Operating Systems and Systems Integration

Regular Expressions, `sed` and `awk`

1 Background

For the background information you need to answer these questions, please refer to the shell programming lecture notes.

The program `egrep` stands for *extended* `grep`. It supports so-called *extended regular expressions*. Well, since I have taught you the syntax of extended regular expressions in the lecture, I suggest that you use `egrep` rather than `grep` with regular expressions.

Both `grep` and `egrep` support an option `-o` to print *only* the part of the line that matches the expression.

The GNU `sed` program supports the `-r` option that tells `sed` to use regular expressions like `egrep`. So when you use `sed`, use it with `sed -r`.

The GNU `awk` program has an option `--posix` that makes `awk` behave like `egrep` so that it understands the use of $\{n\}$, $\{n,m\}$ without having to put in extra backslashes '\'.

> You might wonder what is the difference between "extended regular expressions" and the regular expressions that `grep` uses? The difference is explained at the end of the `info` page:
>
> `$ info '(grep)Regular Expressions'`
>
> where it says, "In basic regular expressions the metacharacters '?', '+', '{', '|', '(', and ')' lose their special meaning; instead use the backslashed versions '\?', '\+', '\{', '\|', '\(', and '\)'."

2 Questions

2.1 egrepping through the dictionary

Your dictionary is a file `/usr/share/dict/words`. Use `egrep` to:

1. Find all words containing three letter 'a's.
2. Find all words containing *no* vowels. (A *vowel* is one of the letters 'a', 'e', 'i', 'o' and 'u'.)
3. Find all words containing *at least* 5 vowels. Count the number of matching words.
4. Find all words containing *exactly* 5 vowels. Count the number of matching words.

2.2 `egrep`: Selecting data from student records

1. Save the file `http://nicku.org/snm/lab/regular-expressions/artificial-student-data.txt` to your local directory. For this data, write a regular expression that will select each of the following. Test it on the data using `egrep -o`

2. student number

✎

3. Hong Kong ID. Count the number of Hong Kong IDs.

✎

4. the course code. Count the number of courses.

The course and year are shown in this case on the sixth line: `2241/2`. The course is 2241; this is the second year of study.

✎

5. the year of study

✎

6. The company the student works for

✎

7. The home telephone number

✎

8. The gender of the student

✎

9. The student's name

✎

2.3 Using `sed`

Write a `sed` expression to output *only* the data for which you wrote *each* of the regular expressions above. For example, write a `sed` command that will print *only* the HK IDs and *all* the HK IDs from the file, using the regular expression you wrote for question 3. You should write eight `sed` expressions.

2. student number

3. Hong Kong ID.

4. the course code.

5. the year of study

6. The company the student works for

7. The home telephone number

8. The gender of the student

9. The student’s name

2.4 Using `awk`

Use `awk` and `ls` to add up the size of all the files in your current directory.

✐