Faculty of Science, Engineering and Technology

Fundamentals of Data Management

Credit Tasks 1.2: Advanced Regular Expressions

Overview

As a first step, install the VMWare Player and open the Ubuntu environment provided on Blackboard. Then work through the text processing tasks and document your answers to the questions.

Purpose

Learn how to use command-line tools to manipulate text without the need for an editor. Use regular expressions to search for different variations of text.

Task

Open the VM provided in a VMWare Player. Use command line tools and regular expressions to find strings in files.

Time

This task should be completed in your first lab class and submitted for feedback at the end of lab 1 or the beginning of lab 2.

Resources

- Online modules (from Blackboard)
- Regular Expressions tutorial: http://www.regular-expressions.info/
- Free VMWare Player, available:

https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_player/7_0

- Video demonstration on Blackboard (vm-intro.mp4)

Feedback

Discuss your solutions with the tutorial instructor.

Next

Get started on module 2.

Credit Tasks 2 — Submission Details and Assessment Criteria

Document your solutions to the tasks in a report using a word processor and upload to Doubfire as pdf. Your tutor will discuss the tasks with you have uploaded them.

Subtask 1.2.1

IPv4 addresses cover the range 0.0.0.0 to 255.255.255.255. (There are four numbers, separated by dots (full stops), each number is between 0 and 255.)

138.168.96.3 is a valid IP address, but 256.168.93.6 is not, neither is 255.355.0.199.

To solve this problem, you have to specify alternative combinations of characters: If the first two numbers are 25, the following number cannot be higher than 5. If the first number is 2, but the second number is lower than 5, the following number can be 0-9, etc.

How do we do such alternative combinations? This example matches days in a month (where we can have 0-31; no heed is being paid to the month):

(3[01]|[012]\d) - this option enforces a leading zero

- | stands for 'or'
- \d is the same as [0-9]

(3[01]|[12]\d|\<0?\d\>) - this option ensures that leading zeros can be omitted (why do we need \< and \>?)

Given this example, write an expression that matches IP addresses. Test on IPaddresses.txt. You should obtain 12 results.

Test how many IP addresses you can find in access.log. There should be 967.

Document your answer and upload.

.