

Operációs rendszerek alapjai VII. gyakorlat

sed

sed

- **stream editor**
- a standard inputról vagy fájlból érkező adat szerkesztésére szolgál
- sorról sorra módosítja az adatot
- a szerkesztés eredményét alapértelmezetten a standard outputra írja, de fájlba is irányítható a kimenet
- használata:
 - `sed [options] commands [file-to-edit]`

sed

- alapesetben (ha nincs utasítás) a képernyőre írja ki az adatot
 - `sed " input.txt`
 - a " jelek között nincs utasítás, ezért csak kiírja az input.txt fájl tartalmát a konzolra
 - `cat input.txt | sed "`
 - a cat kimenetét kapja meg a sed és nem csinál vele semmit
- **p - print**
 - kiírja a szöveget
 - `sed 'p' input.txt`
 - minden sort kétszer fog kiírni egymás alá, mivel a p utasítás is szerepel
- **-n kapcsoló**
 - elnyomja az automatikus kiírást
 - `sed -n " input.txt`
 - nem csinál semmit
 - `sed -n 'p' input.txt`
 - kiírja egyszer az input.txt tartalmát

sed

- megadhatjuk, hogy mely sor(oka)t szeretnénk kiírni:
 - `sed -n '1p' input.txt`
 - kiírja az input.txt első sorát
 - `sed -n '1,4p' input.txt`
 - kiírja az input.txt első 4 sorát
- vagy, hogy melyeket nem
 - `sed -n '1~2p' input.txt`
 - minden páratlanadik sort írja ki
 - (`sed -n '1~3p' input.txt` az első sort, majd utána minden 3. sort írja ki...)

sed

- d - delete
 - törölhetünk vele sorokat
 - sed '1d' input.txt
 - törli az input.txt első sorát
 - sed '1,3d' input.txt
 - törli az input.txt első 3 sorát
 - 1,3 - intervallum amit töröl
 - sed '1~2d' input.txt
 - az első sort és utána minden második sort törli

sed

- a kimenetet természetesen át tudjuk irányítani a konzolon kívül máshova is
 - `sed '1p' input.txt > output.txt`
 - az output fájl az input.txt fájl első sorát kétszer, a többi sorát egyszer tartalmazza
 - `sed -n '1p' input.txt > output.txt`
 - csak az input.txt első sora kerül az outputba
 - `sed '1d' input.txt > output.txt`
 - az input.txt minden sora, kivéve az első az outputba kerül
 - `sed -n '1d' input.txt > output.txt`
 - ez mit fog csinálni?



sed

- a kimenet lehet maga az input fájl is
- -i - in-place
 - `sed -i -n '1p' input.txt`
 - az input.txt fájl első sora megmarad, a többi nem
- van lehetőség backup fájl készítésére is
 - a -i kapcsoló után megadunk egy tetszőleges “kiterjesztést”, az lesz a backup fájlunk “kiterjesztése”, vagyis:
 - `sed -i.vmi -n '1p' input.txt`
 - az input.txt fájl első sora megmarad, a többi nem
 - készül még egy input.txt.vmi fájl is, ami az eredeti input.txt fájlal egyezik meg

sed

- s - substituting

- a reguláris kifejezéssel megadott mintát megkeresi és kicseréli a megadott kifejezésre
- `sed 's/old_word/new_word/' input.txt`
 - az input.txt fájl minden sorában az első megtalált old_word-öt kicseréli a new_word-re
 - a "/" helyett más elválasztó karakter is használható (pl.. ; vagy , stb...)
 - fontos: a végén (jelen esetben a new_word után) is szerepelnie kell ennek az elválasztó karakternek
- ha nem csak az első találatot az adott sorban, hanem mindegyiket ki szeretnénk cserélni, akkor a g (global) flag-et kell használni
 - `echo cat catamaran catapult | sed 's/cat/dog/g'`
 - dog dogamaran dogapult
- de lehet például csak a másodikat cserélni
 - `echo cat catamaran catapult | sed 's/cat/dog/2'`
 - cat dogamaran catapult

G07F01

- Az input.txt fájlban a Mr.-t és a Mrs.-t cseréljük ki egymással (vagyis ahol eddig Mr. volt ott Mrs. legyen, ahol Mrs. volt ott pedig Mr.)
 - az eredményt egy newinput.txt fájlba írjuk

G07F02

- Az input2.txt fájlban az összes tyger szót tegyük “()” jelbe (a kis és nagybetű nem számít)
 - az eredményt egy newinput2.txt fájlba írjuk
- A & jelet írjuk át “and”-re
 - az eredményt szintén a newinput2.txt fájlba mentjük
- Az “or” helyett pedig írjunk || jeleket (de csak az “or” helyett, vagyis például a work ne legyen w||k, hanem maradjon work)
 - az eredményt szintén a newinput2.txt fájlba mentjük
- Írjuk ki a konzolra a newinput2.txt tartalmát, úgy, hogy az üres sorokat ne írjuk ki a páratlan sorokat viszont kétszer

G07F03

- Készítsünk egy ZH javító bash scriptet!
- A script 2 fájlt kap paraméterül, sorrendben: **zh.txt** és **megoldas.txt** (ezeket töltsétek le!)
 - A **zh.txt** egy hallgató válaszait tartalmazza
 - minden sor a hallgató egy kérdésre adott válasza (10 db kérdésre 10 darab válasz)
 - minden kérdés egy pontot ér, melyet a megfelelő kulcsszavakra lehet megkapni
 - A **megoldas.txt** tartalmazza az elvárt kulcsszavakat
 - minden sorában az adott sorhoz várt kulcsszavak találhatók (a zh.txt és a megoldas.txt sorai azonos sorrendben vannak!)
 - számít a szavak sorrendje, ha adott sorban több szó van, csak a megoldas.txt-nek megfelelő sorrend fogadható el
 - A kijavított zh a **corrected.txt** fájlba kerüljön
 - a corrected.txt az eredeti zh.txt minden sorát tartalmazza, azzal kiegészítve, hogy
 - a megtalált kulcsszavak || || között vannak (pl.: ezek a ||jó kulcsszavak||)
 - minden sor végén szerepel, hogy 0 vagy 1 pontot kapott az adott kérdésre
 - ha rossz a válasz (0 pontot kapott rá), akkor () -jelek között szerepeljenek a sor legvégén a helyes kulcsszavak is
 - a kijavított zh után új sorban szerepeljen, hogy hány pontot szerzett a 10-ből.
- Javítás után a konzolra írja ki, hogy hány százalékos lett a zh.