Projects

- 4 Projects
- Work individually for the first project, groups of 3 after that
- You need to have basic Unix and C skills
  - I’ll go over a little today and point you to more resources
- Don’t wait until the last minute!
  - Projects require a substantial amount work and you’ll be much happier if you start early
- Lots of helpful info on the course web page
Project 1

- Introduces you the Unix and C skills you’ll need for the rest of the class
- Teaches you how to build and run Linux in VMWare
- Two parts
  - Write a simple shell
  - Write a syscall
- First steps:
  1. Read the project
  3. Start the project
Computing Resources

Instructional Servers
attu (used to be fiji, sumatra, ceylon, tahiti)

Our Dedicated Servers
coredump and spinlock

CSE 006 Lab

DO NOT USE!!!!

Use these instead

- Develop your code on coredump or spinlock.
- Test your code in VMWare.
- WARNING: Do not run your code on the instructional servers (attu). You can easily fork bomb a server, which can bring down the server. Use only spinlock, coredump, and VMWare.
VMWare

- VMWare is a software simulation of an x86 computer.
  - The OS thinks it is running on a real machine
- It allows you to run an OS in a sandbox.
  - Easy to reset to known good state
**VMWare Basics**

- **Power functions (on, off, reset)**

- **Machine name**

- **VMWare config**

**All disks are non-persistent.** This means that using the Power Off function will lose any changes you made to the file system inside VMWare.

- A soft reboot will preserve data. Type “shutdown -r now” or “reboot” from the shell instead of hitting the reset button.

**Network Adapter is Host Only.** This means that only the host computer can talk to the VMWare network card. You must transfer files (like your kernel) to the host machine first.
VMWare Tips

• There is only one user: “root”
• The password to root on the vmware machines is “rootpassword”
• Use ftp to get files into VMWare
  – Move files onto your local machine
  – Open Internet Explorer to access VMWare through the URL,
    ftp://root:rootpassword@192.168.93.2
  – If you decide to use another ftp method, remember to check binary mode. Otherwise you’ll get garbage instead of your file.
Unix Help

- ACM Unix tutorial
  - Wed Oct 8, 5-7pm, CSE 022
- Unix tutorials online at the ACM website
Intro to C

• What is C?
  – C is a procedural language (not an object oriented language)

• Major Syntactic Differences in C
  – No classes
    • Structs are not the same as classes in C++ or Java
  – Variables must be declared at the top of a block (after an opening brace) before any other kind of statement
  – No new/delete operators. Use malloc and free functions (see man pages – man malloc).
  – void* is used often in C data structures
  – No boolean type
    • 0 is false, anything else is true
Basic C type mechanisms

Structs

- Structs are meant for grouping structured data.
- They are not classes. Structs do not have inheritance and methods like classes do (though you can simulate inheritance and methods fairly well. Look at glib).
- You use structs for 2 reasons.
  - Ordering memory
    - Because structs guarantee a memory layout, they are useful for communicating with hardware
  - Grouping related items
    - This is the more common usage. You can use this to create really dumb “objects”.
Basic C type mechanisms

Typedefs

- Typedefs are a way of creating aliases for a type. So if you type:

  ```c
  typedef unsigned char byte;
  ```

  From this point on, a “byte” is the same as char.

- You use typedefs for 3 reasons.
  - Making a shorthand (often done with structs and function pointers).
  - Adding an extra level of abstractions to the type. (say you’re waffling between using a short int or a long int).
  - Designating a logical difference. A byte is the same as an unsigned char, but when you see “byte” you think of 8 bits where when you see “char” you think of ‘a’ or something similar.