Announcements

- Project #5 extended until Dec. 10
- Reading: 7.6
- HW#2 Due today
- TA Extra office hours
  - In 3122
  - Tu: 2-3
  - W: ??
WWW (cont.)

- **HyperText Markup Language**
  - based on SGML
    - font changes, text placement
    - includes support for images
  - supports references to other document (links)
  - supports alternatives to display if browsers can’t support a format

- **HyperText Transport Protocol**
  - used to move HTML from server to client
  - Basic protocol
    - GET: get a page
    - PUT: store a page
    - POST: append to a page
Interactive Web Pages

- **Forms**
  - HTML can describe fields which permit users to enter data
    - textboxes, checkboxes, lists, etc.
  - contain an action
    - a URL to POST the completed form

- **Common Gateway Interface (CGI)**
  - Servers can be told that some pages are really programs
    - could be executable binaries, perl programs, etc.
  - An attempt to POST to a CGI script runs it
    - the form data is taken as input
    - CGI script returns an HTML page as output
      - output can be a function of the input
  - common examples:
    - perl scripts
    - interfaces to database systems
Applications of Runtime Code Patching

- **Performance measurement**
  - Recording application behavior

- **Correctness debugging**
  - Fast conditional breakpoints
  - Data breakpoints

- **Execution driven simulation**
  - Architecture studies

- **Testing**
  - Code coverage testing
  - Forcing hard to execute paths to be taken
Structure of the Dyninst Library

Mutator App

Dyninst Code

Machine Dependent Code

Ptrace or procfs

Application Code

Snippets

Run-time Library
API Library

- **Provides**
  - Functions for control of mutatee
  - Runtime code generation
  - Information about mutatee

- **A set of C++ classes**
  - Machine independent representation of a program
  - Processes and threads
  - Representation of new code to patch into program
Representing Code Snippets

- **Platform Independent Representation**
  - Same code can be inserted into apps on any system

- **Simple Abstract Syntax Tree**
  - Can refer to application state (variables & params)
  - Includes simple looping construct
  - Permits calls to application subroutines

- **Type Checking**
  - Ensures that snippets are type compatible
  - Based on structural equivalence
    - allows flexibility when adding new code
Binary Object Version Problem

- Installing a new version of DrawIt
  - SpeakOut 2.0 runs fine
  - AddUp stops workings
Ways to Maintain Library Versions

- **Full replication**
  - Forces each app. to have a copy of its objects
  - Ensures correct version, but wastes space
- **Name identifies an object**
  - Doesn’t support multiple versions
  - Installing a new version can break applications
- **<name, version> tuple identifies an object**
  - Supports multiple incompatible objects
  - Requires explicit management of namespace
    - who gets to create an object name
    - what is the number for the next version
Content-Derived Names (CDN)

- **Use object content to create a name**
  - Each object will have a unique name
  - No need for explicit naming of versions
  - Multiple versions may co-exist on a single system

- **Key idea: use a secure hash function**
  - Produces the name (hash value) based on content
  - Use MD5 - produces 128 bit output

- **What about collisions?**
  - Causes two objects to have the same name
  - Probability can be made arbitrarily low
    - use hash function with more bits in output
Other Possible Uses of CDNs

- **Mobile computers**
  - Can use any close fileserver
  - Request a Content-Derived Name
  - Verify object integrity by computing CDN

- **Network-based software distribution**
  - Treats local disk as a cache
  - Demand loads objects based on CDN
  - Allows garbage collection of old versions of software
  - Permits multi-party purchases