IS 2550: Client Server Systems
Course Introduction

Michael B. Spring
Department of Information Science and Telecommunications
University of Pittsburgh
spring@imap.pitt.edu
http://www.sis.pitt.edu/~spring

9/5/2006 IS 2550: Client Server 2

Fall 2006 (07-1)

- Client-Server and Workstation Systems
  - DIST 2550
  - CRN: 16338
  - Tuesday 6:00-9:00 Room 411
- Related to next week – code, code, code
  - Work on Unix
  - Work on workshop
  - Work on man pages

9/5/2006 IS 2550: Client Server 3

Contact Information

- Instructor: Michael B. Spring
- Office Location: 701A SIS Building
- Website: http://www.sis.pitt.edu/~spring
- Personal Email: spring@imap.pitt.edu
- Class Email: mbsclass@mail.sis.pitt.edu
- Office Hours: Monday-Friday 8:00-6:00
- Phone: 412-624-9429
Instructor-Research Interests

- Distributed systems
  - Client server system design
  - E-services
- Document systems
  - Collaborative authoring
  - Navigation
- Interactive systems
  - Office Automation
  - Interface design
  - Visualization
  - Agents
- Standards
  - Document and data
  - HCI

Course Expectations

- Prerequisites
  - Facility with mail, ftp, and utilities
  - Knowledge of operating systems, particularly unix
  - Ability to use development tools (IDEs)
  - Ability to write solid code in C
- Preparation
  - Reading and thinking
  - Thinking and “doing” – beyond what I have “done”
- Engagement
  - Experimentation
  - Planning

Course Resources

- The textbooks – they are to be read and digested – beware the ease of reading Comer
- The lectures – they will go beyond and around the readings and assignments
- The web and manual – there is more around than you will ever digest
- Your own experimentation and discussion with other students
- Files provided by the instructor – CASCADE and sunfire
CASCADE

• CASCADE is a client server application written as part of one of the instructors research projects.
• It is used in class as a means of sharing information with students
• CASCADE stands for Computer Augmented Support for Collaborative Authoring and Document Processing

To run CASCADE:
• (On Solaris Machines)
  • type cascade
  • Let the system store local information is being set up.
  • You will need to set up the server information
• (On lab PCs)
  • select Cascade from the start menu.
  • You will need to set up the server information each time you run the system
• (On your own PC)
  • Obtain a copy of the CASCADE client form the CASCADE web site -- www.sis.pitt.edu/~cascade -- and follow the setup instruction.
  • set up the server information

CASCADE Server Information:

• There is a drop down combo box that allows you to set up a name, host, and port name.
  • name can be any string you want,
  • host for the class accounts is "augment.sis.pitt.edu"
  • port is "7000"
• In class, you will be provided with a username and password that will allow you to login to the CASCADE server.
• If you want to try to access the server before getting an account name, the account “guest” with the password “guest” provides minimal read only access.
To learn more about CASCADE

- Use the extensive online help system.
- Go to the web site and:
  - Read the help documents
  - Take a virtual tour of CASCADE
  - Watch the ten videos that can be run on a PC.

Submission of Assignments

- Note that the instructor reserves the right to shift from project based grading to examination based grading
- Care should be taken by students to make sure assignments submitted contain all the necessary information needed for assessment
- Plagiarism will not be tolerated in the course. If the instructor suspects students are submitting non-original material, a grade of 0 will be submitted for the assignment and a failing grade for the course will be assigned

Projects

- Projects should be:
  - Thoroughly tested and documented
  - Code from any source other than your head should be noted including code adapted from some source. Failure to note such use is cause for a grade of 0 on the assignment and an F in the course.
  - Source code and executables should be included.
    - As mail attachments. If there are many parts, use a zip jar, or tar file.
    - Include supporting DBMS and lib and jar files.
    - Include a readme file that explains constraints or setup
  - In the mail note and the main project file, include:
    - The names Email addresses and SSNs of all participants
    - The course, term, CRN, and assignment
Assignments

• Informal
  • Develop any C code making system calls and using workshop

• Formal
  • Fix a client (working to good code)
  • Modify a server (iterative to concurrent)
  • Develop a pair (protocol, GUI, 3-tier)

Final Project

• Syllabus
  • Collaboration tool
    • Calendar, chat, idea manager
  • Collaborative editor
    • XML messaging and management – granular locks and shared spaces.
  • E-services
• Added
  • Multi-user game/simulation

Seven Imperatives for 2550

• Learn to read man pages
• Get a gut level feel for distributed processing
• Get a gut level feel for data storage and manipulation
• Learn how processes relate to the operating system and how errors of various types occur
• Learn how to use Unix a and programming tools effectively
• Understand what a protocol is – learn how to read standards
Man Pages

- Unix has literally thousands of man pages
- Related to function and system calls, they have a particular form
  - Prototype of the function
  - Synopsis of functionality
  - Description of arguments
  - Return values
  - Error conditions
  - Other pages of interest
- Return values and error conditions are very important to us.

Distributed Processing

- When using a PC, almost everything happens on the PC. Even when using the web, we are encouraged not to think about the fact that the system is distributed
- On Unix, and in Client-Server, it is important to understand how things are distributed. Consider three examples:
  - NFS
  - The X Window System

Network File Service (NFS)

- NFS is Sun’s version of a distributed file service.
- A central robust file server maintains file data
- Clients on authorized workstations access this file system caching it locally as needed.
- From the user perspective, the entire file system appears local on each machine
**X Window System**

- The X Window System reverses the client server paradigm.
- Processors exist around the network and do the basic processing chore.
- The input to the processes and the results are sent to one or more windows on a “window server”
- The user, at the window server, must have the right to start processes on the processors.
- From the user perspective, all of the processes appear to run on the local machine.

**Data Storage**

- How data is stored on a machine is arbitrary
- For example, consider an integer:
  - 2, 4, or 8 bytes – your choice
  - First bit most significant or least significant
  - The binary integer 0 could be zero or a negative number that is half the largest number storage will allow
- A float can be any number of exponent and mantissa orders and sizes
- A string can be defined several ways
  - Any set of characters terminated a null
  - An integer followed by a string of that length

**A Data Storage Question**

- Consider if you will the following:
  ```c
  char buffer[80];
  int a;
  char b, c[20];
  strcpy(c, "hello");
  b='r';
  a=10;
  memcpy(buf, &b);
  memcpy(buf+1, &a);
  memcpy(buf+5, c);
  printf("%s", buf); //what prints and why
  ```
**Programs, Processes & Errors**

- A program is first compiled – yielding compile errors, and then linked – yielding link errors.
- When it is run, there may be run time errors. It is these that debugging is most important for.
- When a program is linked your code, library functions you call, and other functions are woven together.
- Part of the magic code is code to handle signals from the operating system and other processes.

---

**Unix and Programming Tools**

- The Unix operating system provides a rich environment for developers. There is a 30 year accumulation of tools and libraries – literally 1000’s of modular tools and APIs.
- There is an increasingly sophisticated WYSIWYG interface in the Common Desktop Environment and dozens of “drag and drop” tools:
  - netscape as a browser/editor/mailtool
  - nedit and xemacs for editing
  - xv for image processing
  - workshop (c/c++) and forte4j (java) Integrated Development Environments (IDEs)

---

**Protocols**

- In general, a protocol is a set of rules, or a procedure. In diplomacy, it defines who is allowed to say what at what point in an interaction.
- A protocol, in the client server world, defines the rules two independent programs use to communicate.
- Protocols are defined in standards documents.
  - RFC 822 defines the structure of internet mail messages
  - RFC 823 defines the rules for sending and delivering messages
  - ISO 10646 defines the international character set (UNICODE)
  - IEEE 802.3 defines messages on the ethernet
  - RFC 2616 defines the http protocol – a very simple one.
    - The client makes a request
    - The server responds to the request