NLP 100 Exercise - Chapter 2: Unix Commands

10. Line Count

Count the number of lines of the file. Confirm the result by using `wc` command.

```
$ wc popular-names.txt
2780 11120 55026 popular-names.txt
```

2780 lines
11120 words
55026 characters

More info about `wc` command here: https://linuxize.com/post/linux-wc-command/

11. Replace tabs into spaces

Replace every occurrence of a tab character into a space. Confirm the result by using `sed`, `tr`, or `expand` command.

Information for `sed` command: https://www.geeksforgeeks.org/sed-command-in-linux-unix-with-examples/

```
$ sed 's/\t/ /g' popular-names.txt
```

Information for `tr` command: https://www.geeksforgeeks.org/tr-command-in-unix-linux-with-examples/

```
$ cat popular-names.txt | tr '\t' ' '
```

12. col1.txt from the first column, col2.txt from the second column

Extract the value of the first column of each line, and store the output into `col1.txt`. Extract the value of the second column of each line, and store the output into `col2.txt`. Confirm the result by using `cut` command.

```
$ cut popular-names.txt -f 1 > col1.txt
$ cut popular-names.txt -f 2 > col2.txt
```

More info: https://linuxize.com/post/linux-cut-command/

13. Merging col1.txt and col2.txt

Join the contents of `col1.txt` and `col2.txt`, and create a text file whose each line contains the values of the first and second columns (separated by tab character) of the original file. Confirm the result by using `paste` command.

```
$ paste col1.txt col2.txt
```

More info: https://www.geeksforgeeks.org/paste-command-in-linux-with-examples/

14. First N lines

Receive a natural number N from a command-line argument, and output the first N lines of the file. Confirm the result by using `head` command.

```
$ head -n 15 popular-names.txt
Mary	F	7065	1880
Anna	F	2604	1880
Emma	F	2003	1880
Elizabeth	F	1939	1880
Minnie	F	1746	1880
Margaret	F	1578	1880
Ida	F	1472	1880
Alice	F	1414	1880
Bertha	F	1320	1880
Sarah	F	1288	1880
John	M	9655	1880
William	M	9532	1880
```

```
James   M   5927    1880
Charles M   5348    1880
George  M   5126    1880
```

15. Last N lines

Receive a natural number N from a command-line argument, and output the last N lines of the file. Confirm the result by using `tail` command.

```
$ tail -n 15 popular-names.txt
Charlotte    F   12940   2018
Mia F   12642   2018
Amelia  F   12301   2018
Harper  F   10582   2018
Evelyn  F   10376   2018
Liam    M   19837   2018
Noah    M   18267   2018
William M   14516   2018
James   M   13525   2018
Oliver  M   13389   2018
Benjamin    M   13381   2018
Elijah  M   12886   2018
Lucas   M   12585   2018
Mason   M   12435   2018
Logan   M   12352   2018
```

More info: https://www.baeldung.com/linux/head-tail-commands

16. Split a file into N pieces

Receive a natural number N from a command-line argument, and split the input file into N pieces at line boundaries. Confirm the result by using `split` command.

N = 3

```
$ split -n l/3 popular-names.txt
```

(this command splits the file into 3 chunks xaa, xab, xac by line)

More info: https://www.geeksforgeeks.org/split-command-in-linux-with-examples/
(also `man split` is quite helpful for this one)

17. Distinct strings in the first column

Find distinct strings (a set of strings) of the first column of the file. Confirm the result by using `cut`, `sort`, and `uniq` commands.

Sort command sorts the lines by alphabetical order and also lowercase/uppercase.

```
$ sort col1.txt > col1_sorted.txt
```

Uniq command removes duplicate lines only if they are adjacent.

```
$ uniq col1_sorted.txt > col1_unique.txt
```

18. Sort lines in descending order of the third column

Sort the lines in descending numeric order of the third column (sort lines without changing the content of each line). Confirm the result by using `sort` command.

```
$ sort -nr -k 3 popular-names.txt
```

`-nr` : sort by numerical reverse order
`-k` : sort by column. `-k 3` means to sort by the 3rd column

19. Frequency of a string in the first column in descending order

Find the frequency of a string in the first column, and sort the strings by descending order of their frequencies. Confirm the result by using `cut`, `uniq`, and `sort` commands.

```
$ uniq -c col1_sorted.txt > col1_number_of_frequency.txt
$ sort -nr col1_number_of_frequency.txt > col1_number_of_frequency_sorted.txt
$ cat col1_number_of_frequency_sorted.txt
    118 James
    111 William
    108 Robert
    108 John
     92 Mary
     75 Charles
     74 Michael
     73 Elizabeth
     70 Joseph
     60 Margaret
     58 Thomas
     58 George
     57 David
     51 Richard
     45 Helen
     43 Frank
     43 Christopher
     41 Anna
     40 Edward
     39 Ruth
     38 Patricia
     37 Matthew
     36 Dorothy
     35 Emma
     32 Barbara
     31 Joshua
     31 Daniel
     26 Sarah
     26 Linda
     26 Jennifer
     26 Emily
     25 Jessica
     25 Jacob
     24 Susan
     24 Mildred
     24 Betty
     23 Henry
     23 Ashley
     22 Nancy
     21 Andrew
     20 Marie
     20 Florence
     20 Donald
     20 Amanda
     19 Samantha
     18 Olivia
     18 Melissa
     18 Madison
     18 Lisa
     18 Karen
     17 Stephanie
     17 Abigail
     16 Sandra
     16 Mark
     16 Ethel
     15 Michelle
     15 Isabella
     15 Heather
     15 Frances
     15 Ethan
     15 Carol
     15 Angela
     14 Shirley
     14 Kimberly
     14 Ava
     14 Amy
     13 Virginia
     13 Sophia
     13 Nicole
     13 Jason
     13 Hannah
     13 Deborah
     13 Brian
     12 Minnie
     12 Donna
     12 Bertha
     11 Cynthia
     10 Ronald
     10 Noah
     10 Nicholas
     10 Mia
     10 Doris
     10 Brittany
     10 Alice
      9 Tyler
      9 Joan
      9 Debra
      8 Taylor
      8 Mason
      8 Judith
      8 Ida
      8 Clara
      8 Alexis
      8 Alexander
      7 Tammy
      7 Steven
      7 Sharon
      7 Liam
      7 Harry
      7 Brandon
      6 Anthony
      5 Jeffrey
      5 Jayden
      5 Gary
      5 Charlotte
      5 Annie
      4 Lillian
      4 Kathleen
      4 Justin
      4 Chloe
      4 Benjamin
      4 Austin
      3 Megan
      3 Harper
      3 Evelyn
      3 Elijah
      3 Aiden
      2 Rebecca
      2 Oliver
      2 Logan
      2 Lauren
      2 Larry
      2 Bessie
      2 Amelia
      1 Walter
      1 Tracy
      1 Scott
      1 Rachel
      1 Pamela
      1 Lucas
      1 Lori
      1 Laura
      1 Kelly
      1 Julie
      1 Crystal
      1 Carolyn
```