Filters and Regexps

This file contains examples of the use of the most common Unix filter programs (`egrep`, `wc`, `head`, etc.) It also contains solutions to the exercises discussed in lectures.

1. Consider a a file course_codes containing UNSW course codes and names.

```
$ ls -l course_codes
-rw-r--r-- 1 cs2041 cs2041 603446 May 30  2019 course_codes
```

```
$ wc course_codes
 18181  79223 603446 course_codes
```

```
$ head course_codes
ACCT1501 Accounting & Financial Mgt 1A
ACCT1511 Accounting & Financial Mgt 1B
ACCT2101 Industry Placement 1
ACCT2507 Intro  to Accounting Research
ACCT2522 Management Accounting 1
ACCT2532 Management Accounting (Hons)
ACCT2542 Corporate Financial Reporting
ACCT2552 Corporate Financial Rep (Hons)
ACCT3202 Industry Placement 2
ACCT3303 Industry Placement 3
```

It looks like the code is separated from the title by a number of spaces. We can check this via `cat -A`:

```
$ head -5 course_codes | cat -A
ACCT1501 Accounting & Financial Mgt 1A$
ACCT1511 Accounting & Financial Mgt 1B$
ACCT2101 Industry Placement 1$
ACCT2507 Intro  to Accounting Research$
ACCT2522 Management Accounting 1$
```

This shows us that our initial guess was wrong, and there's actually a tab character between the course code and title (shown as `^I` by `cat -A`). Also, the location of the end-of-line marker (`$`) indicates that there are no trailing spaces or tabs.

If we need to know what COMP courses there are:

```
$ egrep -c COMP course_codes
191
```

```
COMP1001 Introduction to Computing
COMP1011 Computing 1A
COMP1021 Computing 1B
COMP1081 Harnessing the Power of IT
COMP1091 Solving Problems with Software
COMP1400 Programming for Designers
COMP1711 Higher Computing 1A
COMP1721 Higher Computing 1B
COMP1911 Computing 1A
COMP1917 Computing 1
COMP1921 Computing 1B
COMP1927 Computing 2
COMP2011 Data Organisation
COMP2021 Digital System Structures
COMP2041 Software Construction
COMP2091 Computing 2
COMP2110 Software System Specification
COMP2111 System Modelling and Design
COMP2121 Microprocessors & Interfacing
COMP2411 Logic and Logic Programming
COMP2711 Higher Data Organisation
COMP2811 Computing B
```

Either of the two commands below tell us which courses have "comp" in their name or code (in upper or lower case).

```
$ tr A-Z a-z <course_codes | egrep comp
aciv2518 eng computational methods 1
acsc1600 computer science 1
acsc1800 computer science 1e
acsc2015 interactive computer graphics
acsc2020 computer science core b2
acsc2021 computer systems architectrue 2
acsc2107 computer languages b
acsc2601 computer science 2a
acsc2602 computer science 2b
acsc2802 computer science 2ee
acsc3003 computer project
acsc3029 computing project 3
acsc3030 cryptography & computer securi
acsc3601 computer science 3a
acsc3603 computer science 3c
acsc4191 computer science 4 (hons) f/t
acsc7304 computer graphics
acsc7306 computer speech processing
acsc7336 computer security
acsc8248 computer graphics (12 cpt)
```

```
$ egrep -i comp course_codes
ACIV2518 Eng Computational Methods 1
ACSC1600 Computer Science 1
ACSC1800 Computer Science 1E
ACSC2015 Interactive Computer Graphics
ACSC2020 Computer Science Core B2
ACSC2021 Computer Systems Architectrue 2
ACSC2107 Computer Languages B
ACSC2601 Computer Science 2A
ACSC2602 Computer Science 2B
ACSC2802 Computer Science 2EE
ACSC3003 Computer Project
ACSC3029 Computing Project 3
ACSC3030 Cryptography & Computer Securi
ACSC3601 Computer Science 3A
ACSC3603 Computer Science 3C
ACSC4191 Computer Science 4 (Hons) F/T
ACSC7304 Computer Graphics
ACSC7306 Computer Speech Processing
ACSC7336 Computer Security
ACSC8248 Computer Graphics (12 Cpt)
```

The second one looks better because the data itself isn't transformed, only the internal comparisons.

If we want to know how many courses have "computing" or "computer" in their title, we have to use `egrep`, which recognises the alternative operator "|", and `wc` to count the number of matches. There are a couple of ways to construct the regexp:

```
$ egrep -i 'computer|computing' course_codes | wc
    262    1149    9027
```

```
$ egrep -i 'comput(er|ing)' course_codes | wc
    262    1149    9027
```

If you don't like the irrelevant word and character counts, use `wc -l`.

Most of these 80 matches were CSE offerings, whose course codes begin with COMP, SENG or BINF. Which of the matches were courses offered by other schools?

Think about it for a moment.... There's no "but not" regexp operator, so instead we construct a composite filter with an extra step to deal with eliminating the CSE courses:

```
$ egrep -i 'computer|computing' course_codes | egrep -v '^(COMP|SENG|BINF)'
ACSC1600 Computer Science 1
ACSC1800 Computer Science 1E
ACSC2015 Interactive Computer Graphics
ACSC2020 Computer Science Core B2
ACSC2021 Computer Systems Architectrue 2
ACSC2107 Computer Languages B
ACSC2601 Computer Science 2A
ACSC2602 Computer Science 2B
ACSC2802 Computer Science 2EE
ACSC3003 Computer Project
ACSC3029 Computing Project 3
ACSC3030 Cryptography & Computer Securi
ACSC3601 Computer Science 3A
ACSC3603 Computer Science 3C
ACSC4191 Computer Science 4 (Hons) F/T
ACSC7304 Computer Graphics
ACSC7306 Computer Speech Processing
ACSC7336 Computer Security
ACSC8248 Computer Graphics (12 Cpt)
ACSC9000 Computer Science Research F/T
```

The last ones are from the Computer Science school at ADFA.

2. Consider a file called enrollments which contains data about student enrollment in courses. There is one line for each student enrolled in a course:

```
$ ls -l enrollments
-rw-r--r-- 1 cs2041 cs2041 855297 May 30  2019 enrollments
```

```
$ wc enrollments
  7569  42802 855297 enrollments
```

```
$ head enrollments
COMP1511|5013566|Xin, Mackenzie Darren                          |3648/2|COMPI1
MTRNAH|071.800|17s2|19910428|M
COMP9902|5079970|Park, Xue Hannah Vanessa                       |8543  |ELECAH
|079.333|17s2|19900209|F
COMP1511|5059072|Chung, Michael Jia Tianyu                      |3778/1|COMPCS
|057.250|17s2|19990801|M
COMP1521|5060774|Lim, Stephanie Lauren                          |3785/1|COMPA1
|000.000|17s2|19890113|F
COMP1531|5060774|Lim, Stephanie Lauren                          |3785/1|COMPA1
|000.000|17s2|19890113|F
COMP2521|5060774|Lim, Stephanie Lauren                          |3785/1|COMPA1
|000.000|17s2|19890113|F
COMP9020|5060538|Bi, Samuel Shiyu                               |6021  |COMPA1
|078.125|17s2|19911004|M
COMP9021|5060538|Bi, Samuel Shiyu                               |6021  |COMPA1
|078.125|17s2|19911004|M
COMP9902|5072116|Hu, Kai Zhi Patrick                            |3707/1|SENGAH
|070.750|17s2|19930424|M
COMP1511|5036926|Fang, Rebecca Lauren                           |8543  |COMPCS
|000.000|17s2|20000921|F
```

The following commands count how many students are enrolled in COMP2041 or COMP9041. The course IDs differ only in one character, so a character class is used instead of alternation.

The first version below is often ferred because initially you may want to know "Ḏṙ ỹ N̥I Ŏ̈ă ӾӾӾ", then having found that out the next question might be, "ỹ Ǵk̠ḅẹ̑ỴǴN̥ǴI Ẏl N̥ ṿk̠Ǵṙ ḂÁ̓ḁ̊ ṙ ẃẎr ṙ ḂÁ̃Ḏ̠ǴN̥". Then it's a simple matter of replacing `wc` by `head`.

```
$ egrep '^COMP[29]041' enrollments | wc -l
511
```

```
$ egrep -c '^COMP[29]041' enrollments
511
```

The last field field in the enrollment file records the student's gender. This command counts the number of female students enrolled in the courses.

```
$ egrep '^COMP[29]041' enrollments | egrep 'F$' | wc -l
106
```

Not a very good gender balance, is it?

By the way, the two `egrep`s could have been combined into one. How?

This command will give a sorted list of course codes:

```
$ cut -d'|' -f1 enrollments | sort | uniq
COMP1400
COMP1511
COMP1521
COMP1531
COMP2041
COMP2121
COMP2521
COMP3151
COMP3161
COMP3222
COMP3331
COMP3421
COMP3431
COMP3511
COMP3601
COMP3901
COMP4121
COMP4161
COMP4336
COMP4418
```

The student records system known to users as myUNSW is built on top of a large US product known as PeopleSoft (the company was taken over by Oracle in 2004). On a scale of 1 to 10 the quality of the design of this product is about 3. One of its many flaws is its insistence that everybody must have two names, a "Last Name" and a "First Name", neither of which can be empty. To signify that a person has only a single name (common in Sri Lanka, for example), the system stores a dot character in the "First Name" field. The enrollments file shows the data as stored in the system, with a comma and space separating the component names. It has some single-named people (note that the names themselves have been disguised):

```
$ egrep ', \.' enrollments
COMP1511|5007185|Nguyen, .                                         |3764/1|COMPCS
|000.000|17s2|19861014|M
COMP3331|5071779|Yuan, .                                           |3785/2|COMPCS
|072.063|17s2|19901016|M
COMP3511|5071779|Yuan, .                                           |3785/2|COMPCS
|072.063|17s2|19901016|M
COMP4920|5071779|Yuan, .                                           |3785/2|COMPCS
|072.063|17s2|19901016|M
COMP9021|5054494|Dang, .                                           |8543  |ELECAH
PHYSC1|000.000|17s2|19931117|M
COMP3421|5072547|Zhou, .                                           |3978/2|COMPA1
|068.167|17s2|19870503|M
COMP3431|5072547|Zhou, .                                           |3978/2|COMPA1
|068.167|17s2|19870503|M
COMP3601|5072547|Zhou, .                                           |3978/2|COMPA1
|068.167|17s2|19870503|M
COMP9901|5065745|Lo, .                                             |8543  |COMPAS
COMPIS|082.545|17s2|19981128|M
COMP9024|5099838|Chen, .                                           |3978/1|COMPCS
|072.500|17s2|19980127|F
```

What would have happened if we forgot the backslash?

If we wanted to know how many different students there were of this type rather than all enrollments, just cut out the second field (student ID) and use `uniq`. It's not necessary to sort the data in this case only because the data is ***clustered,*** that is, all equal values are adjacent although they're not necessarily sorted.

```
$ egrep ', \.' enrollments | cut -d'|' -f2 | uniq | wc
      6       6      48
```

3. Now let us turn our attention from students and courses to programs. The enrollments file, as well as linking a student to the courses they're taking, also links them to the program (degree) that they are currently enrolled in. Consider that we want to find out the program codes of the students taking COMP2041. The following pipeline will do this:

```
$ egrep 'COMP[29]041' enrollments | cut -d'|' -f4 | cut -d/ -f1  | sort | uniq
1540
1650
2765
3133
3436
3529
3554
3564
3645
3647
3648
3707
3710
3711
3715
3725
3731
3736
3761
3762
```

If we want to know how many students come from each program, ordered from most common program to least common program, try this:

```
$ egrep COMP[29]041 enrollments | cut -d'|' -f4 | cut -d/ -f1 | sort | uniq -c | sort -nr
    175 8543
     76 3707
     38 3978
     26 3778
     17 1650
     14 7543
     11 5543
     11 3784
     10 3967
      9 3772
      9 3764
      9 3645
      8 3983
      6 3781
      6 3768
      5 6021
      5 3969
      5 3959
      5 3715
      4 3970
```

Note that a tab is usually inserted between the count and the data, but not all implementations of the `uniq` command ensure this.

4. Consider a file called program_codes that contains the code and name of each program offered at UNSW (excluding research programs):

```
$ wc program_codes
 1798  6466 46572 program_codes
```

```
$ head program_codes
0350 Medicine (Prince Henry/POW)
0351 Medicine (SWS Clinical School)
0352 Medicine (St George)
0353 Medicine (St Vincent's)
0360 Pathology
0370 Physiology and Pharmacology
0375 Rural Health
0380 Obstetrics and Gynaecology
0390 Psychiatry
0400 Surgery (Prince Henry/POW)
```

We can use this file to give more details of the programs that COMP2041 students are taking, if some users don't want to deal with just course codes.

```
$ egrep COMP[29]041 enrollments | cut -d'|' -f4 | cut -d/ -f1 | sort | uniq | join - program_codes
1540  Economics
1650  Computer Science and Eng
2765  Computer Science and Eng
3133 Mat Sci and Eng Hons/BiomedEng
3436 Music
3529 Commerce/Science
3554 Commerce (Co-op)
3564 Economics / Science (Adv Math)
3645 Computer Engineering
3647 Bioinformatics
3648 Software Engineering
3707 Engineering (Honours)
3710 Mechanical & Manufacturing Eng
3711 Mechanical & Manf Eng/Science
3715 Engineering/Commerce
3725 Electrical Engineering/Science
3731 BE ME Electrical Engineering
3736 BE (Hons) ME Elec Eng
3761 Adv Math (Hons) / Eng (Hons)
3762 AdvSci(Hons)/Engineering(Hons)
```

We can combine the enrollment counts (for both courses) with the program titles to produce a self-descriptive tally. It's even better if it's in decreasing order of popularity, so after joining the tallies with the program titles, re-sort the composite data:

```
$ egrep 'COMP[29]041' enrollments | cut -d'|' -f4 | cut -d/ -f1 | sort | uniq -c | join -1 2 -a 1 -
program_codes  | sort -k2rn
8543 175  Information Technology
3707 76 Engineering (Honours)
3978 38 Computer Science
3778 26
1650 17  Computer Science and Eng
7543 14  Computing
3784 11
5543 11  Information Technology
3967 10 Commerce / Computer Science
3645 9 Computer Engineering
3764 9 Engineering (Hons)/Commerce
3772 9 Engineering(Hons)/Computer Sci
3983 8 Science/Computer Science
3768 6 Eng (Hons) / MBiomedE
3781 6
3715 5 Engineering/Commerce
3959 5
3969 5 Media Arts (Hons) / Comp Sci
6021 5  Exchange Program
```

Note the curious extra space before the title of programs 8543, 6021, and others. It took me a while to work it out, can you? (Hint: how are the programs shown in the enrollment file?) Suggest an appopriate change to the pipeline.

5. Lecture exercises on `wc`:

   a. how many different programs does UNSW offer?

   ```
   $ wc -l program_codes
   1798 program_codes
   ```

   b. how many times was WebCMS accessed?

   ```
   $ wc -l access_log
   59779 access_log
   ```

   c. how many students are studying in CSE?

   ```
   $ wc -l enrollments
   7569 enrollments
   ```

   The above solutions assume that we're talking about total enrollments. If the question actually meant how many distinct indivduals are studying courses offered by CSE, then we'd answer it as:

   ```
   $ cut -d'|' -f2 enrollments | sort | uniq | wc -l
   3791
   ```

   d. how many words are there in the book?

   ```
   $ wc -w book
   60428 book
   ```

   e. how many lines are there in the story?

```
$ wc -l story
87 story
```

COMP(2041|9044) 20T2: Software Construction is brought to you by
the School of Computer Science and Engineering
at the University of New South Wales, Sydney.
For all enquiries, please email the class account at cs2041@cse.unsw.edu.au
CRICOS Provider 00098G