

Lecture 9 (2-2-26)

Pipelines, Regex, Filters (continuation from last lecture)

Motivating Questions

1. Why is the pipeline pattern so powerful?
 2. How do we use regular expressions to match text?
 3. How do we translate characters?
 4. How do we extract fields?
 5. How do we search for patterns?
 6. How do we modify text streams?
 7. How do we perform more complex text processing?
-

What is a Unix Pipeline?

- A problem is broken up into separate steps (input, processing, ..., output)
 - so then you compose the "smaller blocks" to work together to accomplish the bigger task
 - also for **concurrency**
 - A **process** is a running instance of a program
-

Regular expression are "filters" you can use on plain text to extract information.

- regex is run by **state machines** (will be covered in Theory of Computing)
- if you want to use extended regex in grep you need to use `-E` flag.
 - EX: `ps aux | grep -E '(bash|zsh)`
 - if you add `-o` then you can see just instances of `bash` and `zsh`
 - EX: `ps aux | grep -Eo '(bash|zsh)' | sort | unique`
 - this will show the unique instances
 - EX of regex:
 - `echo "SAMUEL" | grep -E "."`
 - matches any character one time

- `echo "Samuel" | grep -E ".*"`
 - matches 0 or more times
 - `echo "Samuel" | grep -E '.'`
 - matches 1 or more times
 - `echo "Samuel" | grep -E '[aeiou]*'`
 - match all vowels
 - `echo "Samuel" | grep -E '[^aeiou]'`
 - match instances of NOT vowels
 - `echo "Aynaz" | grep -Ei '^a'`
 - matches anything that starts with an A (`-i` makes it case insensitive)
 - `echo "Aynaz" | grep -E 'z$'`
 - matches anything that ends with lowercase z
 - `echo "Aynaz" | grep -E '(.)\1'`
 - this says match the first character and capture it, then keep going until you see it again
-

Examples of regex : assume `pokemon.txt` has a list of pokemon names (no spaces):

```
cat pokemon.txt | grep -E '^ch'
```

-finding all instances of lines that start with `ch`

```
cat pokemon.txt | grep -E 'tt'
```

- this is all lines that contain consecutive t's

```
cat pokemon.txt | grep -E '[^aeiou]'
```

- this is all lines that don't start with a vowel

```
cat pokemon.txt | grep -E '[aeiou]{2}'
```

- finds all lines that have two consecutive vowels

```
cat pokemon.txt | grep -E '(.)\1'
```

- finds all lines with two consecutive letters

```
cat pokemon.txt | grep -E '^(\.)\1$'
```

- finds all lines of that begin and end with the same letter

```
cat pokemon.txt | grep -E '^[^rst]*[rst][^rst]*[rst][rst]*$'
```

- matches any lines that has r,s, or t **exactly** two times