Linux Classroom Series – 11/Sept/2020

Regular Expressions

- They are also referred as *regex, RegEx*
- Truth about Regex:

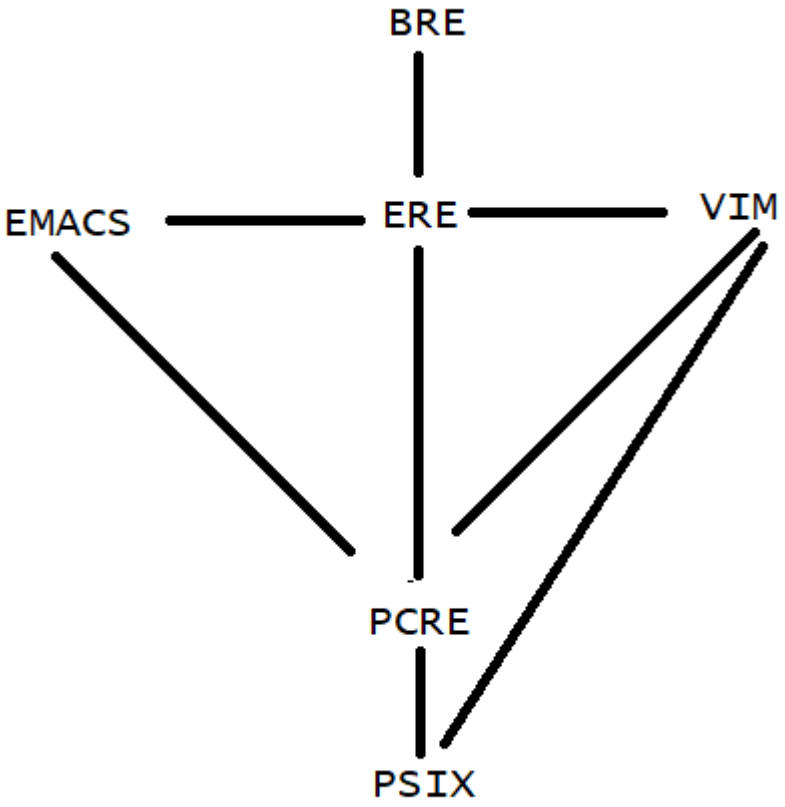
1. Most of the cases regex is defined as *A regular expression is a declarative specification of describing the textual structure to match string*

2. The problem with above definition is regular expressions are not declarative, A regular expression is imperative. Regular expression is subroutine/function/method

- In What Language we write regular expressions. In Regular expressions we have six major dialects

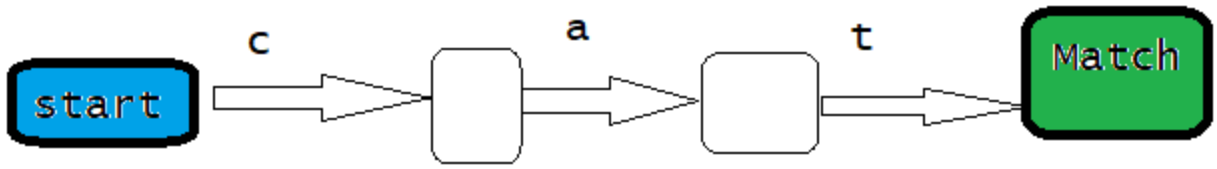
1. BRE:
   - This is basic regular expressions.
   - Tools: ed, sed, grep
2. ERE:
   - GNU extended regular expressions
   - Tools & Languages : egrep, gawk, Notepad++, TCL
3. EMACS:
   - This is Emacs regular expressions
   - Tools: Emacs
4. VIM:
   - TOOLS: VIM.
5. PCRE:
   - This is PERL(5) compatible regular expressions
   - TOOLs & Languages: PERL, .NET, APACHE, C#, Java, JavaScript, PHP, Powershell, Python, R, Ruby,
6. PSIX:
   - Perl 6 Regular Expressions
   - Languages: Perl 6

- These dialects have similarities & dissimilarities. There are relationships b/w



  dialects
- How are regular expressions implemented?
  - Theoretically Regular expression are implemented on a Finite State machine(FSM). But Languages Practically implement regular expressions on stack-based machine.
  - To understand regular expressions we will be using FSM
  - To search for a word cat in the text sequence in all six dialects mentioned above the regex is */cat/*

- Lets represent cat in transition graph



- If the above regular expression is represented by code

```
for(index=0; index<len(message); index++) {
    match_position = index

    try {

        message[match_position] == 'c' or
throw Backtracking
        match_position++;

        message[match_position] == 'a' or
throw Backtracking
        match_position++;

        message[match_position] == 't' or
throw Backtracking
        match_position++;

       return TRUE;




    }
}
return FALSE;
```