Számítógép Architektúrák

4. Gyakorlat

Téma

- Levelezés.
- Adatcsatornák, átirányítás.
- Visszatérési érték. -> Vezérlésben fontos.
- Metakarakterek, kvótázás.
- Parancs, csővezeték, parancslista.
- Jobkezelés, Szűrők

Levelezés a tanszéken

- Mindenkinek van saját E-mail címe!
- **felhasználónév**@iit.uni-miskolc.hu

Levelezés a tenszéken

- Több mód van rá, hogy kezeljük levelezőfiókunk.

1.: webmail.iit.uni-miskolc.hu

(sajnos hitelesítési hiba van, de semmi nem lehet tökéletes…)

2.: Külső levelező klienssel. Bővebben a szolgáltatások doksiban.

Levelezés a tanszéken

- 3.: Linux programok:
  pine, mutt, mail …

$ mail --help
$ mail -s ‘tárgy’ cím@cim.hu [enter]
Ide jön az üzenet szövege, ha kész…
CTRL+D [enter]
cc: {vagy megadod, vagy nem}

pine

- Levelező program. CUI.

- c – levél küldés
- m – főmenübe lép
- q – kilépés

- COMPOSE MESSAGE -> üzenet írása
- FOLDER LIST -> bejövő üzenetek

mutt

- Levelező program. CUI. Színes, csicsa …

- Q – kilépés
- M – levélírás
- Y – levélküldés
- G – bejövő levelek megnézése
- Levél szövegének írására vi – t használ

A burok processz

- Önálló entitás, azonosítója a **pid** (process identification number)
- A /bin/sh (vagy /bin/bash) program fut benne
- Van 3 nyitott adatfolyama

  – A 0 leírójú stdin (szabványos bemenet), ahonnan a parancsokat, csöveket, parancslistákat olvassa.

  – Az 1 leírójú stdout (szabványos kimenet), ahová az eredményeit írja.

  – A 2 leírójú stderr (szabványos hibakimenet), ahová a hibaüzeneteit írja.

Szabványos csatornák

Kifejezés	Rövidítés	Fájl-leíró	Szabványos eszköz
Szabványos adatbevitel	stdin	0	Billentyűzet
Szabványos adatkivitel	stdout	1	Konzol
Szabványos hibaüzenet	stderr	2	Konzol

- A szabványos adatbevitel és adatkivitel elvét a UNIX-ban találták ki. A shell lehetővé teszi a szabványos csatornák átirányítását a ,,>'' és a ,,<'' (kacsacsőr-jelek) segítségével.
- Parancsok végrehajtásának eredményét a következő módon tudjuk fájlba irányítani:
- *zsiga@debian:~$* ***ls >lista***

Szabványos csatornák

- Így az ***ls*** parancs kimenete nem jelenik meg a képernyőn, hanem beleíródik a *lista* nevű fájlba. Ebben az esetben, ha nem létezett a *lista* nevű fájl, akkor létrejön, ha létezett, akkor felülíródik. Abban az esetben, ha nem felülírni, hanem hozzáfűzni szeretnénk egy létező fájlhoz, akkor a következő formában kell kiadnunk a parancsot:
- *zsiga@debian:~$ ls >>lista*
- Azok a parancsok, amelyek a szabványos adatbeviteli csatornáról várnak adatokat, az átirányítás révén az adatbevitelt fájlból is megkaphatják:
- zsiga@debian:~$ cat <lista

- Lehetséges a szabványos hibacsatorna átirányítása is. Erre például abban az esetben lehet szükségünk, ha a háttérben indítjuk el egy program futását, de látni szeretnénk, ha valami probléma támadt a futása közben.
- *broda@debian:~$ ls > lista 2> error*
- Tehát először az *ls* parancs eredményét átirányítjuk a *lista* nevű fájlba, ugyanakkor a hibaüzeneteket átirányítjuk az *error* nevű fájlba. Több csatornát is átirányíthatunk egyszerre.
- A cső (pipe) segítségével átirányíthatjuk az egyik parancs kimenetét egy másik parancs bemenetére.
- broda@debian:~$ who | sort

- cat > lista //szerkeszteni lehet a lista fájlt (ctrl+c)

- ls -l > lista.txt
- ls >> lista
- cat lista.txt
- //szóköz bármennyi lehet 0-…

- echo ”Szöveg1” > proba1.txt
- cat proba1.txt
- echo ”Szöveg2” > proba2.txt
- cat proba2.txt
- cat proba1.txt proba2.txt > proba.txt

Visszatérési érték

- Lehet normális (0),
- Lehet nem normális (nem 0), ennek oka többféle:
  – valami hiba van,
  – nincs hiba, de szemantikailag van gond.

(Pl. grep szűrő nem talál minta-egyezést, vagy test parancs tesztelése nem igaz.)

A visszatérési értéket a programvezérlésben használhatjuk majd.

Visszatérési érték példa

- $ test –f .bash_history ; echo **$?**
- > 0

- $ test –d .bash_history ; echo **$?**
- > 1

- **$?** -> belső változó!

Metakarakterek (reguláris kifejezések)

tömören írhatunk le vele általános karaktermintákat,

„.” vagy „?”	bármely 1 karakterre illezkedik
„*”	0 vagy több bármilyen karakterre illeszkedik
„+”	a megelőző kifejezés egyszeri vagy többszöri előfordulására illeszkedik (a zeuson nem működik)
„[..]”	zárójelbe tett karakterek egy, a zárójelben megadott karakterre illeszkednek; pl: [0-9a-z] – az összes számjegyre és a kisbetűkre illeszkedik (angol abc)
„[^..]”	bármely egy karakterre illeszkedik, ami nincs felsorolva a zárójelben; pl: [^0-9] – bármely egy karaktert helyettesíthet, mely nem szám.
„^”	sor elejére illeszkedik
„$”	sor végére illeszkedik

Metakarakterek semlegesítése (quotázás):

echo ’*’
echo ”*”
echo *
echo ”’” vagy echo ’”’

Idézőjelet a másik típusú idézőjellel tudunk semlegesíteni.

*$ echo **
*$ echo **

Ismételjük: a parancs fogalma

Fehér karakterekkel határolt szavak sora

- első szó a parancs neve,
- többi szó az argumentumok.

Az sh beolvassa, értelmezi, átalakítja, végrehajtja

- saját maga (belső p.),
- gyermek processzben (külső p.)

Mindkét esetben van visszatérési értéke!

Vannak szabványos adatfolyamok!

A csővezeték fogalma

A csővezeték (pipe) parancsok sora | operátorral összekötve:

parancsbal | parancsjobb

Szemantikája: végrehajtódik a parancsbal, szabványos kimenete egy csőbe képződik, majd végrehajtódik a parancsjobb, aminek szabványos bemenete erre a csőre képződik.

A cső visszatérési értéke: a parancsjobb visszatérési értéke.

A parancs degenerált cső. Példa: $ ypcat passwd | grep kovacs

A parancslista

Csővezetékek sora listaoperátorral összekötve:

csőbal op csőjobb

Listaoperátorok:

&& || # magasabb precedencia, de alacsonyabb mint a |

& ; \n # alacsonyabb precedencia

A szemantika:

; \n soros végrehajtása a csöveknek

& aszinkron végrehajtás (csőbal háttérben)

&& folytatja a listát, ha csőbal normális visszatérésű

|| folytatja a listát, ha a csőbal nem normál visszatérésű

Parancslisták

A lista visszatérési értéke az utolsó cső visszatérési értéke.

Háttérben futó cső visszatérési értéke különlegesen kezelhető.

A cső degenerált lista (ahol ezentúl listát írunk, írhatunk csövet, sőt parancsot is!)

A && és || operátoros listáknál először láthatjuk a visszatérési érték értelmét! Valóban a vezérlés menetét befolyásoljuk!

Példák

```
$ cd  ide  &&  rm junk    # csak akkor töröl, ha ...

$ ls  ide  ||  cp  valami  ide    # ha nincs ide,
   készíti

$ ( mv  a  tmp  && mv  b  a )  && mv  tmp  b
```

Processzek (job, taszk, folyamat)

sort < lista >eredmény //a shell végrehajtja, majd megjelenik a prompt

sort < lista >eredmény **&** //a shell nem várja meg a program befejeződését, hanem megjelenik a folyamat PID-je és a prompt

háttérprocessznek nem interaktív programok alkalmasak, amik nem olvasnak a billentyűzetről és nem írnak a képernyőre;

egy háttérprocessz kap egy jobszámot és egy processzazonosítót;

ne futtassunk a háttérben olyan processzt, mely interaktív, mert összekuszálja a képernyőnket

Job vezérlés

- ^z (ctrl+z) fel lehet függeszteni vele egy processz futtatását **fg** (foreground) paranccsal lehet előtérbe hozni
  pl: mcedit ->^z; cat lista -> ^z //2 felfüggesztett jobunk van

- **jobs** - a felfüggesztett jobokat lehet kilistázni vele
- jobs -l kírja a PID-et is
- fg [%job] – újra fut a processz (%1, %mcedit)
- fg esetén az aktuálisat a „+”- al megjelöltet fogja indítani, amit utójára függesztettünk fel
- **bg** paranccsal egy felfüggesztett jobot lehet a háttérbe küldeni. (akkor jó ha egy processzt normál módon indítottunk és később felfüggeszthetjük és a háttérbe küldhetjük)
- **ps**
- **kill** -9 szignált küld

Szűrők

- Adatot olvas, valamilyen műveletet hajt végre rajta, majd az eredményt kiírja a kimenetre.

cat

- legegyszerűbb szűrő; olvassa az adatot nem hajt végre rajta semmilyen műveletet és kiírja az stdin-re.

- cat lista
- cat > lista
- cat lista1 lista2 > lista

head

- **head -x** a bemeneten kapott szöveg x számú első sorát írja ki.

- $ ls > lista; cat lista | head –5

tail

- az opciónak megfelelő számú sort ad vissza a beolvasott adat végéből

$ cat lista | tail +3 //a lista fájl tartalmát a 3. sortól a végéig kiírja

$ cat lista | tail -4 //a lista fájl utolsó 4 sorát írja ki

$ cat lista | tail -3 | head -1 //a lista fájl 3. sorát

more

- Lapozó szűrő. Ilyen még a less és a pg is.

$ more lista

$ cat lista | **more**

$ cat lista | **less** //q-val lehet kilépni

grep

- Mintát keres a megadott szövegben.
- grep [kapcsolók][minta][file]
- visszatérési értéke:
  0 talált mintát
  1 nem talált mintát
  >1 error

grep

kapcsolók:

-c -- azoknak a sorok számát adja vissza, melyekben megtalálta a mintát;

-i/y -- nem különbözteti meg a kis és nagybetüket;

-n -- kiírja, hogy hányadik sorban találta meg a mintát;

-l -- ha több fájlban is keresünk, kiírja, hogy melyikben találta meg a mintát;

-w -- csak teljes szavakat keres;

-v -- azokat a sorokat adja vissza, melyek nem tartalmazzák a mintát;

ls | grep list* //kiírja a munkajegyzékben található list kezdetű fájlokat

grep –l m lista* //azon fájlok nevét írj ki, amelyekben megtalálta a mintát

sort

- Rendezi az állományt.

sort -u :az azonos sorokból csak egyet hagy meg

sort -o lista :lista -rendezette teszi a lista állományt

sort -r lista :visszafelé rendezi a lista állományt

sort -f lista :hatására a kis és nagybetűket egyformán fogja kezelni

sort lista

wc

- Karaktereket, sorokat és szavakat lehet vele számolni.

wc -c lista -megszámolja, hogy hány karakter van a lista fájlban

wc -l lista -sorokat

wc -w lista –szavakat

pl: who | wc –l //hány felhasználó van bejelentkezve

uniq

- **rendezett adatokon** végez szűrést az opcióknak megfelelően.

uniq -c lista -kirja, hogy a melyik sor hányszor szerepel a fájlban

sort -o lista lista; cat lista.txt | uniq –c
//megszámolja, hogy melyik sor hányszor szerepel benne

-

cut

- Oszlopot vág ki a szövegből.

- cut -c1,3-4 lista -a lista fájlból csak a 1,3,4.oszlopokat írja ki.
- rwho | cut -c1-8
- cut –f4,5 –d: /etc/passwd

//a /etc/passwd állományból a 4,5 oszlopot fogja kiírni, mivel a mezőelválasztó karakter ‘:’

tr

- char1 -et char2 -re cseréli

cat lista | tr lista korte //a lista karaktereit a korte megfelelő karaktereire cseréli

cat lista | tr -c A-Za-z @ //ami nincs benne a megadott karakterek között, azt @-ra cseréli

cat lista | tr -c a-df-z @

cat lista | tr -s a f //akárhányszor szerepel az ’a’ mindig csak ’f’-t ír helyére

cat lista | tr -d al //törli a lista-ból az ’al’ karaktereket

rev

- fordítva írja ki soronként a fájl tartalmát

- cat lista | rev

tee

- Csővezeték szétágaztatása.

cat lista | tee lista1 | more

cat lista | tee lista1 lista2 | more

cat lista | tee lista1 lista2 | head –10

cat lista | tee lista1 | head -10 | tee lista4 | tail –3 | cat