CS35L – *Fall 2018*

Slide set:	2.2
Slide topics:	Lab Assignment and HW
Assignment:	2

Lab 2

Assignment 2 Details

Submit 3 files:

Script "buildwords"

Simple text file "lab2.log"

- 80 character limit per row

Check everything on SEASnet!

Assignments graded on SEASnet servers (eg. lnxsrv07)

What is Lab 2 About?

Build a spelling checker for the Hawaiian language

(Get familiar with sort, comm and tr commands!)

Steps

- Download a copy of web page containing basic English-to-Hawaiian dictionary
- Extract only the Hawaiian words from the web page to build a simple Hawaiian dictionary. Save it to a file called hwords (**site scraping**)
- Automate site scraping: buildwords script (cat hwnwdseng.htm | buildwords > hwords)
- Modify the command in the lab assignment to act as a spelling checker for Hawaiian
- Use your spelling checker to check hwords and the lab web page for spelling mistakes

Useful Text Processing Tools

- wc: outputs a one-line report of lines, words, and bytes
- head: extract top of files
- tail: extracts bottom of files
- tr: translate or delete characters
- grep: print lines matching a pattern
- sort: sort lines of text files
- sed: filtering and transforming text

Lab2.log

- .log is the same as .txt – no difference
- Ex:
  - 1. I used wget to download the webpage
  - 2. I ….
  - 3. Answer to #3 here
- Should read basically like a lab journal
- Keep things concise!

Lab Hints

- Run your script on seasnet servers before submitting to CCLE
- sed '/patternstart/,/patternstop/d‘
  - delete patternstart to patternstop, works across multiple lines
    will delete all lines starting with patternstart to patternstop
- The Hawaiian words html page uses \r and \n for new lines
  - od –c hwnwdseng.htm
  - to see the ASCII characters
- You can delete blank white spaces such as tab or space using
  - tr -d '[:blank:]’
  - Use tr -s to squeeze multiple new lines into one
- sed 's/<[^>]*>//g' a.html to remove all HTML tags

Buildwords

- Hawaiian.html -> buildwords -> hwords
- Buildwords
  - Read from STDIN and perform work on input
  - Output to STDOUT
- Ex:
  - $ ./buildwords < hawaiian.html > hwords

Homework 2

Real life problem

Files have different encodings, it is hard for applications to decide without reading the whole file

Use a header to indicate the encoding so that the application doesn’t have to read the whole file

Text encoding – UTF8 and ASCII

Character encoding is used to represent a repertoire of characters by some kind of encoding system. (Wikipedia)

ASCII is highly limited, though usually sufficient for the English language and daily use

UTF-8 encompasses far more characters, and is backward compatible with ASCII

While ASCII uses one byte to encode a character, UTF-8 is variable length and uses 1-4 bytes

Homework – find UTF8 and ASCII files

- Locate file(s) in a given path
- Check if they are plain-ASCII or UTF-8 (but not plain ASCII)
- If they are plain-ASCII but with a header, or UTF-8 without a header, we have a problem
- Just print out the names of the file(s) with these issues

Commands and Options

- `find –exec`
- `grep`
  - `-l`
  - `-L`
  - `-v`
  - `-a`
- ASCII character set
  - `[\x00-\x7F]`
  - Hexadecimal representation of 0-127