Lecture 8

Regular expressions and grep

COP 3344 Introduction to UNIX
Fall 2007

Acknowledgment: These slides are modified versions of Prof. Sudhir Aggarwal's slides

Regular Expressions and Wildcards

- Many Unix utilities use regular expressions
  - They use special characters in a different manner than are used by the shell for file name expansion
- Concatenation
  - This is given by a sequence of character

  `abc` matches the character a followed by b followed by c
- * operator
  - Indicates zero or more instances of the preceding character
    - Can also be used after a group enclosed in parentheses *( )*

  `ab*c` matches `ac`, `abc`, `abbc`, etc

  `(ab)*c` matches `c`, `abc`, `ababc`, etc

  `(ab*)*c` matches `c`, `abc`, `abbc`, `ababc`, `ababbc`, etc
- + operator
  - Matches one or more instances of the preceding character

Matching from a Set of Characters

- `.` matches any single character except newline

  `a.b` matches a followed by any character, then b, for example `adb`
- `[]` is used to indicate one of a set of characters
  - The `-` is used to define a range
  - A `^` after `[` means match anything not in the set

  `[adkr]`matches a, d, k, r

  `[0-9]`matches any decimal digit

  `[a-z]` match any lower case letter

  `[^aeiou]` matches any character except a vowel

  `[^0-9]` matches any character except a decimal digit

Anchors

- Anchors ^ and $ can be used to indicate that a pattern will only match when it is at the beginning or end of a line respectively
  - Note: This use of ^ is different from its use in `[^...]`

  `^alpha` matches `alpha` only when it is at the beginning of the line

  `[A-Za-z]+$` matches a word consisting of lower and upper case letter, which occurs at the end of a line

  `^alpha*zeta$` matches `alph` at the start of a line, followed by an number of a followed by `zeta` and end of the line

Alternation and Grouping

- Use the | character to choose between alternatives

  `a|b` matches a or b

  `a*|b` matches any number of a's or a single b
- Use parentheses are for grouping

  `(ab*a)*` matches any number of `ab*a`
    - Example: `aba`, `abaaba`, `abbbaaba`, etc

grep and *egrep*

- *grep* searches for strings in files that match a regular expression and prints out the lines that contain these matches to stdout
  - If no file is specified, then *grep* uses stdin
- General form

  `grep [options] pattern [files]`
- *egrep* extends the syntax of regular expressions
  - Generally *grep* does not support the parentheses, the + operator, the | operator or the ? operator (zero or one occurrence)
  - The `-E` flag in *grep* generally gives egrep behavior

grep

- Examples
  - `grep int *.c` (find all occurences of the pattern `int` in all files with a `.c` extenstion)
  - `grep 'main()' testprog1.c` (enclosing the pattern in quotes is useful when using special characters)
  - `grep 'm.*n' myfile` (the `.` matches a single character, the `.*` matches any number of characters; this finds anything starting with an `m` and ending with an `n`)
- `grep` has many options
  - `-c` print number of lines matched
  - `-i` ignore case
  - `-n` display the line numbers
  - `-l` display only names of files with matched, and not actual lines
  - `-P` pattern is a Perl regular expression
  - `-v` output lines that do not match
  - `-w` match entire words
- Read the following tutorial for more help
  http://www.panix.com/~elflord/unix/grep.html

grep Examples

- `grep alpha junk` looks for `alpha` in file `junk`
- `grep "ii*" junk` looks for a string of one or more `i`'s
- `grep ^begin junk` looks for a line that starts with `begin`
- `grep receive *.sh` looks for `receive` in any file ending in `.sh`
- `grep "[abc].*" junk` looks for a string with an `a`, `b`, or `c`, followed by any number of other characters