ICCS207: Term I/2018-19

Lecture 4: RegEx and String Processing Tools

Sunsern Cheamanunkul (sunsern.che@mahidol.edu)

Mahidol University

Grep

- grep is a program which will **search** a given set of data and print every line which contains a given pattern.
- grep has many variations:
  - **grep** is used for simple patterns and basic regular expressions
  - **egrep** can handle extended regular expressions. (Same as **grep -E**)
  - **fgrep** is quicker than both grep and egrep, but can only handle fixed patterns
  - **zgrep** is similar to grep but the input must be a compressed file

Grep

It is a truth universally acknowledged, that a single man in possession of a good fortune, must be in want of a wife.
However little known the feelings or views of such a man may be on his first entering a neighbourhood, this truth is so well fixed in the minds of the surrounding families, that he is considered the rightful property of some one or other of their daughters.
"My dear Mr. Bennet," said his lady to him one day, "have you heard that Netherfield Park is let at last?"
Mr. Bennet replied that he had not.

grep "man" input_file

It is a truth universally acknowledged, that a single **man** in possession
However little known the feelings or views of such a **man** may be on his
by a young **man** of large fortune from the north of England; that he came

egrep

- Extended Regular Expression

grep -E "family|families" input_file

of the surrounding families, that he is considered the rightful property

Grep

- Pattern can be a regular expression (RegEx)

```
grep "\si.*d\s" input_file
```

first entering a neighbourhood, this truth **is so well fixed** in the minds
of the surrounding families, that he **is considered** the rightful property
"But **it is," returned she; "for Mrs. Long has just been here, and** she
"Do you not want to know who has taken **it?" cried** his wife impatiently.
down on Monday **in a chaise and four to see the place, and** was so much
delighted with **it, that he agreed** with Mr. Morris immediately; that he

RegEx

- Regular expressions are useful in searching and extracting text-based information from documents.
- Involves searching for one or more specific patterns
- Found in many other programming languages.

Anchor

^ and $

```
^The        matches any string that starts with The
end$        matches a string that ends with end
^The end$   exact string match (starts and ends with The end)
roar        matches any string that has the text roar in it
```


Quantifiers

*** + ? {}**

```
abc*        matches a string that has ab followed by zero or more c
abc+        matches a string that has ab followed by one or more c
abc?        matches a string that has ab followed by zero or one c
abc{2}      matches a string that has ab followed by 2 c
abc{2,}     matches a string that has ab followed by 2 or more c
abc{2,5}    matches a string that has ab followed by 2 up to 5 c
a(bc)*      matches a string that has a followed by zero or more copies of the
sequence bc
a(bc){2,5}  matches a string that has a followed by 2 up to 5 copies of the
sequence bc
```

OR operator

| or []

```
a(b|c)      matches a string that has a followed by b or c
a[bc]       same as previous
```


Character classes

\d \w \s and .

```
\d          matches a single character that is a digit
\w          matches a word character (alphanumeric character plus underscore)
\s          matches a whitespace character (includes tabs and line breaks)
.           matches any character
```

```
\D          matches a single non-digit character
\S          matches a non-whitespace character
```


More on bracket expressions

[]

```
[abc]           matches a string that has either an a or a b or a c -> is the same as a|b|c
[a-c]           same as previous
[a-fA-F0-9]     a string that represents a single hexadecimal digit, case insensitively
[0-9]%          a string that has a character from 0 to 9 before a % sign
[^a-zA-Z]       a string that has not a letter from a to z or from A to Z. In this case the ^ is used as negation of the expression
```


RegEx Examples

- ^#.*
- datafile[0-9]+\.txt
- -{0,1}[0-9]+
- -{0,1}[0-9]*\.[0-9]+
- ^Sunsern
- ^Sunsern$

Sed

- The sed stream editor is a text editor that performs editing operations on information coming from standard input or a file.
- Sed edits line-by-line and in a non-interactive way.

```
sed [options] commands [file-to-edit]
```


sed

- Most well-known for search-and-replace command

```
sed 's/man/woman/g' input_file
```

It is a truth universally acknowledged, that a single woman in possession of a good fortune, must be in want of a wife.
However little known the feelings or views of such a woman may be on his first entering a neighbourhood, this truth is so well fixed in the minds

sed

```
sed 's/[A-Za-z]*ing/(&)/g' input_file
```

It is a truth universally acknowledged, that a (sing)le man in possession of a good fortune, must be in want of a wife.
However little known the (feeling)s or views of such a man may be on his first (entering) a neighbourhood, this truth is so well fixed in the minds of the (surrounding) families, that he is considered the rightful property

cut

- cut is a quick and dirty utility that comes in handy across all sorts of scripting.
- Useful for extracting column data.

```
cut [-d] [-b|c|f] files
```

cut

a,b,c
1,2,3

```
cut –d, –f1,3 input_file
```

a,c
1,3

tr

- tr command translates certain characters into certain other characters

```
tr string1 string2
```


tr

a,b,c
1,2,3

```
tr "123" “456" < input_file
```

a,b,c
4,5,6

tr

a,b,c
1,2,3

```
tr '\n' ',' < input_file
```

a,b,c,4,5,6,

A Note On Input and Output Files

Don't do this

```
cat somefile.txt | tr "\015" "\012" > somefile.txt
```

Do this

```
cat $1 | tr "\015" "\012" > /tmp/$1.$$
mv /tmp/$1.$$ $1
```

MUIC: File Processing

References

- https://medium.com/factory-mind/regex-tutorial-a-simple-cheatsheet-by-examples-649dc1c3f285
- https://regex101.com/
- https://www.digitalocean.com/community/tutorials/the-basics-of-using-the-sed-stream-editor-to-manipulate-text-in-linux
- http://www.grymoire.com/Unix/Grep.html