Számítógépes alapismeretek

8. előadás

zoltan.illes@elte.hu

Illés Zoltán ELTE IK

Ami eddig volt…

- Számítógépek architektúrája
- Alapvető alkotóelemek
- Hardver elemek
- Szoftver
  - Gépi kódtól az operációs rendszerig
  - Unix alapok
  - Shell script I.
    - Változók, reguláris kifejezések
    - Elágazások, ciklusok

zoltan.illes@elte.hu

Ami ma következik…

- További shell script programozási elemek
  - sed
  - awk
- Webes publikálás
  - Hitelesítés kérése
- Unix-Linux management alapok

zoltan.illes@elte.hu

sed utasítás

- Szűrő, Stream Editor, adatfolyam szerkesztő
- Feladata: Komplex behelyettesítések, cserék a szabványos bemenetre érkező sorokon, eredmény a szabványos kimeneten. Veszi az összes sort (ciklus), majd minden egyes soron a parancsai végrehajtódnak, a módosított sor a kimenetre kerül!
- Példa: cat osztaly| sed ‘s/3/9/g’
  - Keressük meg a sorokban az összes (g) 3-t, majd ezt cseréljük ki 9-re.

zoltan.illes@elte.hu

sed parancs opciók

- -n, Nincs automatikus kiírás a szabvány kimenetre. Csak a sed script kiíró utasításai írnak a kimenetre.
- -f scriptfile, A paraméterül megadott fájlban van a program, a sed script. Jellemzően egy sorba egy parancsot írunk, de a ; elválasztóval több parancs is kerülhet egy sorba.
- -e A sed parancssorában több script van. Egy esetén elhagyható.

zoltan.illes@elte.hu

sed script utasítások

- Több parancs megadása közvetlen script parancsban:
  - ; Pontosvessző a parancsok között!
    - Példa: cat osztaly|sed ‘s/3/9/g ; s/9/21/’ # Cseréljük az osztály soraiban az összes 3-at 9-re, majd az első kilencest 21-re minden sorban!
  - Külön sorba írhatók a parancsok, másodlagos promptot kapunk!

```
$ cat osztaly | sed ‘s/3/9/g
>s/9/21/’ [enter]
zoli 21 4 2 5 4
feri 2 21 4 5 9
juli 4 21 2 9 9
$
```

zoltan.illes@elte.hu

sed fontosabb parancsok I.

- Behelyettesítés: [cím] s /minta/új_minta/[jelző]
- Jellemzően a minta reguláris kifejezés, ezt keressük.
- A cím is lehet reguláris kifejezés.
- A jelző értékei:
  - n: 1 és 512 közti szám, a cserét az első n mintán kell elvégezni, ha elmarad, n=1. $ utolsó sor!
  - g: Az összes mintát le kell cserélni.
  - p: Kilistázza az aktuális sort! (pg együtt is lehet)
  - w fájl: Menti fájlba az aktuális sort!

zoltan.illes@elte.hu

sed fontosabb parancsok II.

- /új_minta &/ Új mintát a minta elejére teszi
  - Pl: echo fradi|sed 's/fradi/hajra &/‘ #hajra fradi
- \n használata,
  - \1 az 1. regurális kifejezés,
  - \ elnyomja a következő metakarater hatását(\.ali)
  - \ enter, soremelés beszúrása (\ >\2/’)

```
illes@panda:~$ echo .ali 4 Ali baba|sed 's/\.ali \([0-9][0-9]*\) \(.*\)/\1. rész\
> \2/‘ (enter)
4. rész
Ali baba
illes@panda:~$
```

zoltan.illes@elte.hu

sed fontosabb parancsok III.

- Törlés: [címtartomány]d
  - d parancs törli a címtartomány sorait
  - Pl: cat osztaly|sed '1d; s/3/9/pg; s/9/21/‘ #első sort törli, majd a többi soron a másik két parancs
- Hozzáfűzés: a
  - Pl: cat osztaly|sed 'a\alma‘ # új sorként az alma, minden sor után
- Beszúrás:
  - Pl: cat osztaly|sed '2,3i\alma‘ # 2,3 sor elé alma

zoltan.illes@elte.hu

sed fontosabb parancsok IV.

- És még sokan mások…

zoltan.illes@elte.hu

sed leírás

- Google: sed
- http://www.szabilinux.hu/ufi/tartalom.html
  - 7. fejezet: sed
- http://www.gnu.org/software/sed/manual/sed.html
  - GNU teljes sed leírás.

zoltan.illes@elte.hu

Mi van a sed után? AWK

- A sed nagyon jó, nagyon hasznos, de …
  - Jellemzően mintacserére használható.
  - Sorokon belüli tevékenység korlátozott.
  - Nincs aritmetikai lehetőség. (Valami van csak nem az igazi…)
  - Vezérlési szerkezetek hiányoznak. (Van ugró, feltételes ugró utasítás…,)
  - …
- A kiút: AWK

zoltan.illes@elte.hu

AWK

- Alfred V. **A**ho, Peter J. **W**einberger, Brian W. **K**ernighan
- Schell hiányosságai szövegfeldolgozáskor
- Gyakorlatilag C nyelvi lehetőségek
- Tipikus szűrő
- Gyakran schell script elemként használt
- Soronkénti szövegkezelés,
- Minden soron végrehajtódó program
- awk –gawk (GNU AWK)
  - http://www.gnu.org/software/gawk/manual/gawk.html

zoltan.illes@elte.hu

AWK program helye, program szerkezet

- whereis awk # /usr/bin/awk, …
- Az awk vagy paraméterként vagy a szabvány bemenetén várja az átdolgozandó adatokat.
- Soronkénti feldolgozás. Első sor előtti, utolsó utáni inicializálási blokk.
- A parancsblokkok {} jelek közti utasítás
- Parancsblokk előtti minta: Példa: /f.*/
  - /reguláris kifejezés/
  - A minta egy logikai kifejezést tartalmaz: $2 == „alma”

zoltan.illes@elte.hu

AWK használata

- Program, közvetlenül mintegy paraméter
  - awk ‘{ print ;}’ adatfile
    - A program minden sorra vonatkozik, kiírja azt
- File-ban a program
  - awk –f programfile adatfile
  - Helyette gyakran az awk programfile a parancs
    - #!/usr/bin/awk –f
    - Ez az első sor parancsa.
    - pl: $ awk_program1 adatfile # jellemző parancs alakja
- Szűrőként
  - Parancs1 | awk-parancsfile

zoltan.illes@elte.hu

Bemeneti sorok elemei

- $0 – a teljes sor
- $1, $2, … - a sor első, második eleme
- Mezőelválasztó: FS (alap: helyköz vagy tab)
  - Lehet több karakter is: FS=„abc”
- Egy sor mezőinek száma: NF
- Sorelválasztó karakter: RS (alap: újsor)
- Az eddig beolvasott sorok száma: NR
  - Több bemeneti fájl esetén: FNR, egy fájl sorszáma

zoltan.illes@elte.hu

AWK példa

- Programkód:

```
#!/usr/bin/awk -f
# Kezdő blokk!
BEGIN {
 print "Kezdem a programot!";
 # ide jöhetnek a tovabbi kezdő, először
 # végrehajtandó parancsok, pl mezőelválasztó
 FS=":"
}
# Minden sorban keresi a feri-t, ha találja kiírja
/feri/ {
 print "A Feri sor tartalma: "$0;
}
# Befejezés
END {
 print "Itt a vége!";
}
```

- Futtatás:

```
$cat osztaly|elsoawk
Kezdem a programot!
A Feri sor tartalma: feri 2 3 4 5 3
Itt a vége!
$
```

awk változók, kifejezések

- Beépített változók nagybetűsek! Pl: NF
- Nincs típus: név=érték, érték:szám, „szöveg”
  - v=”alma”; print v; #alma
- Szövegösszefűzés: nincs operátor, egymás után kell írni a változókat!
  - Pl: {v="alma";f="fa"; print ”5 ”v f} #5 almafa
- Konverzió automatikus, ha úgy érzi, hogy kell:
  - {v="alma";f="fa";print v+f} #0
  - {v="3alma";f="2fa"; print v+f} # 5

zoltan.illes@elte.hu

awk tömbök

- t[0]=3, stb. megadjuk az index elemeket.
- Tömb elemei különböző típusúak lehetnek.
- Valójában asszociatív tömbök:
  - t[„egy”]=1; print t[„egy”];
- Tömb hossza: length(t)
- Egy index bent van-e a tömbben: 4 in t
- Elem törlése: delete t[4]
- Többdimenziós tömbök: tt[1,2]=3;

zoltan.illes@elte.hu

awk matematikai operátorok, függvények

- +, -,++,--, *, /, %,… - szokásos műveletek
- Logikai érték mint C-ben
- !=,==, <,> - logikai operátorok
- && logikai és, || logikai vagy, ! tagadás
- ~, !~ minta illeszkedés, nem illeszkedés
  - A jobb oldali operandus lehet reguláris kifejezés is /…/.
  - A $0 ~ /reg.kif/ alak rövidítve: /reg.kif/ , emiatt szerepel az awk blokkok előtt csak így!
- ** vagy ^ hatványozás, pl: 2**3 #8

zoltan.illes@elte.hu

awk matematikai függvényei

- Fontosabb beépített függvények:
  - int(szám) # egészrész , print int(3.7) #3
  - sqrt(szám) #négyzetgyök
  - sin(x), cos(x) #
  - rand() #]0,1[intervallumban véletlenszámot generál
    - x=int(10*rand()) # 0,1,…9
  - exp(x), log(x) # e ** x, ln(x)
  - atan2(x,y) # arc. tangens x/y

zoltan.illes@elte.hu

Egyéb hasznos awk függvények

- {v=length(”almafa”);print v} #szöveg hossza
- split – szöveg darabolás
  - Pl: split(„ali:pali:robi”, n, „:”); print n[1] #ali
- sprintf(sz, „minta”,változók), mint C-ben
- system(„parancs”) – op. rendszer par. futtatás
  - Pl: {system(”date >datum”) }
- File olvasás:
  - getline <"/home/adat”; print $0;
  - Következő getline, következő sor.

zoltan.illes@elte.hu

awk vezérlési szerkezetek

- Elágazás
  - if (x % 2 == 0) print "x páros,,; else
    print "x páratlan"
- Ciklusok, mint C-ben
  - while (kif) utasítás
  - do … while (kif)
  - for(kif1;kif2;kif3) utasítás
  - for (i in tömb)
    tömb[i] feldolgozása

zoltan.illes@elte.hu

AWK összegzés, példa

- Help: Google: awk, gawk
- BEGIN blokk, a soronkénti feldolgozás előtt hajtódik végre
- END blokk, a soronkénti feldolgozás után hajtódik végre
- Minta {soronkénti blokk}
- Példa: user.awk

zoltan.illes@elte.hu

Unix-Linux boot

- Boot sector – MBR
  - LILO op. Rendszer választás
- Kernel betöltése
- Init processz indítása (/sbin/init)
  - /etc/inittab konfigurációs állomány
    - Alapértelmezett futásszint beállítás
      - Futásszintek 0- rendszeráll, 1- single user mód, 2,3,4,5- rendes szintek, 6- reboot
      - A 2-5 szinteket minden rendszer sajátosan definiálja
    - Mi a teendő az egyes futásszinteknél
      - /etc/init.d/rc szint
    - GETTY indítás

Szerver elérés

- IP konfiguráció
  - Parancssori lehetőség
    - Ifconfig, vagy újabban ip parancs (eth0, lo(opback))
  - Adminisztrációs felület
    - SUSE- yast, IBM-smit, stb.
- Terminál elérés
  - Ssh
  - FTP
    - FTP over SSL
- Webes elérés

zoltan.illes@elte.hu

Web publikáció, hitelesítés

- public_html könyvtár
  - Módosítható, a httpd.conf állományban
  - Rendszergazda (root) jogosítvány
  - Ha egy könyvtárban van index.html (def. dokumentum), akkor azt adja vissza.
  - Ha nincs index.html, akkor mintegy ftp tartalomjegyzék!
- Publikus minden, mindenkinek!

zoltan.illes@elte.hu

Hitelesítés, jelszó védelem

- Adott könyvtárra érvényes, ha .htaccess file létezik a könyvtárban (speciális forma)
- htpasswd, basic, kódolás nincs, apache2 esetén htpasswd2 a név
  - szamalap.inf.elte.hu gépen: /usr/bin könyvtárban
  - Használat: htpasswd [-c] filenév usernév
    - -c filenév új állomány lesz
    - Megkérdezi a jelszót, majd a névvel együtt a file-ba rakja kódolva a jelszót
    - Példa: letolt könyvtár
- htdigest, MD5 kódolás (apache2 esetén htdigest2)
  - Használat: htdigest [-c] filenév azonosító usernév
  - Bőngészőfüggő lehet (IE ?, FireFox ...)

zoltan.illes@elte.hu

Apache irodalom

- Apache Web szerver konfigráció
  - /etc/httpd.conf
  - Vagy újabban: /etc/apache2/ és itt sok kis conf.
- http://httpd.apache.org/docs/1.3/howto/htaccess.html
- Google
- Könyvesbolt

zoltan.illes@elte.hu

.htaccess tartalom

- AuthType Basic
- AuthName "Gyumolcsfa gyujtemeny"
- AuthUserFile /usr/people/illes/public_html/letolt/alma
- Require user alma
- Order deny,allow # sorrend: tiltás, engedélyezés
- Deny from all
- Allow from elte.hu
- allow from 81.82.83.94
- Satisfy any # valamelyik kritériumnak igaznak kell lenni ahhoz, hogy hozzáférésünk legyen a tartalomhoz

zoltan.illes@elte.hu

Unix-Linux management alapok

- Központi management programok
  - SMIT, YAST,YAST2
- Kézi módosítás
  - /etc könyvtár
  - /etc/hosts, /etc/passwd, /etc/shadow, /etc/services
  - /etc/resolf.conf, /etc/sendmail.conf
  - /etc/inetd.conf Internet server konfiguráció
  - /etc/sendmail.cf
  - /etc/httpd.conf

zoltan.illes@elte.hu

Felhasználó management

- Központi adminisztrációs eszköz
  - Létezik minden Unix rendszeren
  - Kényelmes, könnyű használat
  - Hátrány: nagy létszámnál nehézkes, lassú
  - Megoldás: script használat
- Kézi módszer
  - /etc/passwd, /etc/shadow kézi módosítása
  - Létezik adduser, useradd vagy hasonló nevű parancs.
- Példa: useradd.awk (webprogramozas)

zoltan.illes@elte.hu

Apache config- Virtuális gép beállítás

- Httpd.conf
  - Általában /etc könyvtárban
  - Webprogramozáson /etc/apache2 könyvtárban
    - (Suse linux) Nem egy fájl, hanem szét van szedve funkcionalitás szerint.
  - SSI, CGI jogok
    - Könyvtárra, .shtml kiterjesztés
    - Mod_userdir.conf
  - Virtuális könyvtár
    - Vhosts.d könyvtárban, adott nevű .conf állományok

zoltan.illes@elte.hu

Köszönöm a figyelmet!

zoltan.illes@elte.hu