CS3157: Advanced Programming

Lecture #3

Sept 14

Shlomo Hershkop

shlomo@cs.columbia.edu

Outline

- Feedback
- Regular Expressions
- Graphics
- Library Modules
- Creating a library

- Reading: Chapter 4,5 (pg-167)

Feedback from last class

- More code
- Slides online
- Pace of course

3

Announcement

- Bill gates will be on campus October 13, tickets will be available fcfs basis
- Keep eyes open

4

Homework 1

- Download from webpage www.columbia.edu/~cs3157
- use perl in a practical project
- Learn about computer security

5

Computer Security

- System and theory of ensuring the confidentiality, integrity, availability, and control of electronic information and systems.
  - Network
  - Host
  - Data

6

For host based security

- Want to ensure permission system
  - X should only be allowed to do A, B, and C
- Want to ensure accountability
  - If Y does something not allowed, should be noted
- Want to be able to track
  - If something has been tampered with, how can we locate it
  - Both preventative and reactionary

7

Project

- Assuming you are a system administrator or just paranoid
- Take chronological snapshots of your system to compare and find changes
  - Many changes by system
  - Many changes by valid user
  - Might locate malicious user/system changes

8

Useful tips

- Can turn on warning to help prevent errors
- Run in strict mode to catch potential mistypes
- Create debugging statements to help chart progress throughout program…
- Better yet, learn to use the perl debugger (next week).

9

Doing the work

- Find a good perl environment
- Read up on perl
- Can work
  - Home
  - Clic lab
  - Home, remote on clic machine

10

TOOLS: VNC

- www.realvnc.com
- Start server on a clic machine:
  - vncserver
  - Run client on your side
  - demo

11

Regular Expression

- Review
- new examples

12

Regular Expressions

- simplest regular expression is a literal string
- complex regular expressions use *metacharacters* to describe various options in building a pattern.

\	escapes the character immediately following it
.	matches any single character except newline
^	matches at the beginning of a string
$	matches at the end of a string
*	matches the preceding element 0 or more times
+	matches the preceding element 1 or more times
?	matches the preceding element 0 or 1 times
{ ... }	specifies a range of occurrences for the element preceding it
[...]	matches any one of the class of characters in the brackets
(...)	groups expressions
|	(pipe) matches either the expression before or after it

13

Basic

- The most basic match is:
  - $string =~ m/sought_text/;
  - Will return true if sought_text is part of string, false otherwise
  - Perl assume m/???/ when use /???/

1) if (/shlomo/) {….}
2) if (not /shlomo/) {…}
   if($_ !~ /shlomo/) {….}

14

Basic II

1. if ($a =~ /s|h|m/) {…..}
2. if ($a =~ /[a-z]/) {…}

3. How would we look for a phone number?

4. What about a social security?

15

Pattern attributes

- operators:
  - m/pattern/gimosx : match
- g = match globally (all instances)
- i = do case insensitive matching
- e = evaluate right side as an expression
- s = let . match newlines
- m = $ and ^ can refer to inside newlines

16

groups

To allow groups of alternative choices

```
if($string =~ /(A|E|I|O|U|Y)/i)
  { print "String contains a vowel!\n“; }
```

Alternatively we can also specify character choices:

```
if( $string =~ /[AEIOUY]/i )
  { print "String contains a vowel!\n“;  }
```

Can also specify ranges

```
if( $string =~  /^[a-e]/I ) {
     something
}
```

17

Groups II

- To allow us to reference for selection and subsitution
- Each group can be referred to by scalar $1, $2, $3 ….

Example

- “From s@aol.com Wed Jun 3 12:12:12 2005”
- If(/^From (.*) (…) (…) (.*)$/)

18

shortcuts

```
$name = “advanced programming class”

if($name =~ /programming/){
print $`  ;
print $&  ;
print $’  ;
}
```

19

subsitutions

- s/pattern/pattern/

20

transliteration

- tr/search_list/replacement_list/

- -c all characters not in the search list
- -d anything without replacement …delete
- -s squash dublicates

21

quantifiers

- ba*b
- ba{3,5}b
- ba{1}b
-

22

Buffer Overflow

- What is it?

23

Security

- Should use pattern matches as a security check on input
- Example:

```
unless ( $year =~ /^\d\d$/) {
   die (“problem with year input!”);
}
```

24

hashes

- A hash function is a function that converts an input from a (typically) large domain into an output in a (typically) smaller range
- Example:
  - Map each name in the class to a somewhat unique number
- Collision = when different keys map to the same output.

25

Use of hashes

- Hash tables
  - Data structure
  - Unordered list, fast lookup
- Cryptography
- Data processing

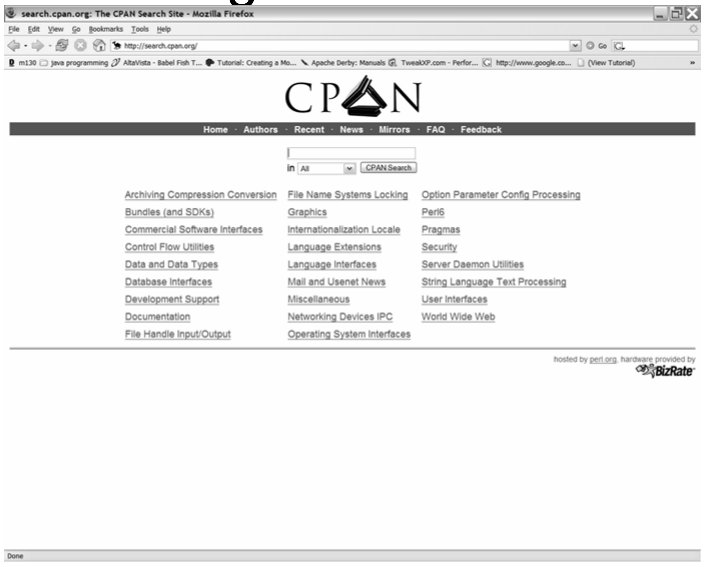
26

MD5 Sum

- MD5 – uses a 128 bit hash value
- Designed in 1991
- Known problems with collision attacks
- http://www.ietf.org/rfc/rfc1321.txt
- http://en.wikipedia.org/wiki/MD5

27

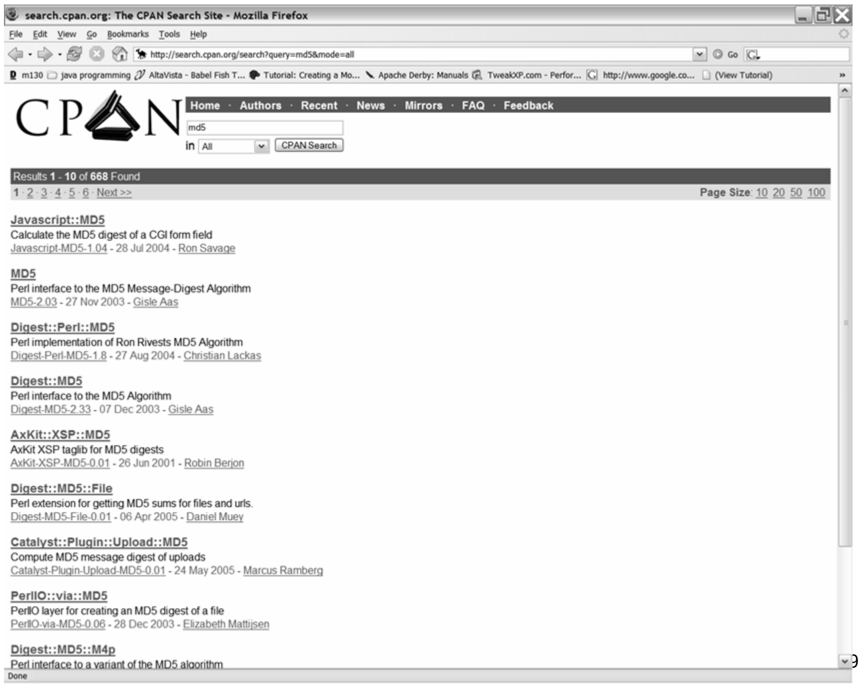
Using Perl Libraries


search.cpan.org: The CPAN Search Site - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://search.cpan.org/
m130 java programming AltaVista - Babel Fish T... Tutorial: Creating a Mo... Apache Derby: Manuals TweakXP.com - Perfor... http://www.google.co... (View Tutorial)
CPAN
Home · Authors · Recent · News · Mirrors · FAQ · Feedback
in All
CPAN Search
Done

Archiving Compression Conversion	File Name Systems Locking	Option Parameter Config Processing
Bundles (and SDKs)	Graphics	Perl6
Commercial Software Interfaces	Internationalization Locale	Pragmas
Control Flow Utilities	Language Extensions	Security
Data and Data Types	Language Interfaces	Server Daemon Utilities
Database Interfaces	Mail and Usenet News	String Language Text Processing
Development Support	Miscellaneous	User Interfaces
Documentation	Networking Devices IPC	World Wide Web
File Handle Input/Output	Operating System Interfaces	

hosted by perl.org, hardware provided by BizRate

28



search.cpan.org: The CPAN Search Site - Mozilla Firefox

http://search.cpan.org/search?query=md5&mode=all

md5

Results **1** - **10** of **668** Found

1 · 2 · 3 · 4 · 5 · 6 · Next >>   Page Size: 10 20 50 100

Javascript::MD5
Calculate the MD5 digest of a CGI form field
Javascript-MD5-1.04 - 28 Jul 2004 - Ron Savage

MD5
Perl interface to the MD5 Message-Digest Algorithm
MD5-2.03 - 27 Nov 2003 - Gisle Aas

Digest::Perl::MD5
Perl implementation of Ron Rivests MD5 Algorithm
Digest-Perl-MD5-1.8 - 27 Aug 2004 - Christian Lackas

Digest::MD5
Perl interface to the MD5 Algorithm
Digest-MD5-2.33 - 07 Dec 2003 - Gisle Aas

AxKit::XSP::MD5
AxKit XSP taglib for MD5 digests
AxKit-XSP-MD5-0.01 - 26 Jun 2001 - Robin Berjon

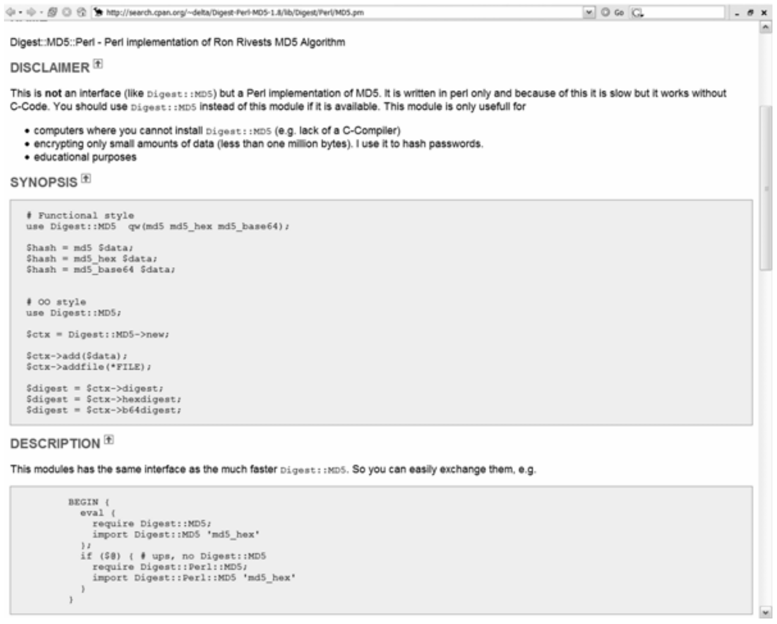
Digest::MD5::File
Perl extension for getting MD5 sums for files and urls.
Digest-MD5-File-0.01 - 06 Apr 2005 - Daniel Muey

Catalyst::Plugin::Upload::MD5
Compute MD5 message digest of uploads
Catalyst-Plugin-Upload-MD5-0.01 - 24 May 2005 - Marcus Ramberg

PerlIO::via::MD5
PerlIO layer for creating an MD5 digest of a file
PerlIO-via-MD5-0.06 - 28 Dec 2003 - Elizabeth Mattijsen

Digest::MD5::M4p
Perl interface to a variant of the MD5 algorithm

9



http://search.cpan.org/~delta/Digest-Perl-MD5-1.8/lib/Digest/Perl/MD5.pm

Digest::MD5::Perl - Perl implementation of Ron Rivests MD5 Algorithm

DISCLAIMER

This is **not** an interface (like `Digest::MD5`) but a Perl implementation of MD5. It is written in perl only and because of this it is slow but it works without C-Code. You should use `Digest::MD5` instead of this module if it is available. This module is only usefull for

- computers where you cannot install `Digest::MD5` (e.g. lack of a C-Compiler)
- encrypting only small amounts of data (less than one million bytes). I use it to hash passwords.
- educational purposes

SYNOPSIS

```
# Functional style
use Digest::MD5  qw(md5 md5_hex md5_base64);

$hash = md5 $data;
$hash = md5_hex $data;
$hash = md5_base64 $data;


# OO style
use Digest::MD5;

$ctx = Digest::MD5->new;

$ctx->add($data);
$ctx->addfile(*FILE);

$digest = $ctx->digest;
$digest = $ctx->hexdigest;
$digest = $ctx->b64digest;
```

DESCRIPTION

This modules has the same interface as the much faster `Digest::MD5`. So you can easily exchange them, e.g.

```
        BEGIN {
          eval {
            require Digest::MD5;
            import Digest::MD5 'md5_hex'
          };
          if ($@) { # ups, no Digest::MD5
            require Digest::Perl::MD5;
            import Digest::Perl::MD5 'md5_hex'
          }
        }
```

30

Digests

- The 128-bit (16-byte) MD5 hashes (also termed message digests) are typically represented as 32-digit hexadecimal numbers.
- Even small change can result in a totally different hash digest
- MD5("The quick brown fox jumps over the lazy dog") =
  - 9e107d9d372bb6826bd81d3542a419d6
- MD5("The quick brown fox jumps over the lazy cog") =
  - 1055d3e698d289f2af8663725127bd4b
- MD5(“”)
  - d41d8cd98f00b204e9800998ecf8427e

31

MD5 Attacks

- Recent work has found flaws with the MD5 sum.
- Will not consider this in our class.

32

scope

- Default scope is main
- $name can also be referred to as

$main::name

- package NAMESPACE
  - Within any block of code, can declare that the rest of the code will belong to a specific namespace

33

Scope II

- my
  declares the variable and value local to the current scope
- our
  confines the name to local scope
- local
  confines the value to local scope
- More than one variable in parenthesis!!

34

What exactly is a module

- Collection of useful subroutines or objects for a specific task

35

Creating a simple library

- Will do it in next weeks lab

36

Graphics

```
#!c:\perl\bin
use Tk;

my $mwin = MainWindow->new;

$mwin->Button(-text => "Hello World!", -
   command => sub{exit})->pack;
MainLoop;
```

37

Graphics

- Will not cover in depth
- Good to know about
- Might need to one day debug someone else’s code (GASP!)

38

For next time

- Reading
- Make sure you have cs account for next week lab
- Start sketching the homework

39