Corrigé du cours Linux I/O

@22/03/2021

Partie 1

download the zipped data:

```
wget https://object.pouta.csc.fi/OPUS-UN/v20090831/mono/en.txt.gz
```

unzip a copy of the .gz archive:

```
gzip -dk en.txt.gz
gunzip -k en.txt.gz # also works - gunzip is basically an alias for $gzip -d
```

check the file metadata:

```
ls -lh en.txt
# -rw-r--r-- 1 kjm kjm 20M 17 nov.   2018 en.txt
```

FAST: display nb of **lines, words** and **bytes** in the file (no extra formatting):

```
cat en.txt | wc > params-en
# 74067 3037545 20826518
```

NEXT LEVEL: create a fancy script called **gen_params.sh**

```
#!/bin/bash

# remove the file if it already exists
FILE=./params
if [[ -f $FILE ]]; then
    echo $FILE exists. The file will be overwritten
    rm $FILE
fi

for f in $@
do
    echo ---------------------------------- >> params
    echo parameters for $f >> params
    echo number of lines: $(cat $f | wc -l) >> params
    echo number of words: $(cat $f | wc -w) >> params
    echo number of bytes: $(cat $f | wc -c) >> params
done

cat params
```

which gives the following output when called with two arguments (**./data/en.txt** and **./data/fr.txt**):

```
$ ./gen_params.sh ./data/en.txt ./data/fr.txt
./params exists. The file will be overwritten
---------------------------------
parameters for ./data/en.txt
number of lines: 74067
number of words: 3037545
number of bytes: 20826518
---------------------------------
parameters for ./data/fr.txt
number of lines: 74067
number of words: 3402633
number of bytes: 23634544
```

Partie 2

pré-traitement (cat, sed, grep)

1. remove the punctuation:

```
sed -e "s/[[:punct:]]//g"
```

2. change all the characters to lower-case:

```
sed -e "s/./\L&/g"
```

3. remove all the alphanumeric characters:

```
sed -e "s/[0-9]*//g"
```

4. remove all the empty lines:

```
grep -v '^$' # removes lines that are COMPLETELY empty
grep -v '^\s*$' # also removes lines containing spaces only
```

5. store the result in a new text file:

```
standard_output_from_previous_command > preproc_en.txt
```

6. all the commands chained together:

```
cat en.txt | sed -e "s/[[:punct:]]//g" -e "s/./\L&/g" -e "s/[0-9]*//g" | grep -v '^$' > preproc_en.txt
```

comptage d'occurences (sort, unique, head, tail)

```
cat preproc_en.txt | sed -e "s/\ /\n/g" | sort | uniq -c | sort -n > index_en
```

! new-lines should also be removed !

This can be done with:

```
tr '\n' ' ' # the 'TRansform' command
```

the previous command then becomes:

```
cat preproc_en.txt | tr '\n' ' ' | sed -e "s/[[:space:]]\+/\n/g" | sort | uniq -c | sort -n > index_en
```

generate the top/bottom 30 words from this index list:

top 30:

```
$ tail -n 30 index_en | sort -nr
286969 the
 190234 of
 149033 and
 102173 to
  70993 in
  37948 on
  33223 for
  28153 a
  25579 united
  24255 that
  22149 nations
  21362 its
  20606 with
  20381 as
  19744 by
  18444 international
  16854 states
  14121 at
  13948 resolution
  13701 all
  12610 development
  11428 their
  11066 republic
  10572 general
   9919 assembly
   9760 rights
   9578 human
   9519 report
   9420 december
   9168 including
```

bottom 30:

```
$ head -n 30 index_en
1 aaccrp
1 aahsg
1 abandon
1 abatement
1 abductees
1 abdullah
1 aberrations
1 abided
1 abiotic
1 abject
1 abovedescribed
1 absorbing
1 absorptions
1 abstain
1 abstinence
1 abuseibid
1 acac
1 acadd
1 accentuated
1 accentuates
1 acceptability
1 acceptably
1 accessed
1 accommodated
1 accommodations
1 accountants
1 accounted
1 accountfunded
1 accountopening
1 accountssee
```

! apparently some spaces have been forgotten !

→ check initial pre-processing (instead of completely removing the punctuation, substitute it by a space `sed -e "s/[[:punct:]]/" "/g"`, that should already get you a long way)

Bonus

awk exercise:

resources: https://www.grymoire.com/Unix/Awk.html

Transform the number of occurrences of the top_30 words into a percentage of how often they appear in the text, for this we'll use the following **awk_example.sh**:

```
#!/bin/bash

PREPROC_FILE="./preproc_en.txt"
words=$(cat $PREPROC_FILE | wc -w)
echo "total number of of words: $words"

INDEX_FILE="./top_30"
awk '
BEGIN {print "Frequency of appearance of the different words:\n"}
{printf("%*s\t-> %.2f%% of the time\n", 20, $2, $1/'$words'*100)}
END {print "\nDone"}
' $INDEX_FILE
```

output:

```
$ bash ./awk_example.sh
total number of of words: 2920353
Frequency of appearance of the different words:

                 the    -> 9.83% of the time
                  of    -> 6.51% of the time
                 and    -> 5.10% of the time
                  to    -> 3.50% of the time
                  in    -> 2.43% of the time
                  on    -> 1.30% of the time
                 for    -> 1.14% of the time
                   a    -> 0.96% of the time
              united    -> 0.88% of the time
                that    -> 0.83% of the time
             nations    -> 0.76% of the time
                 its    -> 0.73% of the time
                with    -> 0.71% of the time
                  as    -> 0.70% of the time
                  by    -> 0.68% of the time
       international    -> 0.63% of the time
              states    -> 0.58% of the time
                  at    -> 0.48% of the time
          resolution    -> 0.48% of the time
                 all    -> 0.47% of the time
         development    -> 0.43% of the time
               their    -> 0.39% of the time
            republic    -> 0.38% of the time
             general    -> 0.36% of the time
            assembly    -> 0.34% of the time
              rights    -> 0.33% of the time
               human    -> 0.33% of the time
              report    -> 0.33% of the time
            december    -> 0.32% of the time
           including    -> 0.31% of the time

Done
```