CSCI 3308 Software Development Methods and Tools
Instructor: David Graham

Lab 2 - Regular Expression

—

TA: Yawen Zhang

Lab 2 - Regular Expressions

Objectives

- Use regular expressions with common Unix commands
- Practice using some useful Unix commands
- Practice creating and running bash shell scripts
- Practice using pipes

Lab Link

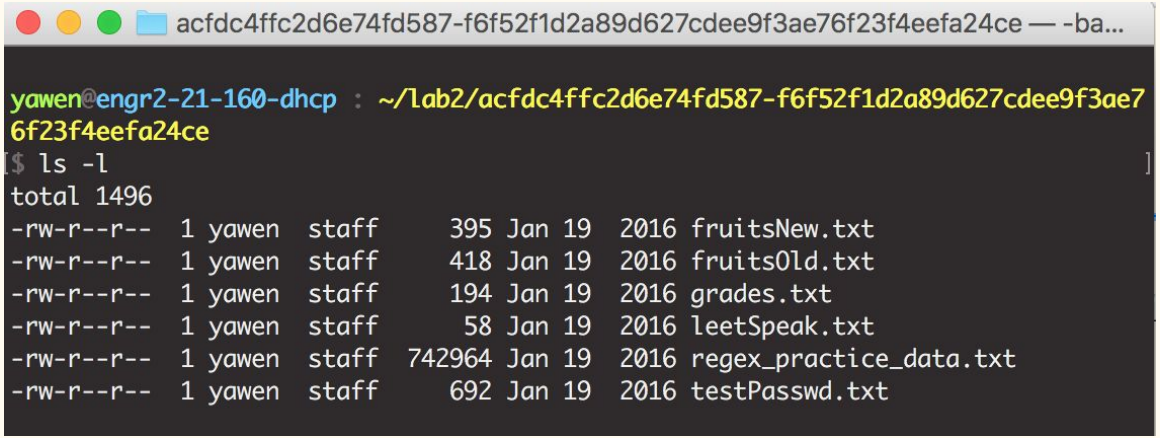
- http://www.lousymedia.com/csci-3308/labs/lab-2

Preparation: download practice files (step 1)

curl -L https://gist.github.com/dgraham/acfdc4ffc2d6e74fd587/archive/f6f52f1d2a89d627cdee9f3ae76f23f4eefa24ce.zip > lab2.zip

unzip lab2.zip -d lab2

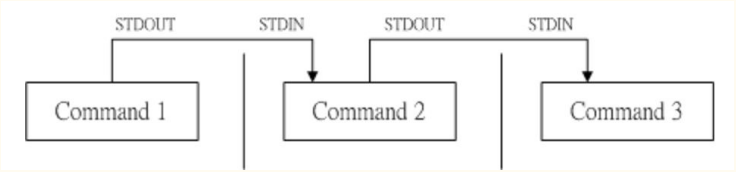
cd lab2/acfdc4ffc2d6e74fd587-f6f52f1d2a89d627cdee9f3ae76f23f4eefa24ce



Practice Unix Commands

- **diff :** diff file1 file2
- **wc :** wc -l file1
- **cut :** cut -d : -f 3 file1
- **pipe :** cut -d : -f 3 file1 | sort > file2
- **sed :** sed s/yourname/myname/g file1
- **awk :** awk 'NR > 1{print $1}' file1
- **grep :** grep -c **'^[0-9]'** file1

General Unix command format:
command –option1 argument -option2 argument ...



awk (step 7)

- a little more difficult, but very powerful text processing command
- work with lines in a file
- build-in variables: NR, NF(number of fields), FS (field separator, space by default)

```
Syntax:


awk '/search pattern1/ {Actions}
     /search pattern2/ {Actions}' file
```

awk (step 7)

- awk Example 1. Default behavior of awk
- command: **awk ‘{print;}’ grades.txt**

awk (step 7)

- awk Example 2. Print only specific field
- command: **awk ‘{print $1“ ”$2}’ grades.txt**

awk (step 7)

- awk Example 3. Print the lines which matches with the pattern
- command: **awk '$3 >= 1{print $1“ ”$2}' grades.txt**

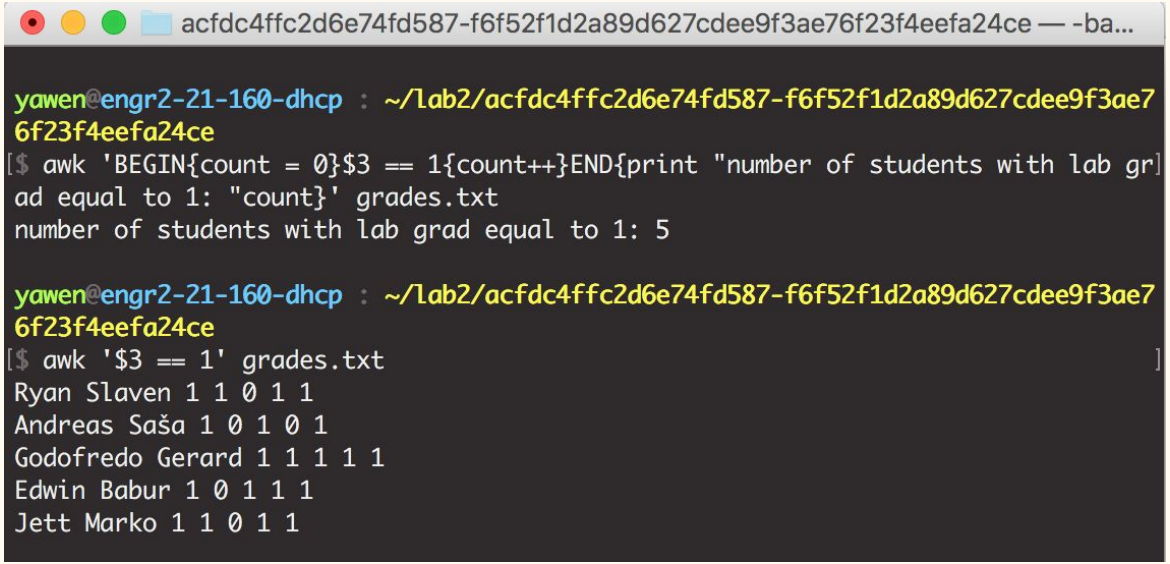
awk (step 7)

- awk Example 4. Initialization and Final Action (BEGIN + END)
- command: **awk 'BEGIN{count = 0}$3 == 1{count++}END{print "number of students with lab grad equal to 1: "count}' grades.txt**

```
Syntax:

BEGIN { Actions}
{ACTION} # Action for everyline in a file
END { Actions }

# is for comments in Awk
```



awk (step 7)

- awk Example 5. Using build-in variables
- command: **awk 'NR > 1{print $1" "$2}' grades.txt**

Basic Syntax in RE (step 8)

abc...	Letters
123...	Digits
\d	Any Digit
\D	Any Non-digit character
\w	Any Alphanumeric character
\W	Any Non-alphanumeric character
\s	Any Whitespace
\S	Any Non-whitespace character
[abc]	Only a, b, or c
[^abc]	Not a, b, nor c
[a-z]	Characters a to z
[0-9]	Numbers 0 to 9

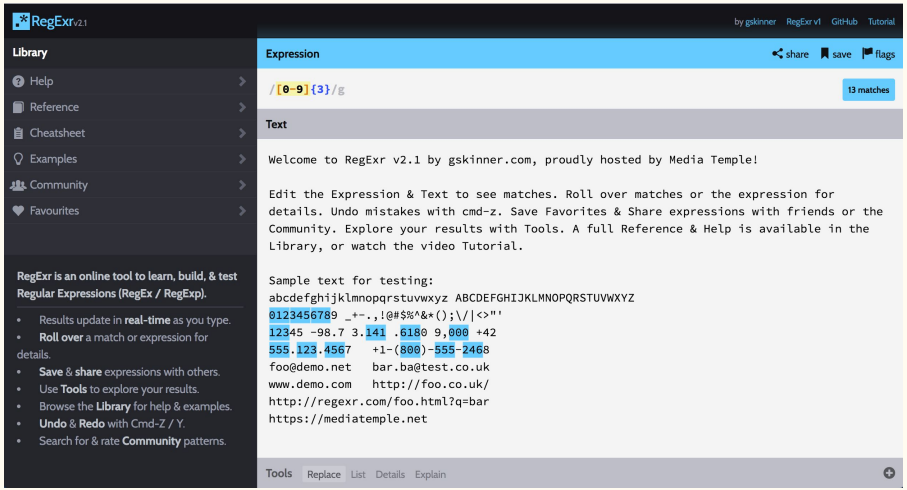
{m}	m Repetitions
{m,n}	m to n Repetitions
.	Any Character (one)
*****	Zero or more repetitions
+	One or more repetitions
?	Optional character
\.	Period
^a	Starts with a
a$	Ends with a (the end of a string)
(abc)	Capture Group
(a(bc))	Capture Sub-group
(abc|def)	Matches abc or def

- **On-line testing with your regular expression commands**

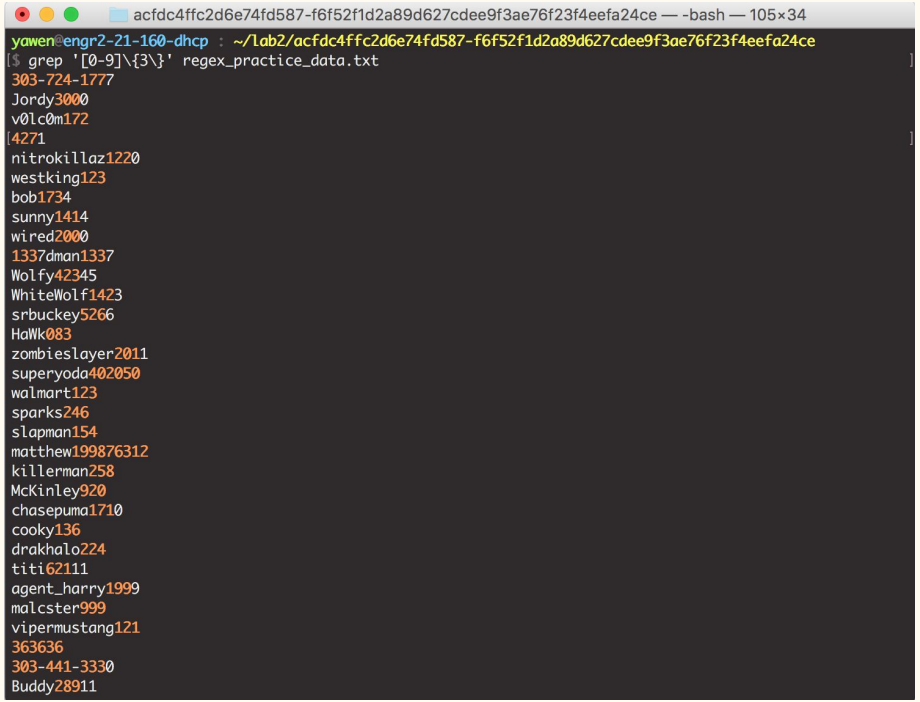
http://regexr.com/

Basic Syntax in RE (step 8)

Testing with RE: **[0-9]{3}**



With grep command: **[0-9]\{3\}**



Lab Task

- Task: **step 1 - 8**
- For submission, prepare a .txt file (you may try editing it with **vim** or **any text editor** you like)
- Save all the commands you use for **step 2 - 8** in the .txt file
- When you **finish**, show me your **.txt file** and get grade for the lab

Search the Internet !

when you get puzzled