Számítógép Architektúrák

8. Gyakorlat
AWK 1, CSS

awk

- Az **awk** egy általános célú programozási nyelv, amelyet szöveges állományok földolgozására terveztek. Elnevezése a megalkotói – Alfred **A**ho, Peter **W**einberger és Brian **K**ernighan – családnevének kezdőiből született.
- Ideális szöveges állományok szűrésére, átformálására, kiértékelésére. Ma is minden unix rendszeren van legalább egy awk változat.
- A Free Software Foundation a gawk nevű változatot gondozza. A linux disztribúciókkal jellemzően a gawk implementációt szállítják. Windows rendszerekhez is letölthető a gawk például a Cygwin részeként.

Mintakereső és feldolgozó eszközök

program	leírás
grep	soronként végignézi a fájlokat és a mintával egyező sorokat keres kapcsolókkal, opciókkal lehet a működését befolyásolni
sed	szerzője: Lee McMahon (sed=stream editor) programozható szövegszerkesztő, amely egybetűs parancsok és szabályos kifejezések segítségével vezérelhető a mintát és a tevékenységet is általánosítja vesz egy (szerkesztőparancsokból összeállított) "programot" és a fájl minden sorára végrehajtja
awk	C jellegű programnyelv (néhány szerkezet hiányzik belőle, mások pedig eltérnek tőle) kevésbé alkalmas a szöveg átalakítására, mint a sed tartalmaz aritmetikai műveleteket, változókat, beépített függvényeket

awk használata

$ awk 'program' fájlnevek
$ awk -f parancsfájl fájlnevek
$ kimenet | **awk** ...

- Egy tipikus awk program a végrehajtása során a bemeneti adatokat egy másféle kimenetté formálja át.
- A programok általában mintából és a mintához tartozó parancsokból állnak:

a ’program’ az alábbiakból áll:

minta (tevékenység)

minta (tevékenység)

…

Az awk soronként olvassa a bemenetet. Minden beolvasott sort összehasonlít a mintákkal, és ha illeszkedést talál, a parancsokat végrehajtja. A mintákat a szabályos (reguláris) kifejezések szabályai szerint értelmezi.

A bemeneti fájlokat nem változtatja meg. A tevékenység rész egy C szerű utasítás(blokk).

A különleges parancsformák:

BEGIN {
parancs(ok)
}
adatbeolvasás **előtt** ezeket a parancsokat végrehajtja
{ minta (tevékenység) }
END {
parancs(ok)
}
adatbeolvasás és a többi parancs végrehajtása **után** ezeket a parancsokat végrehajtja

- **awk ’/regularis kifejezés/’** fájlnevek
  kiírja a mintára illeszkedő sorokat.

- **awk ’(print)’** fajlnevek
  kiírja a fájl(oka)t (minden sort)
  Tulajdonképpen ugyanazt csinálja, mint a **cat**, de sokkal lassabban.

- **awk -f parancsfájl** fájlnevek
  Az awk parancsnak is megadhatjuk a programot fájlból. Ebben az esetben a -f opciót szükséges használni
  parancsfájl első sorába be kell írni: **#! /bin/awk**

Működése

- Az awk minden bemenő sort mezőkre bont.
- **mező**: szóközökkel, vagy tabulátorokkal elválasztott, szóközt nem tartalmazó karakterlánc.
- **mezőelválasztó**: (default) az awk a fehér (üres) karaktereket (szóköz, tabulátor)tekinti, de meg lehet adni neki valamely más karaktert (de csak 1 karaktert!) is mezőelválasztónak (FS).

Mezőelválasztó

```
$ awk -F: '{print $2}' /etc/passwd
$ awk 'BEGIN {FS=":" }{print $2}' /etc/passwd
```

- a beolvasott minden sor mezőit rendre az 1,2,3, …., NF változóba teszi.
- ha a mezők tartalmára szeretnénk hivatkozni, akkor azt, a $1, $2, …. tehetjük meg
- **$NF**: az aktuális sor utolsó mezőjének értékét tartalmazza,
- **$0**: az aktuális teljes sort tartalmazza,

- az awk esetén a shellel ellentétben csak a mezők neve elé kell $ jelet tenni, ha annak tartalmára szeretnénk hivatkozni.

```
$ who | awk '{print $3, $4, $5, $1, $2}'
```

Az awk beépített változói

Változó	Jelentése
FILENAME	az aktuális bemeneti fájl neve
NF	bemeneti rekord (sor) mezőszáma
NR	bemeneti rekordok száma
FS	mezőelválasztó karakter (default: tab és space)
RS	bemeneti rekordelválasztó karakter
OFS	kimenő mezőelválasztó karakter/lánc/ (default. szóköz)
ORS	kimeneti rekordelválasztó karakter/lánc/ (default: újsor)
OFMT	kimenő számok formátuma (default: %g lásd printf(3))
$0	a feldolgozás alatt álló egész sor
$1, $2, ... $n	a sor egyes, egymástól szóközzel elválasztott részei.

Megjegyzés

- Az **NF** tehát az aktuális sor **mezőszám**át adja meg, a **$NF** pedig az aktuális sor utolsó mezőjének **érték**ét.
- A shell script-ben és az awk-ban is vannak beépített változók ($0, $1, $2, …) de ezek különbözőek, mást jelentenek !!!

awk további használata

$ awk '{print NR, $0}' lista

kiírja a lista állomány tartalmát, úgy, hogy a sorokat „számozza”

$ cat lista | awk '{print NR, $0}'

$ who | awk ’BEGIN {OFS=”@”}{print NR,$2,$3,$4}’

```
BEGIN {
  OFS=„@”
}
{
  print NR, $2, $3, $4
}
```

az egyes sorokat sorszámozva írja ki, a sorokból csak a 2., 3., 4. mezőt írja ki, és a mezőelválasztó karakter „@” lesz

OFMT

```
$ awk 'BEGIN {
  OFMT = "%.0f" # print numbers as integers
  (rounds)
  print 17.23, 17.54 }‘

17 18
```

printf

- A printf utasítással formázott kiíratást valósíthatunk meg. Ez olyan, mint a C nyelv ilyen nevü függvénye.

```
$ awk ’{printf "%4d %s\n", NR, $0 }’ file002
```

Változók, aritmetika

- A változókat azzal definiáljuk, hogy használni kezdjük. Alapértelmezés szerint a változók kezdőértéke 0, így nem kell törődnünk az inicializálásukkal.
- Aritmetikai műveletek elvégzése is könnyebb, mint az „expr” shell parancs.

Az awk változói karakterláncokat is tárolhatnak

- A szövegösszefüggéstől is függ, hogy a változót numerikus vagy sztring típusú változóként kezeli-e a program.

  s = s +1 számértéket használ a program

  s=”abc”’ karakterérték

  x>y karakterértéket használ, kivéve, ha az oparandusok nyilvánvalóan numerikusak

- Az awk-ban használhatók a C nyelvben használatos aritmetikai rövidítéseket is:

  s += $1; s = s + $1;

 # ls | awk '{print NR,$0}' > lista1

$ awk '{s = s + $1} END {print s}' lista1
az első oszlop számainak összegzése

$ awk '{s = s + $1} END {print s/**NR**}' lista1
az első oszlop számainak átlaga

$ wc lista | awk '{print $3/$1}'
egy fájl átlagos soronkénti karaktereinek száma

wc parancs megvalósítása awk-val

```
{
  nc += length($0)+1
  nw += NF
}
END {print NR, nw, nc}
```

Az awk operátorok növekvő precedenciarendben

Operátor	Hatása
=, +=, -=, *=, /=, %=	értékadás (v op=kif ugyanaz. mint v = v op (kif))
||	vagy/or/ (k I II k2 igaz. ha valamelyik igaz; k2-t nem értékeli, ha k1 igaz)
&&	és/and (k1 && k2 igaz, ha mindkettő igaz; k2-t nem értékeli. ha k1 hamis)
!	negálás
>, >=, <, <=, ==, !=, ~, ! ~	relációs operátorok
semmi	karakterlánc összekapcsolása
+,-	összeadás, kivonás
*,/, %	szorzás, osztás, maradék
++, --	Növelés, csökkentés (elöl vagy hátul)

Minták

$1 == „” az 1. mező üres
$1 ~ /^$/ az 1. mező megegyezik az üres karakterlánccal
$1 !~ /./ az 1. mező semmilyen karakterrel nem egyezik meg
length($1) == 0 az 1. mező hossza nulla

~ reguláris kifejezés egyezése
!~ reguláris kifejezés nem egyezése/különbözősége
! minta tagadása *!($1 == “”)*
NF % 2 != 0 páratlan számú mező van
length($0) > 25 a sor hossza 25 karakternél hosszabb

BEGIN és END minták

A BEGIN utáni tevékenységet az awk parancs az első bemenő sor beolvasása előtt hajta végre. Használhatjuk:

- változó inicializálásra
- mezőelválasztó karakter meghatározására
- fejléc nyomtatására

awk 'BEGIN (FS=":"}(print $2)' /etc/passwd

END

- Az END utáni tevékenységeket az awk a bemenet utolsó sorának feldolgozása után hajtja végre.

$ awk ’END (print NR)’ lista

kiírja a lista állomány sorainak számát

awk parancsfájl

```
#! /bin/awk
BEGIN { print "eleje" }
{
   print "cucc"
}
END { print "vége" }

awk -f program
```

awk parancsfájl

```
SU 01/30 13:15 - ttyq1 jose-root
SU 01/30 13:15 + ttyq1 jose-root
```

```
#! /bin/awk
# works for Solaris, IRIX and HPUX 10.20
BEGIN {
   print "--- checking sulog"
   failed=0
}
{
   if ($4 == "-") {
         print "failed su:\t"$6"\tat\t"$2"\t"$3
         failed=failed+1
   }
}
END {
   print "-----------------------------------
   --"
   printf("\ttotal number of records:\t%d\n", NR)
   printf("\ttotal number of failed su's:\t%d\n",failed)
}
```

CSS

- A CSS (angolul Cascading Style Sheets) a számítástechnika egyik stílusleíró nyelve, amely a HTML vagy XHTML dokumentumok megjelenését írja le. A CSS-t a html, xhtml dokumentumokban arra használják, hogy a lapok színét, hátterét, betűtípusait, elrendezéseit stb. beállítsák.

CSS

- A CSS bevezetésével elkülönítették a dokumentumok struktúráját a megjelenéstől, így a weblapok használhatóbbak, kezelhetőbbek, egyszerűbbek lettek, ugyanis a CSS használatával a kódból eltűntek egyebek között a karakterformázó tag-ek.
- A CSS ugyancsak alkalmas arra, hogy a dokumentum stílusát a megjelenítési módszer függvényében adja meg, így elkülöníthető a dokumentum formája a képernyőn, a nyomtatási lapon, a hangos böngészőben (amely beszédszintetizátor segítségével olvassa fel a weblapok szövegét), vagy braille készüléken megjelenítve.

Stíluslapok (CSS) használata I

- a dokumentumszerkezet megváltoztatása nélkül lehet befolyásolni a megjelenést és az elrendezést
- <STYLE> </STYLE>
- fejrészbe kerül, mivel nem képez semmiféle kimenetet
- szabályok halmaza

```
<style TYPE="text / css" >
        B, H2, H4 {color: green;
                font-size: 13px;
                text-align: center;}
</style>
```

Stíluslapok (CSS) használata II

- Szabályok részei:
  - a HTML elem neve, melyre a szabály vonatkozik
  - egy vagy több tulajdonságnévből (ez itt a color)
  - a tulajdonsághoz tartozó értékből (green)
- Egyedi szabályok készítése:

```
<styleTYPE=”text / css” >
        #stilus1 {color: green;}
         body {background-color: #7EEF00; }
</style>

<p ID=”stilus1”>Ez az első stíluslapom</p>
```

Stíluslapok (CSS) használata III

- Stíluslapokkal befolyásolható tulajdonságok:
  - Térköz
  - Színek
  - Betűtípusok
  - Margók, keretek

Stíluslapok (CSS) használata IV

- Szöveg igazítása
  - **letter-spacing**: betűk közötti távolság megadása
  - **text-decoration**: vonalakat helyezhetünk el a szöveg alatt, felett, vagy a szöveg belsejében
  - **text-align**: a szöveg igazítását határozhatjuk meg vele
  - **vertical-align**: feljebb, vagy lejjebb tolhatjuk az elemeket a vele egy sorban elhelyezett elemekhez képes
  - **text-transform**: kis és nagybetűk használatát szabályozza
  - **line-height**: az aktuális sor teteje és a következő sor teteje közötti távolságot állíthatjuk be segítségével
- háttérszíneket és háttérképeket
  - **color**: egy elem szövegének színét határozhatjuk meg vele
  - **background-color**: egy elem háttérszínét határozza meg
  - **background-image**: az elem hátteréül használt háttérkép kiválasztására szolgál
  - **background**: gyors megoldást kínál az előbb felsorolt háttérbeállítások meghatározására

Stíluslapok (CSS) használata V

- betűtípusok
  - **font-style**: a betűkészlet stílusát határozza meg
  - **font-family**: a szöveg betűtípusát határozhatjuk meg segítségével
  - **font-variant**: a normal érték a kisbetűket a hagyományos módon, a small –caps érték pedig kiskapitális formájában jeleníti meg
  - **font-size**: a betűkészlet pontmérete
  - **font-weight**: a szöveg vastagságát határozhatjuk meg
- általános elrendezés
  - **margin**: ugyanazt azt értéket rendeli mind a négy margószélességhez
  - **width**: egy elem szélességét határozza meg
  - **height**: egy elem magasságát határozza meg
  - **float**: egy elem szöveggel történő körbefuttatására szolgál
  - **clear**: ezzel a tulajdonsággal fejezhetjük be az elemek szövegekkel történő körbefuttatását

Stíluslapok (CSS) használata VI

- Mértékegységek használata stíluslapoknál
  - **px**: képpont; a számítógép kijelzőjének és más eszközöknek a legkisebb megcímezhető egységei
  - **pt**: pont; a betűkészletek méretének szabványos mértékegysége
  - **ex**: az x karakter hozzávetőleges magassága az adott betűkészletben
- Külső stíluslapok használata
  - Dokumentum átláthatósága érdekében
  - *<link REL=STYLESHEET TYPE="text/css" HREF="index.css">*

Leszármazottak alapján

- Előfordul olyan eset, amikor egy felsorolásban nem minden elemre kell linket tennünk, de azoknak különbözniük kell a felsorolás többi elemének színétől stb. Nézzük a CSS-t:

Leszármazottak CSS

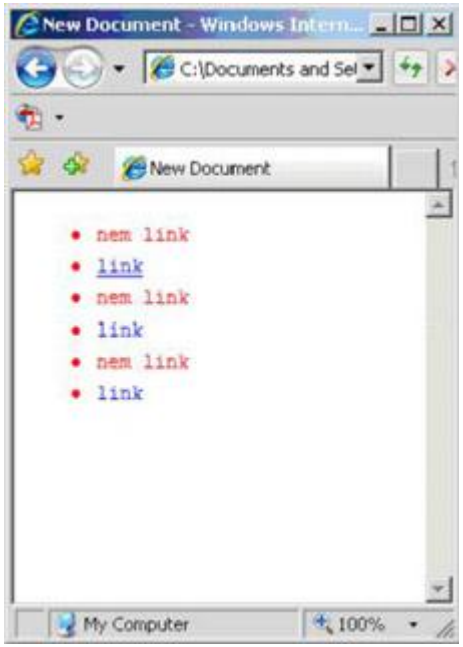
```
li {
    font-size:12px;
    font-family:'courier new';
    color:red;
}
li a:link, li a:visited{
    font-size:12px;
    font-family:'courier new';
    color:blue;
    text-decoration:none;
}
li a:hover{
    font-size:12px;
    font-family:'courier new';
    color:blue;
    text-decoration:underline;
}
```

Leszármazottak CSS #2

- Html:

```
<ul>
    <li>nem link</li>
    <li><a href="#">link</a></li>
    <li>nem link</li>
    <li><a href="#">link</a></li>
    <li>nem link</li>
    <li><a href="#">link</a></li>
</ul>
```

Leszármazottak CSS #3



Class vagy id attribútumok alapján

Az id-kat '#' prefix-szel a class-okat pedig '.' prefix-szel látjuk el CSS-ben.

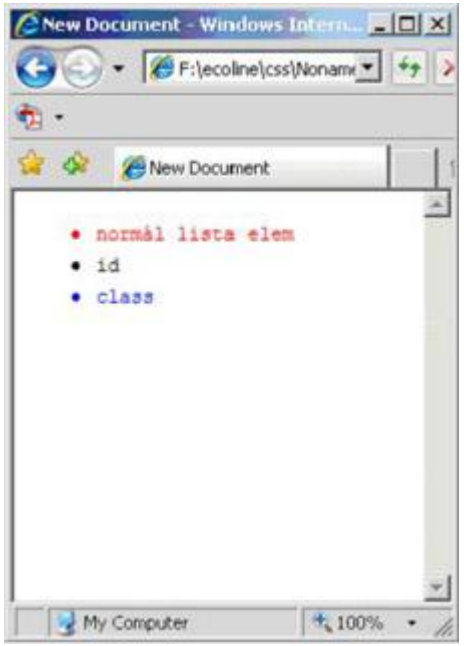
```
li {
        font-size:12px;
        font-family:'courier new';
        color:red;
    }
    . class{
        font-size:12px;
        font-family:'courier new';
        color:blue;

      }
    #id{
        font-size:12px;
        font-family:'courier new';
        color:black;

    }
```

- Html kód:

```
<ul>
        <li>normál lista elem</li>
        <li id="id">id</li>
        <li class="class">class</li>
</ul>
```



Hogyan kapcsolható a CSS a dokumentumhoz?

- **Belső**
- Figyelem: ilyenkor a CSS-t *<style></style>* tag-ek közé kell tennünk

- **Külső**
- Ilyenkor a CSS fájlban nem kell a tartalmat *<style></style>* tag közé tenni.

Belső CSS 1/6

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>

<style type="text/css">
#r {
	background-color: "gray";
	color: "white";
}
body {
	background-color: #7EEF00;
}
select {
	background-color: "red";
	color: "yellow";
}
```

Belső CSS 2/6

```
hr {
    border-top: 1px dashed;
      color: #9D9D96;
      height: 0px;
      width: 400px;
}
table {
      width: 690px;
      align: center;
      background-color: #9EFF90;
}
td {
      text-align: left;
}
```

Belső CSS 3/6

```
td.adat {
	font-size: 12px;
	width: 405px;
	color: #6C665C;
	font-family: 'Trebuchet MS', Tahoma, sans-serif;
	text-align: center;
}
p.left_title {
	width: 180px;
	font-size: 13px;
	font-family: 'Trebuchet MS', Tahoma, sans-serif;
	color: #C56252;
}
.style3 {
	font-family: bookman;
	color: #243D51;
}
```

Belső CSS 4/6

```
a:link {
    color: #243D51;
    text-decoration: none;
    font-size: 10px;
}
a:visited {
    color: #243D51;
    text-decoration: none;
    font-size: 16px;
}
```

Belső CSS 5/6

```
a:hover {
    color: #243D51;
    text-decoration: none;
    font-size: 16px;
}
img {
    margin: 10;
    border-width: 10;
}
</style>
</head>
```

Belső CSS 6/6

- <body>

```
<input name="klubtag_szam1" id="r" type="text" size="15"
maxlength=15><br>
<select>
                    <option>1</option>
                    <option>2</option>
</select>

<table border="1">
          <tr>      <td class="adat">gfdsh</td>    </tr>
</table>

<hr> <br>

<a href="borospince.hu">linkem</a><br>

<img src="button_belep.gif">

</body>

</html>
```

Külső CSS:

```
<head>

<link REL=STYLESHEET TYPE="text/css"
  HREF="css/index.css">

</head>
```