LINUX Opdrachten week 4

Avans Hogeschool, 's-Hertogenbosch, studiejaar 2019/2020, versie 1.0

Bob Polis

Inhoudsopgave

1 Dubbele woorden filteren

Maak één command-line waarmee je de dubbele woorden uit de bijgeleverde text file *words.txt* wegfiltert. Hint: gebruik o.a. `sort` en `uniq`. De command-line geeft dus als output de inhoud van *words.txt*, maar dan zonder de dubbele woorden. In de output hoeven de woorden niet in de originele volgorde te staan.

2 Splitsen van output

Lees de man-page van het commando `tee`. Waarom heet dat zo, denk je?

Maak nu een command-line waarmee je een listing van je huidige directory naar een file *listing.txt* schrijft, èn als output in de terminal het aantal items als getal toont.

3 Windows text files vertalen naar UNIX

In Windows text files eindigen regels met carriage return en line feed (`\r\n`), maar in UNIX is dat slechts een line feed (`\n`).

Lees de man-page van het commando `tr`.

Gegeven een Windows text file *windows.txt* (zie bijlage). Gebruik `tr` om *windows.txt* om te zetten naar een nieuwe file *unix.txt*.

4 Controle van de text files uit de vorige opgave

Met het commando hexdump kun je de bytes bestuderen die in een file staan, als volgt:

```
hexdump -C file
```

Als je dit doet met één van de files uit de vorige opgave krijg je heel veel output, terwijl een klein stukje al genoeg zou zijn om te zien of je vertaling werkt. Je kunt natuurlijk less gebruiken om de output per scherm te bekijken, maar je zou ook `head` of `tail` kunnen gebruiken om alleen het eerste of juist laatste stuk te zien.

- Schrijf een command-line met head om de eerste paar regels tekst als bytes te zien, en zo te controleren of `\r\n` overal netjes in `\n` is vertaald.
- Je kunt ook aannemelijk maken dat de vertaling correct is uitgevoerd door het aantal regels in *unix.txt* te tellen, en dat te vergelijken met de bestandsgrootte van *windows.txt* en *unix.txt*. Welke commando's gebruik je om deze controle uit te voeren?

5 Filteren met `sed`

Bewaar bijgevoegde file *index.html* op een handige plek in je home directory, bijvoorbeeld in `~/Documents`.

Als je met `sed` een file gaat filteren zal hij elke regel tekst sowieso naar de standard output schrijven, onafhankelijk of er regels bewerkt worden of niet. Je kunt dat gedrag indien gewenst onderdrukken met de optie `-n`. In dat geval schrijft `sed` alleen de regels waarvoor je dat expliciet aangeeft. Voorbeeld:

```
sed -n '/id=/p' index.html
```

Dit commando zal alleen de regels laten zien waar de string 'id=' in voor komt. Zoals je ziet zet je het zoekpatroon tussen slashes '/', en staat daar een 'p' achter om aan te geven dat je zo'n matchende regel wil printen. Probeer het uit!

Bedenk zelf het commando waarmee je alleen de regels laat zien met de speciale Internet Explorer-directives. Er zit hier een gemeen addertje onder het gras waarvoor meerdere oplossingen zijn…

6 In welke regels staat `<h2>`?

Maak een command-line om de regels uit *index.html* te tonen waarin `<h2>`-tags voorkomen.

7 Vervang `<h2>` door `<h3>`

Maak een command-line om overal in *index.html* `<h2>`-tags te veranderen in `<h3>`.

8 rot-13 codering

Maak een commando dat rot-13-codering op tekst toepast op de zin: “deze zin wordt rot-13 gecodeerd”. Bij rot-13 worden de letters van het alfabet 13 posities opgeschoven, waarbij de letters die daarbij “er af vallen” er aan de voorkant weer in worden geschoven. Dus a wordt n, b wordt o, … m wordt z, n wordt a, … z wordt m.

9 Filteren met `awk`

In de screencast heb je al kunnen zien dat `awk`-scripts altijd bestaan uit één of meer blokken van de vorm: *pattern* `{` *action* `}`, waarbij *pattern* de regels selecteert waarop de *action* moet worden gepleegd. Zoals daar al kort werd genoemd is `awk` eigenlijk een volwaardige scripttaal op zichzelf, en kun je daardoor ook van allerlei operatoren, functies en flow control gebruik maken. Hieronder een voorbeeld hoe je de oneven regels uit een tekstbestand laat zien.

```
awk 'NR % 2 { print $0 }' lines.txt
```

De file *lines.txt* is als bijlage toegevoegd, zodat je dit commando zelf kunt proberen.

`NR` is een ingebouwde variabele die als waarde steeds het huidige regelnummer heeft. `$0` is de hele input-regel. Het procentteken ‘`%`’ is de modulo-operator die je al uit Java (en andere talen) kent.

Pas het voorbeeld zodanig aan dat je de even regels als uitvoer krijgt.

10 Kolom-selectie

Bekijk de text file */etc/passwd*. Deze bevat informatie over gebruikers van het systeem, waarbij attributen zijn gescheiden door een dubbele punt.

Geef een commando waarmee je usernames en bijbehorende full name laat zien (indien gegeven).

11 Kolommen wisselen met `awk`

Zet de bijgevoegde tab-gescheiden file *data.csv* in *~/Documents*, en bekijk de inhoud.

Gebruik `awk` om een nieuwe file *data-new.csv* te creëren, waarin de volgorde van de velden als volgt is gewijzigd: *email*, *voornaam*, *tussenvoegsel*, *achternaam*.