1. locale Enter

The out put I got is not in standard C but in “UTF-8”

2. export SP LC\_ALL='C'

Change to Standard C

3. locale Enter

Check if change made successful, got “LC\_CTYPE="C"” this time

4. sort SP –c SP /usr/share/dict/words

The result is: “sort: /usr/share/dict/words:2: disorder: 10-point” which means the file is not sorted

5. sort SP –o SP words /usr/share/dict/words

Sort the file and put the output to my working directory

6. wget SP -O SP assign2.txt SP http://web.cs.ucla.edu/classes/winter17/cs35L/assign/assign2.html

Download the web page into a txt file named assign2.txt by using wget

7. tr SP –c SP 'A-Za-z' SP '[\n\*]' SP <assign2.txt> SP Command1.txt

Using first tr command and put the output into tile Command1.txt, the result of this command is delete all non-alphabetic words, such as “<”, and replace thos non-alphabetic words to a new blank line, and put every other word in a new line.

8. tr SP -cs SP 'A-Za-z' SP '[\n\*]' SP <assign2.txt> SP Command2.txt

This command did the same as previous one, but it will delete all the blank lines (squeezing) instead of keeping them.

9. tr SP -cs SP 'A-Za-z' SP '[\n\*]' SP <assign2.txt SP | SP sort SP > SP Command3.txt

This command did the same as previous one, but will also sort the file by every single words, and put each words into a newline.

10. tr SP -cs SP 'A-Za-z' SP '[\n\*]' SP <assign2.txt SP | SP sort SP –u SP > SP Command4.txt

This command did the same as previous one, but will also delete all the repeated words.

11. tr SP -cs SP 'A-Za-z' SP '[\n\*]' SP <assign2.txt SP | SP sort SP –u SP | SP comm SP - SP words SP > SP Command5.txt

This command will compare the difference between assign2.txt and words, it suppose to output three columns which first is the unique parts of assign2.txt, second column is the unique parts of words, the third columns is the common parts of both files. But some how it didn’t show as three horizontal columns but it showed up as 3 parts in one column in my terminal.

12. tr SP -cs SP 'A-Za-z' SP '[\n\*]' SP <assign2.txt SP | SP sort SP –u SP | SP comm SP -23 SP - SP words SP > SP Command5.txt

This command will only shows the common part between assign2.txt and words.

13. wget SP –O SP EngToHaw SP http://mauimapp.com/moolelo/hwnwdseng.htm

Download the website to file called EngToHaw

14. The script I wrote to transfer EngToHaw to the dictionary of Hawaiian is:

#!/bin/sh

grep '<td>.\*<\/td>' |

sed "s/<td>//g;s/<\/td>//g;s/<u>//g;s/<\/u>//g;s/<font.\*font>//g;s/<small>.\*<\/small>//g;s/(.\*)//g" |

sed "s/\`/\'/g" |

tr [A-Z] [a-z] |

sed "s/\s\*//g" |

tr -s '\n' |

sed '1~2d' |

sed "s/\,/\n/g" |

tr -c [![:space:]pk\'mnwlhaeiou] "B" |

sed "/B/d" |

sort |

uniq

grep '<td>.\*<\/td>' |

Grabbing the words between <td> and </td>

sed "s/<td>//g;s/<\/td>//g;s/<u>//g;s/<\/u>//g;s/<font.\*font>//g;s/<small>.\*<\/small>//g;s/(.\*)//g" |

Delete every <td> </td> <u> </u>, all pattern from ‘<font’ to ‘font>’, and all pattern from ‘<small>’ to ‘</small>’, then delete ‘(‘ and ‘)’.

sed "s/\`/\'/g" |

Change ` to Hawaiian letter '

tr -s '\n' |

Delete all leading spaces

sed '1~2d' |

Delete all odd lines which contain English words

sed "s/\,/\n/g" |

Then put the words after ‘,’ into new lines

tr -c [![:space:]pk\'mnwlhaeiou] "B" |

Transfer all non-Hawaiian letter and non-space class(such as \n \r \s) letter to an English letter ‘B’

sed "/B/d" |

Delete all lines which contain letter ‘B’, which means deleted all words which contained non-Hawaiian letters

sort | uniq

sort the file and delete all repeated words. Done

15. cat EngToHaw | ./buildwords | less

Checking if the script works. It works well

16. cat EngToHaw | ./buildwords > hwords

Redirecting the output to the file called hwords, now hwords is a dictionary of Hawaiian

17. wget http://web.cs.ucla.edu/classes/winter17/cs35L/assign/assign2.html

Download assign2.html

18. cat assign2.html | tr -cs 'A-Za-z' '[\n\*]' | tr '[:upper:]' '[:lower:]' | sort -u | comm -23 – words | wc -l (no ‘wc -l’ but ‘less’ to check which words)

Checking the HTML file assign2.html with English dictionary ‘words’, got 39 misspelling as English

19. cat assign2.html | tr -cs 'A-Za-z' '[\n\*]' | tr '[:upper:]' '[:lower:]' | sort -u | comm -23 – hwords | wc –l (no ‘wc -l’ but ‘less’ to check which words)

Checking the HTML file assign2.html with Hawaiian dictionary ‘hwords’, got 409 misspelling as Hawaiian

20. cat assign2.html | tr -cs 'A-Za-z' '[\n\*]' | tr '[:upper:]' '[:lower:]' | sort -u | comm -23 – words > EngMisspelling

Output the English misspelling into a file

21. comm -12 EngMisspelling hwords | less