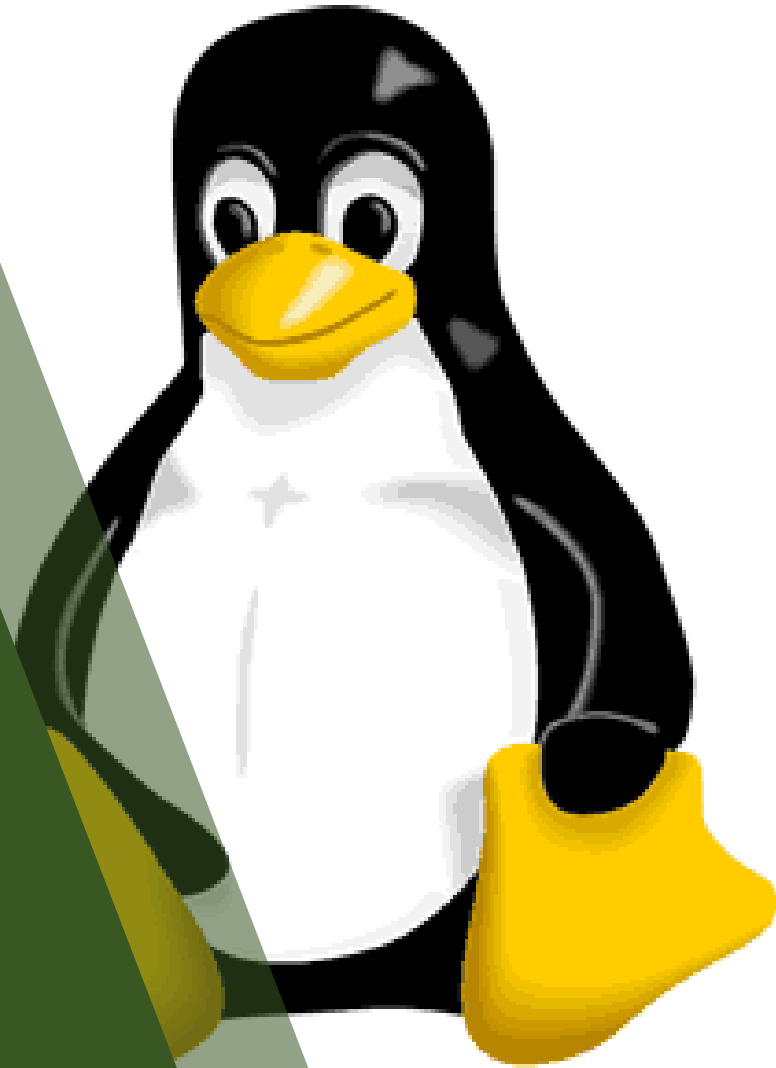


AWK - tömbök, beépített
függvények

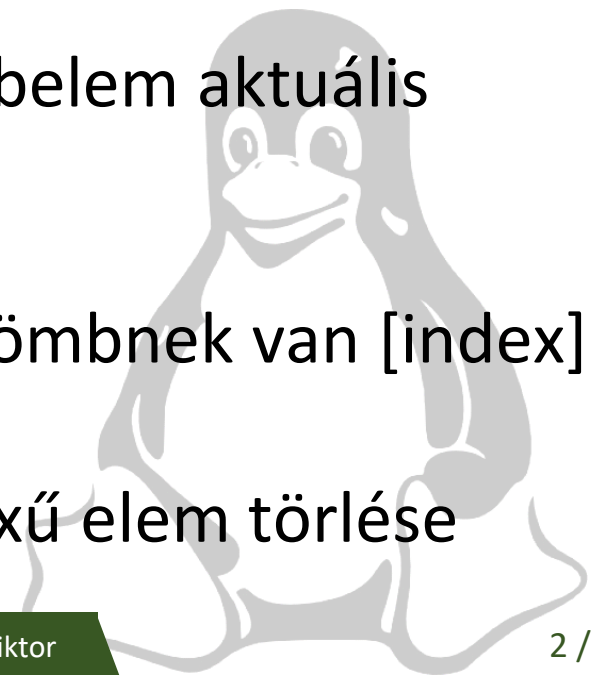
Operációs rendszerek

12. gyakorlat



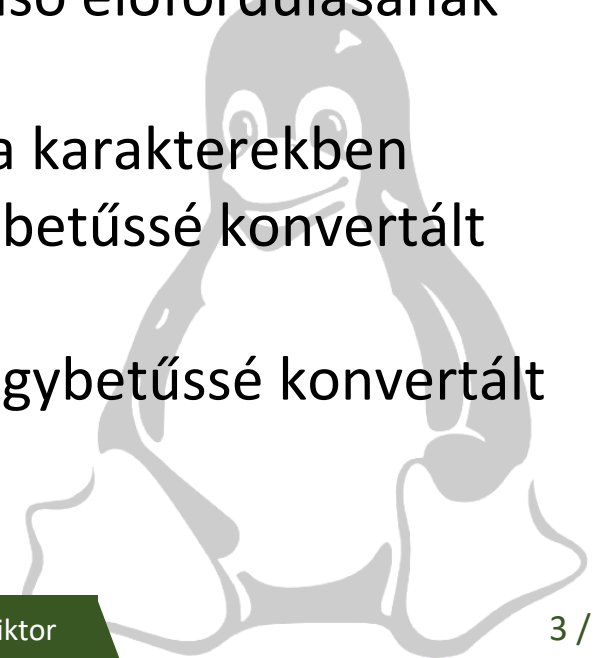
AWK - tömbök

- A tömb méretét nem kell előre lerögzíteni
- Asszociatív tömb: a tömbelemek indexe tetszőleges szöveg lehet (a számokat is szöveggé konvertálja)
- A tömb vegyesen tartalmazhat numerikus és szöveges elemeket is
- `[név][[index]] = [érték]`: értékadás egy létező tömbelemnek, vagy új elem beszúrása
- `[név][[index]]`: a megadott indexű tömbelem aktuális értékét jelöli
 - Definiálatlan elem értéke `""` ill. `0`
- `[index] in [név]`: igaz, ha a `[név]` nevű tömbnek van `[index]` indexű eleme
- **delete** `[név][[index]]`: a megadott indexű elem törlése



AWK - beépített függvények I

- Numerikus függvények:
 - **sin()**, **cos()**: trigonometria - szinusz, koszinusz
 - **sqrt()**: négyzetgyökvonás
 - **exp()**: exponens
 - **log()**: logaritmus
 - **int()**: egészszé konvertálás csonkolással
- Szöveges függvények:
 - **index([szöveg], [rész])**: a [rész] szöveg legelső előfordulásának pozíciója [szöveg]-ben
 - **length([szöveg])**: a megadott sztring hossza karakterekben
 - **tolower([szöveg])**: visszaadja a [szöveg] kisbetűssé konvertált értékét
 - **toupper([szöveg])**: visszaadja a [szöveg] nagybetűssé konvertált értékét



AWK - beépített függvények II

- **split**([szöveg], [tömb], [határolójel]): [szöveg]-et a [határolójel] mentén darabokra bontja, a darabokat a megadott tömbben eltárolja
 - A [szöveg] változatlan marad
 - A tömb elemei a darab sorszámaival lesznek indexelve
- **substr**([szöveg], [index]): a szöveg [index] sorszámu karakterétől kezdődő részét adja vissza
- **substr**([szöveg], [index], [hossz]): mint előbb, de legfeljebb [hossz] karakterből álló részt ad vissza



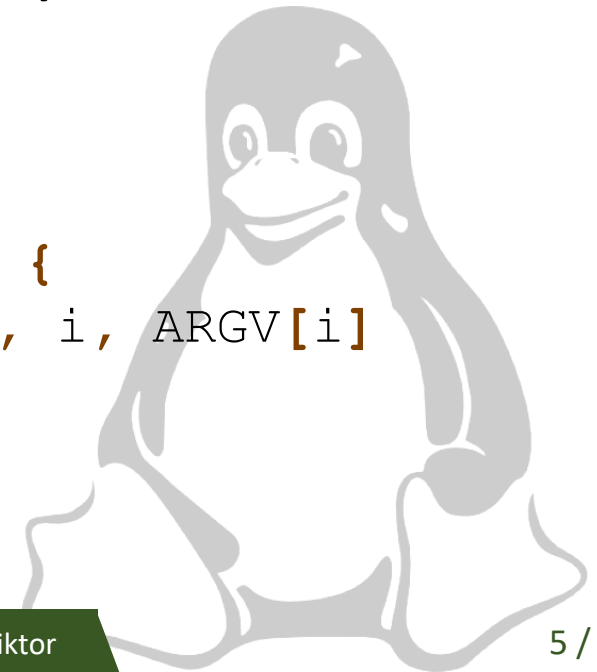
AWK - speciális paraméterek

- Parancssori argumentumok: a script indításakor annak neve után megadott paraméterek
 - Hasonló mint C-ben a parancssori argumentumok, illetve BASH-ban a pozícionális paraméterek
- **ARGC**: a megadott parancssori argumentumok száma
- **ARGV**: a parancssori argumentumokat egy listában tárolja, ennek indexelése 0-tól ARGC-1-ig érvényes

```
#!/usr/bin/awk -f
```

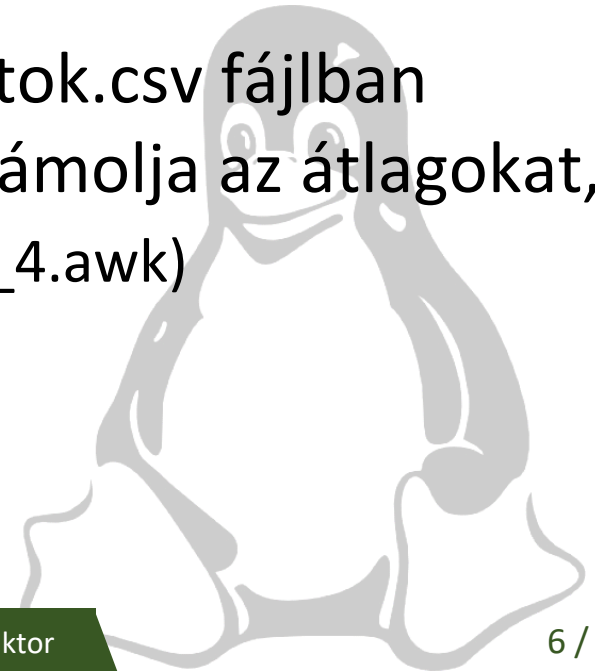
```
BEGIN {  
    for (i = 0; i < ARGC - 1; ++i) {  
        printf "ARGV[%d] = %s\n", i, ARGV[i]  
    }  
}
```

os12_1.awk



AWK - feladatok I

1. Írj olyan AWK scriptet, amely az ls kimenetéből megszámolja, hogy egyes felhasználók hány állományt hoztak létre! (os12_2.awk)
2. Írj olyan AWK scriptet, amely az ls kimenetéből kiszűri azokat az állományokat, melyeket 03-28 és 08-23 közötti időszakban hoztak létre! (év-hónap-nap formátum, év mindegy hogy mi) (os12_3.awk)
3. Írj olyan AWK scriptet, ami a osztalyzatok.csv fájlban található adatokból hallgatónként kiszámolja az átlagokat, a megoldáshoz tömböt használj! (os12_4.awk)



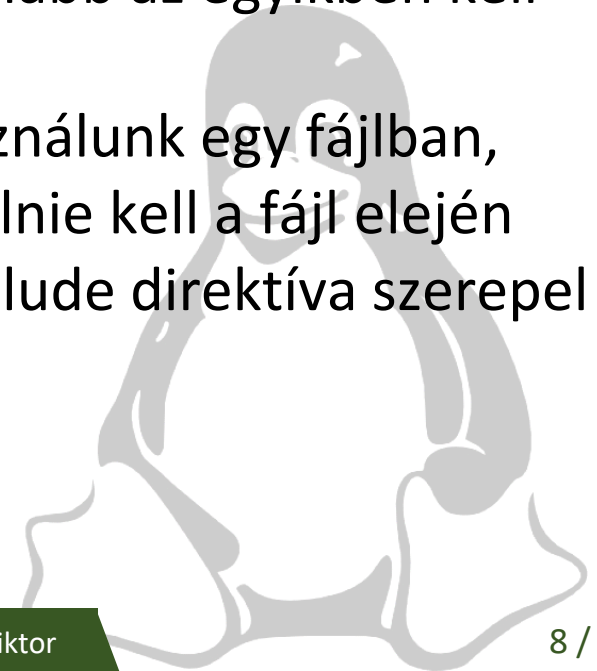
Jövő héten ZH

- A ZH-ra maximum 20 pont kapható, ez a következőképpen alakul:
 - Kettő könnyebb feladat 6-6 pontért
 - Egy BASH, egy AWK
 - Egy összetettebb feladat 8 pontért
 - Egy összetettebb AWK script
 - ZH-n minden lehet amit eddig vettünk:
 - Reguláris kifejezéseket ismerni kell, tudni kell ilyet írni
 - Parancsokat ismerni kell, amiket vettünk
 - BASH-t, AWK-t rutinosan használni!
- ZH-n csak a Rodek jegyzetet és a manuálokat lehet majd használni



AWK/BASH - összetettebb feladatok I

1. Írj olyan AWK scriptet, ami az ls kimenetéből meghatározza annak a két állománynak a nevét, melyek létrehozásának / utolsó módosításának időpontja között a legkevesebb a különbség!
2. Írj olyan BASH scriptet, ami .c kiterjesztésű fájlokat ellenőriz:
 - A mappában lévő állományok közül legalább az egyikben kell hogy legyen main függvény
 - Ha printf vagy scanf függvényhívást használunk egy fájlban, akkor az `#include<stdio.h>` -nak szerepelnie kell a fájl elején
 - Számoljuk meg hogy összesen hány `#include` direktíva szerepel a fájlokban!



AWK/BASH - összetettebb feladatok II

3. Írj olyan AWK scriptet, amely a bocsanat.txt állományban megszámolja az egyes szavak előfordulását. A program a kis- és nagybetűket vegye ekvivalensek, valamint az írásjelek miatt egy szó ne szerepeljen többször. A feldolgozás végén írjuk ki, a szavakat a következő formában:

- [szó] [el. szám] [el. százalék]
- Ahol: [szó] az adott szó, [el. szám] a szó előfordulásainak száma, [el. százalék] pedig hogy az összes szóhoz képest, ez a szó hány százalékban szerepel a szövegben.

