

CS3157: Advanced Programming

Lecture #2

May 24

Shlomo Hershkop
shlomo@cs.columbia.edu

Introduction

- Let me introduce myself
 - My background
 - Where this course fits in
- Go over some class technical details
 - Any questions from last time ?
 - Will be using feedback system
 - For those interested
 - Programming Perl: QA76.73.P22W 2000
 - C how to program: QA76.73.C15.D44 2004

Labs

- Most of your grade in the class
- Easy way to gauge the progress of your learning
- Some labs will build on each other others are geared towards specific learning skills
- Please please please (3X) ask for help if you get stuck (after trying)
 - Goal is to learn not tread in place ☺
 - I read email, am on AIM sometimes
 - Can log into courseworks and leave message on web board

3

Lab Work

- Generally required to get CS account (\$50)
 - I could not arrange free accounts
- If you can bring a laptop into class will not need to lab account
 - But will need to setup environment to work labs on
- We will hold lab hours, last half hour of week, and another slot, we can take a vote on this

4

Outline for today

- More Regular Expressions
- File handling II
- Complex perl examples
- Feedback
- Homework
- More file handling and reg exp
- CGI
- HTML
- CGI & Perl

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

5

General Information

- Slide posting
 - Will post slides within 24 hrs after class
 - Outside code will be also posted (links)
 - Reason for not posting prior to class
 - Please, take notes, and take a look at class slides, make sure you understand what you are looking at
- General Pace
 - Will try to make it easier to take notes
 - Will divide information so easier to digest
 - Will be very technical at certain points...you will thank me later on when trying to solve labs
 - Will attempt more elaborate examples
- Your responsibility
 - Understand the material
 - Stop me to clarify (won't happen on its own)
 - Provide feedback

6

Announcement

- Office hours:
 - M/W 4 - 5:20pm
 - Anytime by appointment
 - On AIM: prof herskop
- Tae
T/TH 4-5pm

7

Announcement II

- Monday memorial day
 - Still need to cover material
 - Options:
 - June 9 Friday ... earlier in the day
 - Online, self study with AIM/Phone/Etc
 - Another day
 - Difficult since some of you are taking tue/thu classes
 - Just meet Monday
 - Will depend on everyone's schedule etc

8

Question

- Did anyone try to write some perl code ?
 - What platform
 - Anything unusual ?
- Anyone attempt the email processor example ?

9

From last time

- What is the default variable ?
- For your own convenience, you should use the “use strict” pragma
- You are in main by default
- Why knowing the scope is important ?

10

From last time II

- When would it be great to use perl?
- Any questions from last class ?

11

subs

- When you pass in values to subroutines:
 - Language independent
 - Need to understand mechanisms for passing in variables
 - For example, programming a cell phone, risc processor, ARM/Thumb environment, what happens at sub call ?

12

2 choices

- Protect variables by copying values
 - Overhead for
 - Memory
 - CPU
- Speed things up by just passing in memory locations
 - Speedup
 - Memory
 - CPU

13

So how does perl do it ?

- First we need to understand global scope!
- Did example last class where passed in scalar and did some changes
 - Anyone remember what happened ?
- We will learn nitty details in c, but passing in the memory location looks like this
 - See board example now

14

Word on quotes

- Perl has 3 different quote operators
- Can either use the quotes or the function name
- Single quotes
 - ‘ ’
 - q{ }
 - Literal meaning, no interop
- Double quotes
 - qq{ }
- Back quotes
 - qx{ }
- Word lists
 - qw { }

15

Some perl

- qw / /
- Will take all tokens between slashes and make "" quotes around things
- Very useful shortcut when lazy i.e. when you have better things to do 😊
- How do you look up new perl commands ?

16

Some more on scope

- Anyone know the difference between a locally created variable and global ??
 - Can you give me a practical example ?

17

Regular Expression

- Lets Review
- More examples

- So what exactly is a regular expression?

18

Regular Expression in perl

- Trying to represent patterns to perl
- m/ /
- What does match return ?
- Very powerful since we can define our program behavior based on general pattern definitions
- Many many shortcuts available

19

Simplest

- Simplest regular expression is a literal string match

```
if ($location =~ m/white house/i ) {  
    snappicture();  
}else {  
    goSleep();  
}
```

20

Regular Expressions

- complex regular expressions use *metacharacters* to describe various options in building a pattern.
 - \ul> - Escape character
- .ul>- Match any single character
- Full list, can you tell me what each thing does ?
\\ | () [] { } ^ \$ * + ? .

21

substitutions

- s/pattern/pattern/
- Instead of return t/f we return number of matches
- And **will** change the applied target
- Here is an example:

22

```
$text = "Hi my email is shlomo\@cs.columbia.edu";  
  
$text =~ s/\W(sh.*)\@(cs\.columbia\.edu)/ cool\@$2/i;  
  
print $text . "\n";
```

23

transliteration

- tr/search_list/replacement_list/
- This concentrates on character replacement....think simple cyphers
- -c all characters not in the search list
- -d anything without replacement ...delete
- -s squash duplicates

24

Argument passing

- Anyone know what is argument passing ?
- @ARGV
- How to get the length ?
- How to get individual items ?
- Note: older style was \$#ARGV

25

STDIN STDOUT

- <STDIN>
- <STDOUT>

26

Unix OS

- Will cover some general unix system ideas
- Please email/ask if you are curious
- Make sure you understand what we are covering

27

File system

- Everything is a file
- Root (top) level file is slash /
- Everything is under root
- General configuraton is usually in /etc/
- The login information is in /etc/passwd

- Actual passwords:
 - Used to be plaintext in /etc/passwd
 - Later scrambled
 - Later put in private shadow file

28

permissions

- Different levels of permissions allow a wide variety of control over underlying operating system
- User/groups/others
- Read/write/execute
- `ls -la [filename]`
- `chmod [ugo][+-][rwx]`

29

Code examples

- We want to process the `/etc/passwd` file
 - Any idea what would be a good application here ?
- Looks like:

```
pcap:x:77:77:ARPCATCH User:/var/arpwatch:/sbin/nologin
ident:x:98:98:pident user::/bin/false
nobody:x:99:99:Nobody::/sbin/nologin
xfst:x:405:405:X Font Server:/etc/X11/fs:/bin/false
mysql:x:6730:1101:mysql server:/var/lib/mysql:/bin/bash
```

30

```

sub read_passwd {
my %users;
my @fields = qw/name pword uid gid fullname
  home shell/;

while(<STDIN>) {
  chomp;
  my %rec;

  @rec ={@fields} = split(/:/);

  $users{$rec{name}} = \%rec;
}
return \%users;
}

```

31

```

my $users = read_passwd();

my @names;

foreach (keys %{$users}) {
  next unless $users->{$_}{fullname};

  my ($fname, $lname) = split (/\/\s+/,
    $users->{$_}{fullname}, 2);

  push @names, "$fname $lname";
}
print map { "$_\n" } sort @names;

```

32

Helpful stuff

- `$| = 1`
will turn off output buffering great when working with cgi (later today)
- In perl, can call external commands i.e. we can execute command line arguments
 1. Backticks (```)
 2. System
 3. exec

33

TOOLS: VNC

- www.realvnc.com
- One way to easily work remotely on clic
- Start server on a clic machine (city.clic.cs.columbia.edu):
 - vncserver
 - Run client on your side
 - demo

34

WWW

- Driven by http
- Technical overview
 - Servers serve http request
 - Clients browsers issue requests

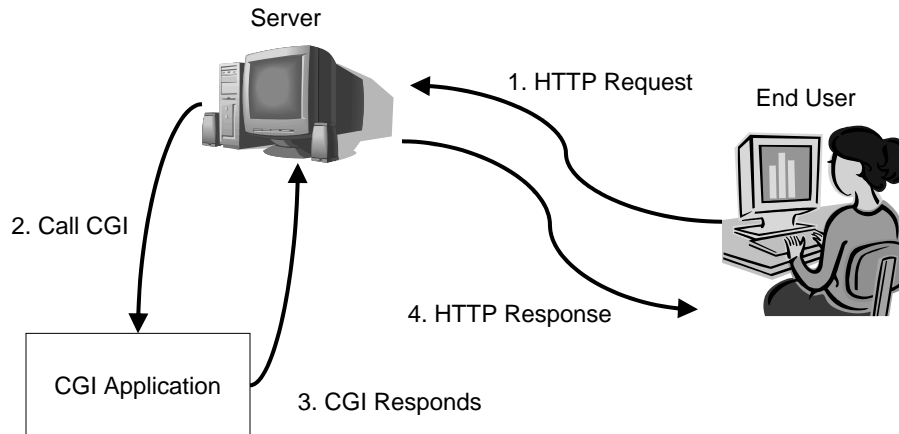
35

Boring vs. Exciting

- Typical
 - Request is served from a file formatted in html
 - Static file of what we would like to render on a web client.
 - Example:
 - Class syllabus
- What is we could tailor each users web experience to what they want.
 - Design of protocol to handle this

36

How does CGI work:



37

Perl + cgi

- Remember:
 - Perl is only a tool here
 - Don't memorize, understand
 - Why
 - What
 - How
 - Don't be afraid to experiment
- STDIN
 - Contents passed to perl script
- STDOUT
 - Will need HTTP headers before printing
- STDERR
 - Depends on server, sometimes just error logs, sometimes error reports on client

38

%ENV

- This is your best friend in PERL CGI
- Way of getting information *from* the client
- Create content is way to pass back information *to* the client

39

Remember

- Need to set permissions:
 - `chmod 0755 ???.cgi`
 - `-rwxr-xr-x`
- Need to place script in correct place
 - Usually cgi-bin/ directory
- Naming
 - Usually need to end in .cgi

40

Sample test4.cgi

```
#!/usr/local/bin/perl

use strict;

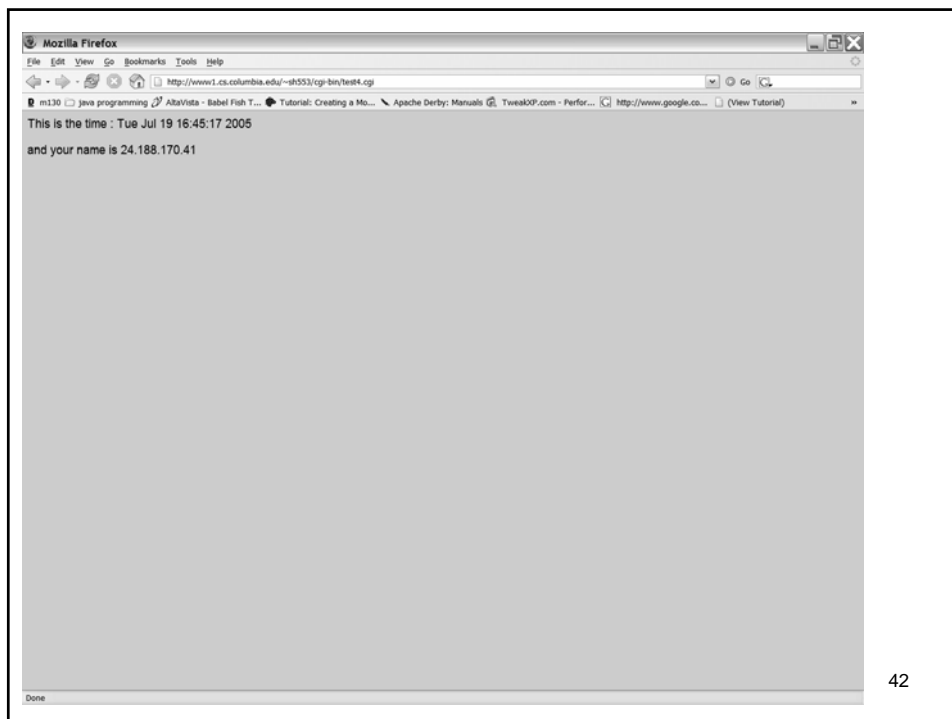
my $time = localtime;
my $remote_id = $ENV{REMOTE_HOST} $ENV{REMOTE_ADDR};

print "Content-type: text/html\n\n";

print <<END_OF_PRINTING;
This is the time : $time
<P>
and your id is $remote_id

END_OF_PRINTING
```

41



42

Some CGI Environmental Variables

- CONTENT_LENGTH
 - Length of data passed to cgi
- CONTENT_TYPE
- QUERY_STRING
- REMOTE_ADDR
 - Ip address of client
- REQUEST_METHOD
- SCRIPT_NAME
- SERVER_PORT
- SERVER_NAME
- SERVER_SOFTWARE
- HTTP_FROM
- HTTP_USER_AGENT
- HTTP_REFERER
- HTTP_ACCEPT

43

Problem

- How can we print out all the environment variables ?

44

Example

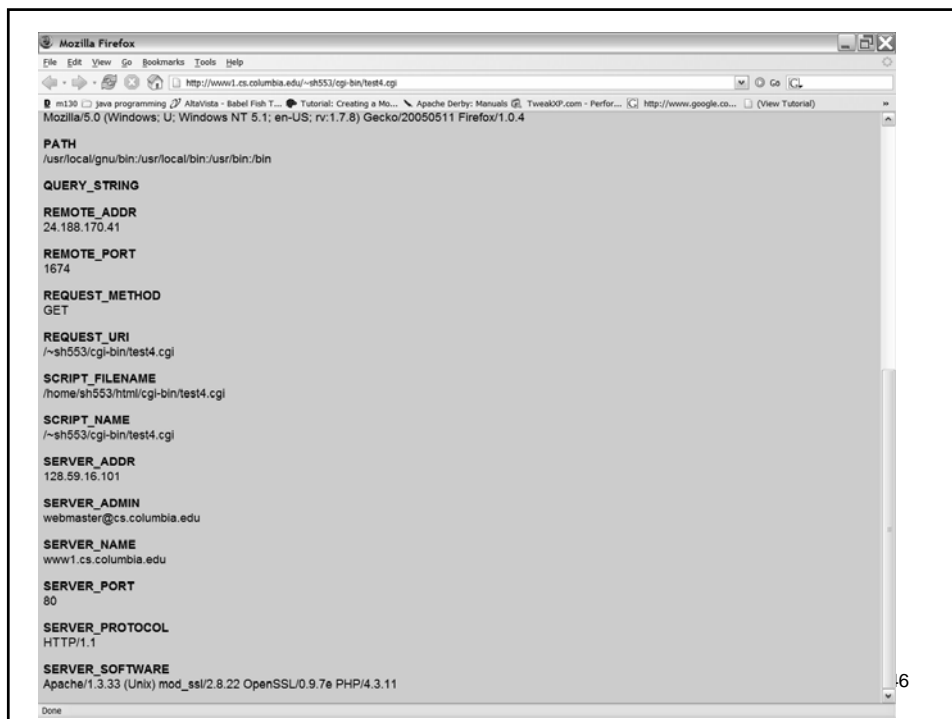
```
#!/usr/local/bin/perl

use strict;

my $vars
print "Content-type: text/html\n\n";

foreach $vars (sort keys %ENV){
    print "<P><B>$vars</B><BR>";
    print $ENV{$vars};
}
```

45



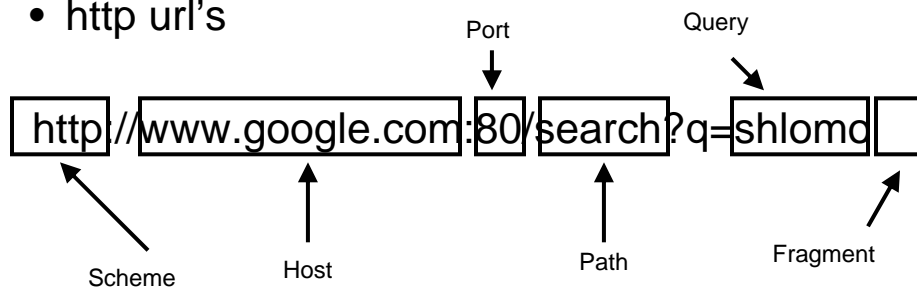
HTML

- Hyper Text Markup Language
- Standard by w3:
<http://www.w3.org/MarkUp/>
- Way of standardizing format of documents so that users can share information between different systems seamlessly
- Evolving to XHTML format

47

HTML

- Hypertext Transfer Protocol
- Language used between web servers and web clients
- http url's



48

Google.com

- <http://www.google.com/search?q=shlomo>

49

Very basics

- Html consists of matching tags
- `<something>` = opening tag
- `</something>` = close tags
- HTML DOC:
 - `<html> <body> </body> </html>`

50

Web pages

- `<title> </title>` (before the body section)
- `<H1> </H1>` (header titles h1, h2, h3)
- `<P>` paragraphs
- `
` line breaks
- ` ... ` bold
- `<i> ... </i>` italicize
- `<u> ... </u>` underline

51

More basics

- ``
- ` something `
- ``
 - Can be referred to by `page.html#Anchor1`
- `<hr>` line
- `<hr width=50%>` half line

52

Lists

- Unordered list

```
<ul> <li> </li> .....</ul>
```

- Ordered list

```
<ol> <li> </li> ..... </ol>
```

- Nested lists

– Lists themselves can be nested within another

53

Tables

- ```
<table>
<tr>
<td>Hello</td>
<td>World </td>
</tr>
</table>
```

|       |       |
|-------|-------|
| Hello | World |
|-------|-------|

54

## comments

<!--

anything you do

-->

55

## More html

- Can get wysiwyg editors
- Word will allow you to save as html
- Can take a look at webpages source code

56

## Browser Issues

- Although HTML should be universal, there are occasional differences between how Microsoft IE renders a webpage and Mozilla firefox

57

## Perl Debugging

- Command line debugger can be started with the -d command argument
- ```
perl -d something.pl
```
- h = help
 - x = examine something
 - Any perl command is read in, and saved
 - s = single step evaluation
 - n = jump over subroutine
 - v [num] = window of commands we are in
 - l x y = list lines x to y

58

Perl debugger

- b num = breakpoint at line num
- c = run until next breakpoint
- d num = delete breakpoint at line num
- X examine all variables

59

Task

1. Create a webpage counter (saying you are visitor x to this page)
2. Now create a graphical counter

60

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>

61

Bottom line

- Still in very wide use
- Allows authentication of files given a file and signature
- Visually authentication against tampering
- What obvious weakness??

62

Md5 of a file

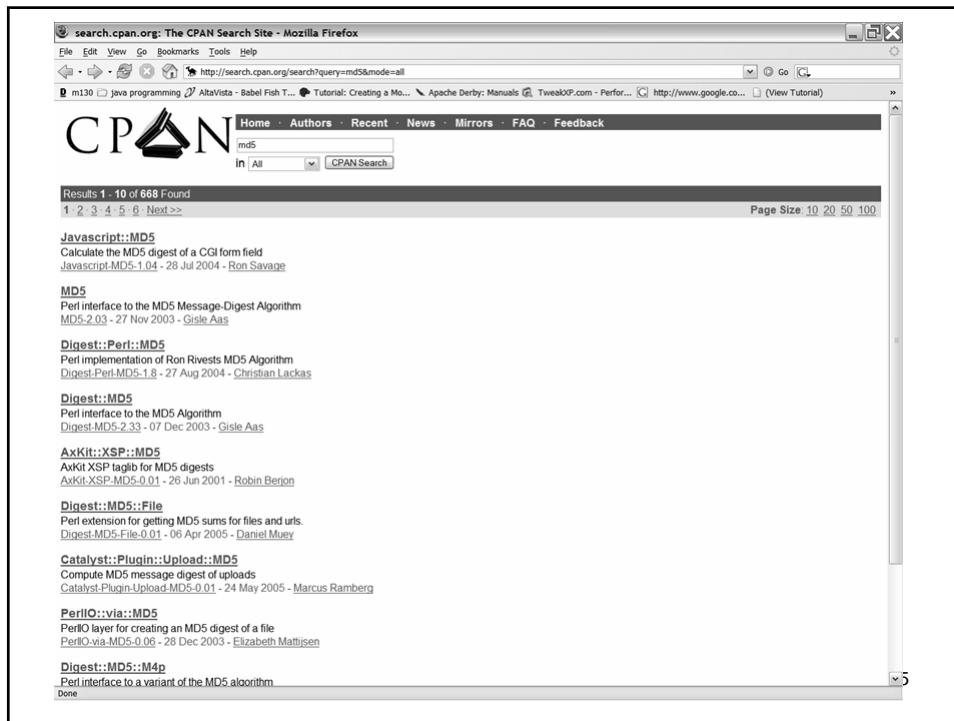
- Can execute md5sum within perl
- Can use perl defined methods
 - Write yourself
 - Find someone else's ☺

63

Using Perl Libraries

The screenshot shows a Mozilla Firefox browser window displaying the CPAN Search Site. The browser's address bar shows the URL <http://search.cpan.org/>. The page features the CPAN logo at the top center, with a navigation menu below it containing links for Home, Authors, Recent, News, Mirrors, FAQ, and Feedback. A search bar is located below the navigation menu, with the text "In All" and a dropdown menu set to "CPAN Search". The main content area is a grid of links organized into three columns. The first column includes links for Archiving, Compression, Conversion, Bundles (and SDKs), Commercial Software Interfaces, Control Flow Utilities, Data and Data Types, Database Interfaces, Development Support, Documentation, and File Handle Input/Output. The second column includes File Name Systems, Locking, Graphics, Internationalization, Locale, Language Extensions, Language Interfaces, Mail and Usenet News, Miscellaneous, Networking Devices, IPC, and Operating System Interfaces. The third column includes Option Parameter Config, Processing, Perl6, Pragmas, Security, Server Daemon Utilities, String Language Text Processing, User Interfaces, and World Wide Web. At the bottom right of the page, there is a small logo for BizRate and the text "hosted by perl.org hardware provided by".

64



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

67

Digests II

- MD5("The quick brown fox jumps over the lazy dog") =
 - 9e107d9d372bb6826bd81d3542a419d6
- MD5("The quick brown fox jumps over the lazy cog") =
 - 1055d3e698d289f2af8663725127bd4b
- MD5("")
 - d41d8cd98f00b204e9800998ecf8427e

68

Recursive directory crawling

- Sample1.pl

69

File::Find

```
use File::Find;

$dir = "c:/example";

find(\&exam1,$dir);

sub exam1{
    print "File: $_ and path is
$File::Find::name\n";
}
```

70

GUI

- There are easy ways to make graphics in perl
- Will not cover in this course
 - But will have enough knowledge to pick this up on your own if you choose
 - Better way: will see later today

71

Graphics

```
#!c:\perl\bin  
use Tk;  
  
my $mwin = MainWindow->new;  
  
$mwin->Button(-text => "Hello World!", -  
             command => sub{exit})->pack;  
MainLoop;
```

72

Graphics

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

73

Computer Security

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

74

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

75

Homework 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
- Want to search filenames
- Want to organize snapshots of system

76

Useful programming 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 (today if time permitting).

77

Doing the work

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

78

Simple example

- <http://www.cs.columbia.edu/~name/a.pl>
- User in browser invokes perl script
- Web server calls script
- Perl script runs and print out a html code
- Web browser renders the webpage

79

Next step

- Not just execute the script want to get some starting information from the user

80