University Of Dundee

MSc. Data Science

Student: Mike Hudgell (140000254)

Subject Matter: Erlang

Hi Andy,

The Purpose of this note is to try and help explain the various files being submitted. I’m assuming marking 17 x 3 assignments is a bit of a nightmare.

My code for each assignment is in week1.erl, week2.erl, week3.erl. Week1 is the Pi estimate. Week2 is the sorting and counting. Week3 is the multithreaded character frequency counter.

Week1 is fairly straight forward, public call is main(), and sorry about all the diagnostics!

**88> week1:main().**

**3.1415826535897198 ok**

**89>**

Week2 again fairly straight forward, call is main([your list]). For example:

**86> week2:main([1,2,3,4,5,1,2,3,4,5]).**

**[1,2,3,4,5] 5 ok**

**87>**

Week2x is the bonus points question, the frequency of words in Hamlet.txt. I came back to this after week3 but have run out of time. The code at the moment gives you a character count and a word count. It strips almost all of the punctuation away and gives you a list of words. Each word is then given a key or token, that is all, cutting the grass took priority I’m afraid. Not necessarily my priority list you understand! Obviously the next thing to do is to sort the keys/tokens, and then go through counting occurrences and hopefully joining it all up into a nice readable result. The file e2.txt was my test file for this. I will finish this, might be fun to run it against my essays!

**93> week2x:wordcount('e2.txt').**

**[{2014,3,23},{20,32,5}] 'Start'**

**[{2014,3,23},{20,32,5}] 'Loaded'**

**[{2014,3,23},{20,32,5}] 'Number of Characters ' 63**

**[{2014,3,23},{20,32,5}] 'Number of Words ' 12**

**[{2014,3,23},{20,32,5}] 'Unsorted List of Words '**

**one**

**two**

**three**

**four**

**one**

**ten**

**5**

**six**

**words**

**well**

**ten**

**then**

**[{2014,3,23},{20,32,5}] 'List of Words with keys '**

**[{2014,3,23},{20,32,5}] [[634,[111,110,101]],[687,[116,119,111]],[1575,[116,104,114,101,101]],[1131,[102,111,117,114]],[634,[111,110,101]],[648,[116,101,110]],[53,[53]],[685,[115,105,120]],[1658,[119,111,114,100,115]],[1077,[119,101,108,108]],[648,[116,101,110]],[1067,[116,104,101,110]]]**

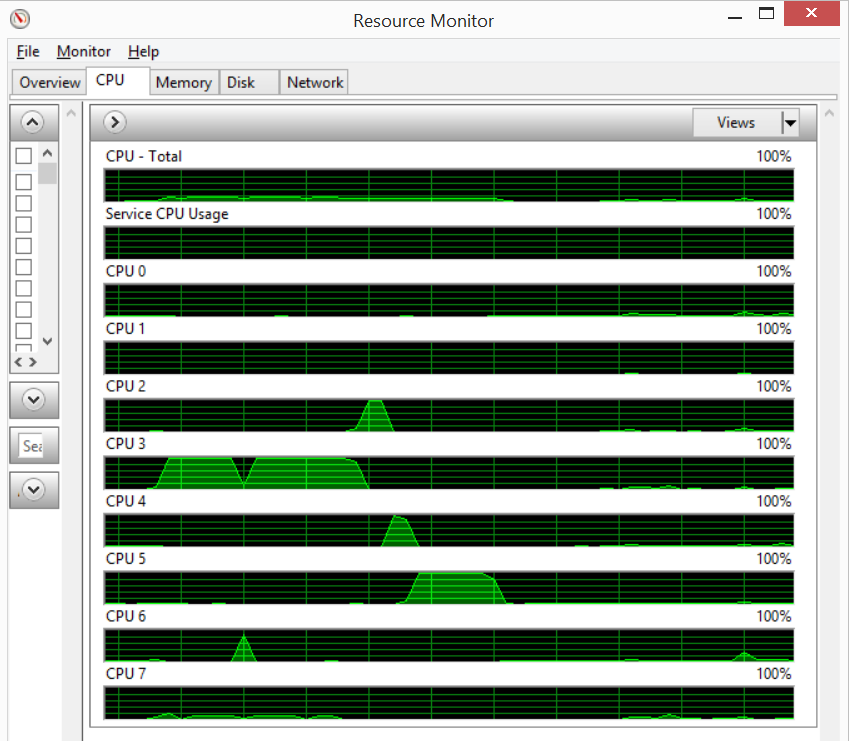
**[{2014,3,23},{20,32,5}] 'Complete'**

**ok**

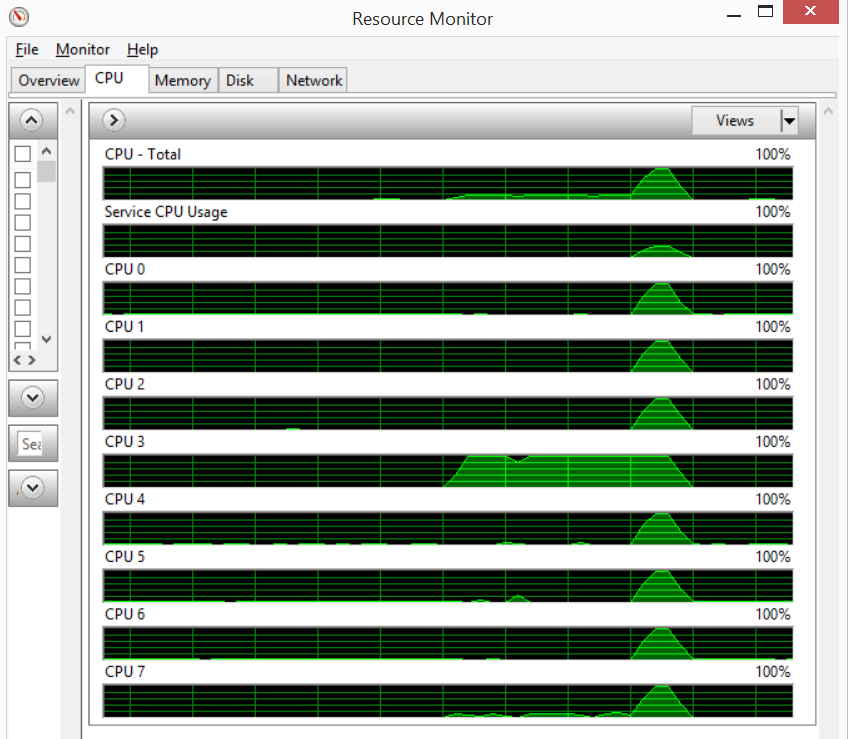
Week3 was turning your charcount into a multi-threaded/multi-processor piece. Cannot tell you how chuffed I was when I got this working! What I did to begin with was create charcount1. This is your routine but with loads of comments and diagnostics in. I also created easy.txt to run against, so that I could see what was going on.

Anyway on my machine ccharcount processes war.txt in 27 seconds and my new routine processes it in 14. This is end to end, load to result. Much better evidence though is the resource monitor.

Charcount:



My routine.



Anyway onwards to exam prep…

Kind regards,

Mike.