# CSCI 2132
# Software Development

---

## Lecture 7:

## Wildcards and Regular Expressions

Instructor: Vlado Keselj

Faculty of Computer Science

Dalhousie University

# Previous Lecture

- Pipes
- Inodes
- Hard links
- Soft links
- Filename Substitution (Wildcards) (started)

# Filename Substitution (Wildcards)

- Also known as **pathname substitution**
- Used to specify multiple filenames (i.e., pathnames)
- Makes use of "wildcards"; i.e., metacharacters expanded by the shell
- Some wildcard types:
  - `?`: matches any single character
  - `*`: matches any string, including empty string
  - `[...]`: matches any single character in the set
  - `[!...]`: any character except characters from the set
  - we can use ranges with '`-`' in brackets

# File Substitution Examples

- `[0-9]`

# File Substitution Examples

- `[0-9]`: any digit between 0 and 9

# File Substitution Examples

- `[0-9]`: any digit between 0 and 9
- `[a-zA-Z]`

# File Substitution Examples

- `[0-9]`: any digit between 0 and 9
- `[a-zA-Z]`: any English alphabet character

# File Substitution Examples

- `[0-9]`: any digit between 0 and 9
- `[a-zA-Z]`: any English alphabet character
- `[unix]`

# File Substitution Examples

- `[0-9]`: any digit between 0 and 9
- `[a-zA-Z]`: any English alphabet character
- `[unix]`: matches either ‘u’, ‘n’, ‘i’, or ‘x’

# File Substitution Examples

- `[0-9]`: any digit between 0 and 9
- `[a-zA-Z]`: any English alphabet character
- `[unix]`: matches either ‘u’, ‘n’, ‘i’, or ‘x’
- `ls ~/csci2132/lab1/*.java`

# File Substitution Examples

- `[0-9]`: any digit between 0 and 9
- `[a-zA-Z]`: any English alphabet character
- `[unix]`: matches either ‘u’, ‘n’, ‘i’, or ‘x’
- `ls ~/csci2132/lab1/*.java` — list Java files

# File Substitution Examples

- `[0-9]`: any digit between 0 and 9
- `[a-zA-Z]`: any English alphabet character
- `[unix]`: matches either ‘u’, ‘n’, ‘i’, or ‘x’
- `ls ~/csci2132/lab1/*.java` — list Java files
- `ls *.????`

# File Substitution Examples

- `[0-9]`: any digit between 0 and 9
- `[a-zA-Z]`: any English alphabet character
- `[unix]`: matches either ‘u’, ‘n’, ‘i’, or ‘x’
- `ls ~/csci2132/lab1/*.java` — list Java files
- `ls *.????` — list all files with 4-character extension

# File Substitution Examples

- `[0-9]`: any digit between 0 and 9
- `[a-zA-Z]`: any English alphabet character
- `[unix]`: matches either ‘u’, ‘n’, ‘i’, or ‘x’
- `ls ~/csci2132/lab1/*.java` — list Java files
- `ls *.????` — list all files with 4-character extension
- `ls lab[1-9]`

# File Substitution Examples

- `[0-9]`: any digit between 0 and 9
- `[a-zA-Z]`: any English alphabet character
- `[unix]`: matches either ‘u’, ‘n’, ‘i’, or ‘x’
- `ls ~/csci2132/lab1/*.java` — list Java files
- `ls *.????` — list all files with 4-character extension
- `ls lab[1-9]` — list all files with the name consisting of word `lab` and a digit from 1 to 9

# File Substitution Examples

- `[0-9]`: any digit between 0 and 9
- `[a-zA-Z]`: any English alphabet character
- `[unix]`: matches either ‘u’, ‘n’, ‘i’, or ‘x’
- `ls ~/csci2132/lab1/*.java` — list Java files
- `ls *.????` — list all files with 4-character extension
- `ls lab[1-9]` — list all files with the name consisting of word `lab` and a digit from 1 to 9
- `ls [!0-9]*`

# File Substitution Examples

- `[0-9]`: any digit between 0 and 9
- `[a-zA-Z]`: any English alphabet character
- `[unix]`: matches either ‘u’, ‘n’, ‘i’, or ‘x’
- `ls ~/csci2132/lab1/*.java` — list Java files
- `ls *.????` — list all files with 4-character extension
- `ls lab[1-9]` — list all files with the name consisting of word `lab` and a digit from 1 to 9
- `ls [!0-9]*` — list all files which name does not start with a digit

# File Substitution Examples

- `[0-9]`: any digit between 0 and 9
- `[a-zA-Z]`: any English alphabet character
- `[unix]`: matches either ‘u’, ‘n’, ‘i’, or ‘x’
- `ls ~/csci2132/lab1/*.java` — list Java files
- `ls *.????` — list all files with 4-character extension
- `ls lab[1-9]` — list all files with the name consisting of word `lab` and a digit from 1 to 9
- `ls [!0-9]*` — list all files which name does not start with a digit
- `cp lab1.bk/*.java lab1/`

# File Substitution Examples

- `[0-9]`: any digit between 0 and 9
- `[a-zA-Z]`: any English alphabet character
- `[unix]`: matches either ‘u’, ‘n’, ‘i’, or ‘x’
- `ls ~/csci2132/lab1/*.java` — list Java files
- `ls *.????` — list all files with 4-character extension
- `ls lab[1-9]` — list all files with the name consisting of word `lab` and a digit from 1 to 9
- `ls [!0-9]*` — list all files which name does not start with a digit
- `cp lab1.bk/*.java lab1/` — copy Java files from one directory to another

# More Examples

- `ls ~/csci2132/lab1/*.java`
- `ls ~/csci2132/lab1/H????World.java`
- `ls H*`
- `ls [!A-Z]*`
- `ls */*/*.java`
- `ls *.java */*.java`
- `echo .*`
- command `echo` — prints out command line arguments
- `cat *.txt > allfiles`

# Regular Expressions

- Regular Expressions are patterns used to match strings, and thus used in fast and flexible text search
- The name comes from Regular Sets defined by the mathematician Stephen Kleene
- Implemented as DFA (Deterministic Finite Automata) or NFA (Non-deterministic Finite Automata)
- Kleene's notation implemented by Ken Thompson into the editor QED to match patterns
- Thompson later added this to the Unix editor ed
- Eventually led to the command `grep`, coming from ed command `g/re/p` (Global search for Regular Expression and Print matching lines)

# Reading about Regular Expressions

- The Unix book: Chapter 3, Filtering Files (p.84)
- Appendix: Regular Expressions (p.665)
- Regular expressions
  - Patterns used for searching and replacing text
  - Used in many contexts, but we will focus on the grep command
  - There are two kinds of regular expressions: basic regular expressions and extended regular expressions

# Basic Regular Expressions

- Using metacharacters:
- `.`: Matches any single character
- `[...]`: Matches any character between brackets, `-` used to specify range; most other metacharacters loose their "meta-meaning" between brackets
- `[^...]`: Matches any character except one of the characters between brackets
- `*`: 0 or more occurrences of the preceding character
- `^`: Matches the beginning of a line
- `$`: Matches the end of a line
- `\`: Inhibits the meaning of any metacharacter

# BRE Examples

- BRE = Basic Regular Expressions
- One or more spaces: spacespace* (replace space by a space character): ‘ *’
- Empty line: `^$`
- Formatted dollar amount:
  `\$[0-9][0-9]*\.[0-9][0-9]`

# Filters, grep command

- Filter is a program that is mostly used to read stdin, process data, and write to stdout
- Often used as elements of pipelines
- One such program is `grep`
- grep reads a file or stdin and outputs lines matching a regular expression
- grep syntax

```
grep [options] BRE [file(s)]
```

# Example

```
Chocolate $1.23 each
Candy $.56 each
Jacket $278.00</pre>
<pre>$44.00
$44
```

If we enter the following command

```
grep '\$[0-9][0-9]*\.[0-9][0-9]' price
```

The output will be the following three lines:

```
Chocolate $1.23 each
Jacket $278.00
$44.00
```

# One more grep example

- We will use the dictionary file:
  `/usr/share/dict/linux.words`
- Write a grep command to find 5-letter words that start with ‘a’ or ‘b’ and end with ‘b’
- Write a grep command to find all words starting with ‘a’ or ‘b’ and ending with ‘b’
- How many are there?

# Similarity between Wildcards and Regular Expressions

- We can get similar results with wildcards and regular expressions; e.g.:

```
ls *.java
ls | grep '\.java$'
```

- List of all files in `/bin`, whose names contain exactly one minus sign (−):

```
ls /bin | grep '^[^-]*-[^-]*$'
```