CEN302 Sistem Programlama

Lecture 5

Basic vs. Extended REs

- In basic regular expressions the metacharacters ?, +, {, }, (,), |, and) have no special meaning (`grep`)
  - To give them special meaning, use the escaped versions: \?, \+, \{, \}, \(, \), and \|
- When using extended regular expressions, these metacharacters have special meaning
  - `grep –E = egrep`

Example:

file1.txt content: File1
File2
File3

- grep “????1” file1.txt -> returns nothing

  Use escape char: grep “\?\?\?\?1” file1.txt -> returns File1
- grep “F???[0-9]” file1.txt -> returns nothing

  Use escape char: grep “F\?\?\?[0-9]” file1.txt -> returns File1
  File2

Using egrep

- `egrep pattern filename(s)`
- To be safe, put quotation marks around your pattern
- Examples:
  - `egrep "abc" textfile`

    (print lines containing “abc”)
  - `egrep -i "abc" textfile`

    (same, but ignore case)
  - `egrep -v "abc" textfile`

    (print lines not containing “abc”)
  - `egrep -n "abc" textfile`

    (include line numbers)
  - `egrep -c "abc" textfile`

    (print a count of lines containing “abc”)

Example

Phone_book.text:

```
Alice Chebba    973-555-2015
Barbara Swingle 201-555-9257
Jeff Goldberg   201-555-3378
Liz Stachiw     212-555-2298
Susan Goldberg  201-555-7776
```

Try all the egrep options on this file.

Metacharacters

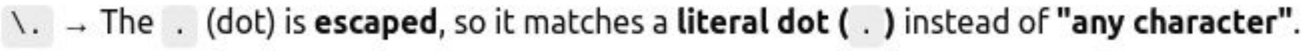
- Period (.): matches *any* single character
  - “a.c” matches abc, adc, a&c, a;c, …
  - “u..x” matches unix, uvax, u3(x,…
- Asterisk (*): matches zero or more occurrences of the previous RE
  - not the same as wildcards in the shell!
  - “ab*c” matches ac, abc,abbc, abbbc,…
  - “.*”matches any string

Metacharacters (cont.)

- Plus (+): matches one or more occurrences of the preceding RE
  - “ab+c” matches abc, abbc, but not ac
- Question mark (?): matches zero or one occurrence of the preceding RE
  - “ab?c” matches ac, abc but not abbc
- Logical or (|): matches RE before or RE after bar
  - “abc|def” matches abc or def

Metacharacters (cont.)

- Caret (^): means beginning of line
  - "`^D.*`" matches a line beginning with D
- Dollar sign ($) means end of line
  - "`.*d$`" matches a line ending with d
- Backslash (\ ): escapes other metacharacters
  - "`file\.txt`" matches `file.txt` but not `file_txt`



`\.` → The `.` (dot) is **escaped**, so it matches a **literal dot (`.`)** instead of **"any character"**.

Metacharacters (cont.)

- Square brackets ([]): specifies a set of characters as a list
  - any character in the set will match
  - ^ before the set negates the set
  - - specifies a character range
  - Examples:
    - “[fF]un” matches fun, Fun
    - “b[aeiou]g” matches bag, beg, big, bog, bug
    - “[A-Z].*” matches a string starting with a capital letter
    - “[^abc].*” matches any string not starting with a, b, or c

Metacharacters (cont.)

- Parentheses (`()`): used for grouping
  - “`a(bc)*`” matches a, abc, abcbc, abcbcbc, …
  - “`(foot|base)ball`” matches football or baseball
- Braces (`{}`): specify the number of repetitions of an RE
  - “`[a-z]{3}`”matches three lowercase letters
  - “`m.{2,4}`” matches strings with m followed by between 2 and 4 characters

Examples:

1. egrep ”^B.*s$” file.txt
2. egrep ”[0-9]{3}” file.txt
3. egrep ”num(ber)? [0-9]+” file.txt
4. egrep ”word” file | wc -l
5. egrep ”[A-Z].*\?” file

Construct egrep commands that find in file:

- Lines beginning with a word of at least 10 characters

Construct egrep commands that find in file:

- Lines beginning with a word of at least 10 characters

```
egrep "[a-zA-Z]{10,}" file.txt
```

Construct egrep commands that find in file:

- Lines Containing a Student ID Number in Standard 3-Part Form
- Example: 11-33-22

Construct egrep commands that find in file:

- Lines Containing a Student ID Number in Standard 3-Part Form
- Example: 11-33-22

```
egrep “[0-9]{2}-[0-9]{2}-[0-9]{2}” file.txt
```

Construct egrep commands that find in file:

- Number of lines not ending in an alphabetic character

Construct egrep commands that find in file:

- Number of lines not ending in an alphabetic character

```
egrep -c “[^a-zA-Z]$” file.txt
```

Construct egrep commands that find in file:

- Find words that contain all five vowels in alphabetical order
- Example: affectious, facetious

Construct egrep commands that find in file:

- Find words that contain all five vowels in alphabetical order
- Example: affectious, facetious

```
egrep “^[^aeiou]*a[^aeiou]*e[^aeiou]*i[^aeiou]*o[^aeiou]*u[^aeiou]*$” file.txt
```