What is a Regular Expression?

- A **regular expression** (RE) is a string of **characters** that specifies a set of strings
- Each of these strings is said to **match** the regular expression
- **Pattern matching** is useful in many real-world situations:
  - searching for a file on the file system
  - finding and replacing text in a file
  - extracting data elements from a database

Unix programs that use REs

- `grep` (search within files)
- `egrep` (`grep` with extended REs)
- `vi`/`emacs` (text editors)
- `sed` (stream editor)
- `awk` (pattern scanning language)
- `perl` (scripting language)

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`

Using `egrep`

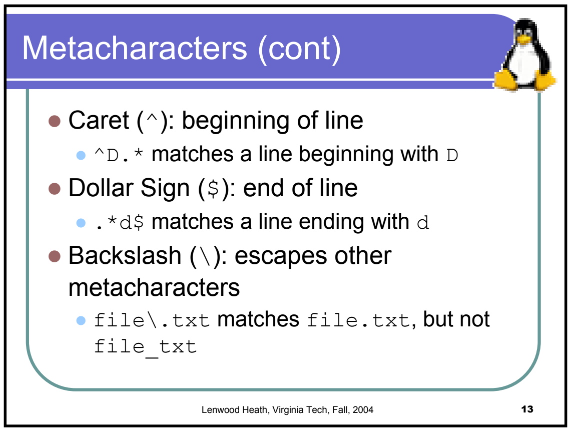
- `egrep pattern filename(s)`
- To be safe, put quotation marks around your pattern
- Examples:
  - `egrep "abc" textfile`
  - `egrep -i "abc " textfile`
  - `egrep -v "abc" textfile`
  - `egrep -n "abc" textfile`

Metacharacters

- Period (`.`): matches ***any*** single character
  - `a.c` matches `abc`, `adc`, `a&c`, and `a;c`
  - `u..x` matches `unix`, `uvax`, and `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`, and `abbbc`
  - `.*` matches any string

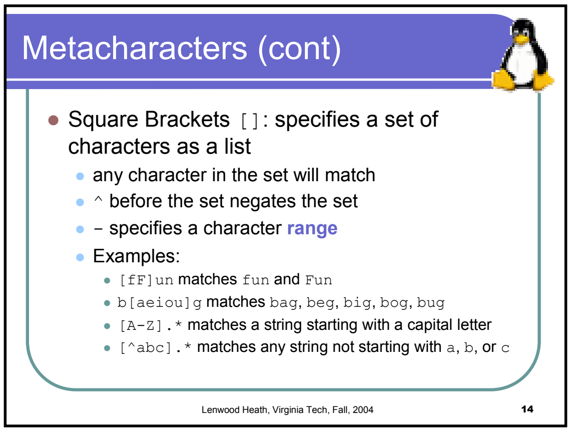
Metacharacters (cont)

- Plus (`+`): matches **one or more occurrences** of the preceding RE
  - `ab+c` matches `abc`, `abbc`, `abbc`, but not `ac`
- Question Mark (`?`): matches **zero or one occurrences** of the preceding RE
  - `ab?c` matches `ac` or `abc`, but not `abbc`
- Logical Or (`|`): matches RE before `|` **or** RE after `|`
  - `abc|def` matches `abc` or `def`



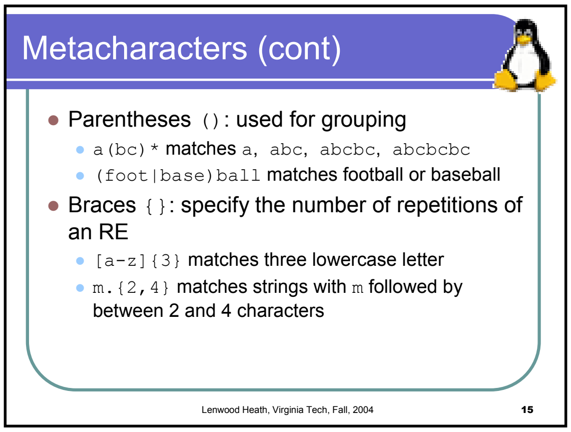
Metacharacters (cont)

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



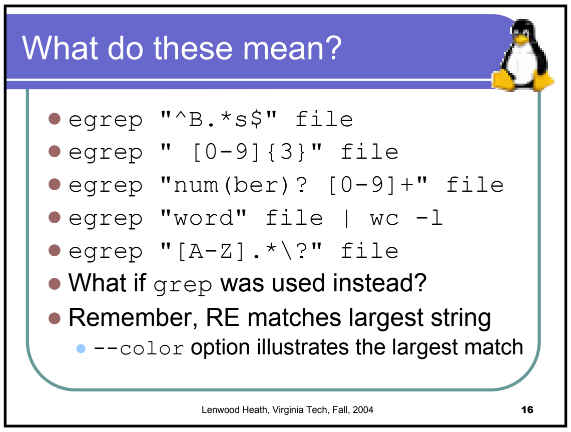
Metacharacters (cont)

- 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` and `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 letter
  - `m.{2,4}` matches strings with `m` followed by between 2 and 4 characters



What do these mean?

- `egrep "^B.*s$" file`
- `egrep " [0-9]{3}" file`
- `egrep "num(ber)? [0-9]+" file`
- `egrep "word" file | wc -l`
- `egrep "[A-Z].*\?" file`
- What if `grep` was used instead?
- Remember, RE matches largest string
  - `--color` option illustrates the largest match