

11

AWK gyakorlás
Reguláris kifejezések

zh

- felépítése
 - két bevezető könnyebb feladat (5-5) pontért (hasonló nehézségű, mint az extra feladatok voltak)
 - egy nehezebb, összetettebb feladat 10 pontért
- anyagrész: minden, ami a félév során elhangzott, de elsősorban AWK
 - AWK scriptet kell írni,
 - lesz „beépített függvény” feladat
 - lesz „tömb” feladat
 - lesz „bejárós” feladat
 - lesz „reguláris kifejezés” feladat
- amit használhatsz (ne bízd el magad, ezek zh-n való lapozgatása nem lesz elég, ahhoz hogy megold a feladatokat):
 - Rodek Lajos gyakorlati jegyzete
- amit nem:
 - gyakorlati diasorjaim, honlapomon, pubban lévő anyagok
 - Erika jegyzete
 - internet

Emlékeztető: Elemi KIfejezések megadása

- `KARAKTER`: közösleges karakter egy példányára illeszkedik
- `\KARAKTER`: spec karakter is közösleges karakter lesz, annak példányára illeszkedik
- `()`: üres szóra illeszkedik
- `.` bármilyen karakter egy példányára illeszkedik (spec karaktereket kivéve)
- `[HALMAZ]`: felsorolt karakterek bármelyikének egy példányára illeszkedik, `[TÓL-IG]` hasonlóan, csak az intervallumba eső bármelyik karakterrel, `[^HALMAZ]` a nem felsorolt karakterek bármelyikének egy példányára illeszkedik
- `^` sor elejére illeszkedik, `$` sor végére illeszkedik

Emlékeztető: Összetett kifejezés

- iterálható:
 - KIF^* : a KIF akárhány egymást követő példányra illeszkedik, de az üres szóra is
 - KIF^+ : a KIF legalább egy egymást követő példányra illeszkedik
 - $KIF?$: 0 vagy 1 előfordulásra illeszkedik (0 előfordulás az üres szó)
 - $KIF\{i\}$: pontosan i egymást követő példányra illeszkedik
 - $KIF\{i,\}$: legalább i egymást követő példányra illeszkedik
 - $KIF\{i,j\}$: legalább i egymást követő példányra illeszkedik, de legfeljebb j egymás után követőre ($i \leq j$ teljesülése mellett)
- KIF_1KIF_2 : összefűzhető (konkatenáció)
- $KIF_1|KIF_2$: legalább az egyik kifejezésre illeszkedik (logikai vagy, alternáció)
- Műveletek erőssége: iteráció, konkatenáció, alternáció. De csoportosítható: $((KIF_1)(KIF_2))$

AWK regexp

`/regexp/`

- igaz ha a sorra illeszkedik a regexp
- hamis ha nem

`változó ~ /regexp/`

- igaz ha a változó értékére illeszkedik a regexp
- hamis ha nem

egrep és AWK összevetése

- írjuk ki azokat a sorokat, amiben szerepel a „nem” szó
- számoljuk meg azokat a sorokat, amiben szerepel a „nem” szó
- most ugyanezt AWK-kal (előző óra)
- most ugyanezt AWK-kal és reguláris kifejezéssel
 - sorra
 - elemenként
- egészítsük ki az adatsorunkat az alábbi sorral:
 - Neeeeeeeeem
- ezt is foglaljuk bele a reguláris kifejezésbe.

Gyakorlás a zh-ra

ls -l tartalmának feldolgozása

- Írjunk awk scriptet, ami kiírja a standard outputra az aktuális könyvtárban lévő alkönyvtárak és közönséges fájlok számát!
- Írjunk awk scriptet, ami felsorolja az aktuális könyvtárban lévő állományokat típus szerint