékezetesek),számok,_,-
- Speciális karakterek: *,?,[],!
  - ? egyetlen karakter helyettesítés
  - * tetszőleges karakter helyettesítés (0 is)
    - * nem helyettesíti a fájl elején álló pontot!
  - [abc] a felsorolt jelek közül egy
  - [!abc] nem a felsorolt karakterek közül egy
    - [A-Z] nagybetű
    - [1-9] 1,9 közti szám

zoltan.illes@elte.hu

Első shell scriptem

- Fájl neve: elso

```
illes@panda:~$ vi elso
illes@panda:~$ chmod +x elso
illes@panda:~$ cat elso
echo Ez az első shell scriptem!

illes@panda:~$ ./elso
Ez az első shell scriptem!
illes@panda:~$ elso
Ez az első shell scriptem!
illes@panda:~$
```

- Egy kis módosítás:

```
illes@panda:~$ cat elso
#!/bin/sh
#
echo Ez az első shell scriptem!
```

zoltan.illes@elte.hu

Kimenet, bemenet, átirányítás I.

- Kimenet, bemeneti eszközök
  - stdin (0) - billentyűzet, alapértelmezett bemenet
  - stdout (1) - monitor, alapértelmezett kimenet
  - stderr (2) – monitor, alapértelmezett hibakimenet
- Átirányítás
  - Kimenet: > jel segítségével
    - Pl: echo alma barack szilva >gyumolcs
    - echo alma >&2 #hogy ne a 2 nevű állomány legyen
  - Bemenet: < jel segítségével
    - Pl: mail juli <level.txt

zoltan.illes@elte.hu

Kimenet, bemenet, átirányítás II.

- Kimenet, hozzáfűzés (append)
  - >> fájlnév, Pl: echo dió >>gyumolcs
  - Ha nem létezik a fájl, akkor létre is hozza!
- Hibakimenet (stderr) átirányítás
  - 2>, 2>>
- Szimmetria miatt(☺): <<
  - Bemenet átirányítás a helyben megadott szövegre

```
$ cp 2>hiba
$ mv ezt 2>>hiba
$ cat hiba
```

```
echo <<alma
<input type=text name=X>
<input type=button>
alma
```

zoltan.illes@elte.hu

Szűrők

- Kimenet közvetlenül egy bemenetre irányul.
  - Bemenetet olvas, kimenetre ír
- Csövek: |
- Fontosabb kész szűrők:
  - cut, tee, sort, wc, grep
  - Példa:

```
$ who >nevek
$ sort nevek          #sorok szerinti sorrend
$ who|sort –r –u # fordított sorrend, egyedi sorok
$ who|wc –l           #bejelentkezett felhasználó szám
```

  - Ami biztos: man (ual) lekérdezése
    - Példa: man sort, man –k kulcsszo

zoltan.illes@elte.hu

CUT - kivágás

- Standard inputon vagy fájlból vághatunk ki mező(ke)t, oszlop(ka)t.
- cut –c1-5 1-5 karakteroszlop kivágás
  - példa: date|cut –c4-8 # Oct
- cut –f1,3,5-7 1,3,5,6,7 mezők kivágása
  - Alapértelmezett mezőelválasztó: Tab
  - Új mezőelválasztó: -d char
  - Példa: cat /etc/passwd|cut –f1,7 –d: # név, shell

zoltan.illes@elte.hu

Grep – mintaillesztés I.

- A paraméterül adott mintával rendelkező sorok kiválasztása.
- Fontosabb paraméterek:
  - -v mintát nem tartalmazó sorok
  - -i kis és nagybetűket nem különböztet meg
  - -w Csak önálló szóként találja meg (traPista nem)
  - -r Rekurzívan a paraméterül adott könyvtárra.
  - -l Csak a file neveket írja ki. (fájlban keres)
  - -c csak a sorok számát írja ki
  - -n megszámozza a sorokat

zoltan.illes@elte.hu

Grep – mintaillesztés II.

- Példa:
  - cat nevsor|grep Pista # Eredményül kapjuk a Pista-t tartalmazó sorokat.
  - grep -r ‘fradi' ./script # script könyvtárban a fradi-s sorokat (fájlokban) keresi
  - grep –r –l par * # Az összes állományban, alkönyvtárakban keresi a par szót, eredményül csak a fájl nevét írja ki.
  - cat param|grep ‘or’ # a or-t keres

zoltan.illes@elte.hu

Grep – mintaillesztés III.

- Egy szövegminta általános megadása – Reguláris kifejezések-speciális karakterek
  - ^ Sor elejétől kell egyezni a mintának.
    - Pl: ‘^alma’ : a sor elején alma szó áll
  - $ Sor végétől kell egyezni a mintának.
    - Pl: ‘barack$’ : a sor végén a barack szó áll
  - . Egy tetszőleges karakter
  - * Előző minta ismétlése 0 vagy többször!
    - Pl: ‘^alma.*fa$ ‘ - alma és fa között akárhány(0 is lehet) karakter

zoltan.illes@elte.hu

Grep – mintaillesztés IV.

- Karakterhalmazok megadása: [..]
  - [a-z] (kisbetű)
  - [^a-z] Nem kisbetű
  - [A-Za-z0-9] Alfanumerikus (szám vagy betű)
    - \w Alfanumerikus, mint előző
    - \W Nem alfanumerikus
  - \d számjegy, azonos a [0-9]-el
  - \s szóköz, tab, sortörés
- Szavak illesztése
  - \< Szó kezdet, PL: grep ”\<Fer”
  - \> Szó vég

zoltan.illes@elte.hu

Grep – mintaillesztés V.

- egrep , fgrep– bővített grep -E, fixed, grep -F
  - ali|éva , vagy kapcsolat,
  - Példa: ls | egrep ”par|pelda”
  - \+ Előző minta legalább egyszer
    - Példa: ls | egrep ”\<p+f” # file elején 1 vagy több p, majd f betű.
  - ? Előző minta nulla vagy egyszer ismétlődik
    - Példa: ls|egrep „param\d?” #param, param1, param2,..
  - {n} előző karakter pontosan n szer!
    - Példa: cat almafa|egrep ^[a-d]\w{5}

zoltan.illes@elte.hu

Grep – mintaillesztés VI.

- {2,4} Előző minta 2,3 vagy 4-szer ismételve
  - {1,} Előző minta legalább egyszer
- () Egy csoportba fogunk egy mintát.
  - Ismétlődéshez célszerű, azaz ezt követi egy ismétlésre vonatkozó utasítás +,?,{n}
  - Példa: [0-9]{8}(\s[0-9]{8}){1,2} #bankszámla
    - A [0-9] helyett \d is jó lenne.
- Speciális karaktert célszerű \ mögé írni.
  - Példa: ^[+-]?\d+([,\.]\d+)? #előjeles szám tizedes ponttal \., vagy vesszővel
- Többi lehetőséghez: man

zoltan.illes@elte.hu

Köszönöm a figyelmet!

zoltan.illes@elte.hu