Looking for handouts and lecture information

Handouts are here.

Also, many lecture topics are or will be turned into wiki pages on the wiki crowd-sourced book and videos

Lecture 17. Pipes for IPC. Files and forking IPC

- Pipes for IPC
- File length
- File position and forking
- Closing file handles after forking

Lecture 16. Page Tables. IPC

- Size of single and a 2 level page table
- TLB
- Simple pipe example

Lecture 15. Dining Philosphers - Code analysis

- Dining Philosphers - Code analysis
- Process Diagrams (unsafe regions)

Lecture 14. Deadlock.

- Deadlock analysis
- Resource Allocation Graphs
- Coffman conditions
- Dining Philosphers (intro)

Lecture 13. Reader Writer #2

- Ring buffer (aka circular buffer) implementation
- Synchronization Code analysis
- An improved Reader Writer Solution

Lecture 12. Condition Variables. Reader-Writer

- Condition Variables
- Reader-Writer problem

Lecture 11. Condition Variables

- Condition Variables

- Analysis of solution to the Critical Section Problem
- Implementing semaphore using condition variables

Lecture 10. The Critical Section Problem

- The Critical Section Problem

Lecture 09. Using Mutex and Semaphore to implement a thread-safe data-structure.

- Mutex and Semaphore review
- Implementing a thread-safe stack
- Using counting semaphores to prevent underflow and overflow
- Incorrect implementation of a semaphore

Lecture 08. Critical Section. Mutex and Semaphore.

- What is a Critical Section?
- Locking and unlocking a mutex
- Wait() and Post() of an unamed counting semaphore

In addition to the handout(see above link), see Mutex locks - Part 1 and Semaphores - Part 1

Lecture 07. Using pthreads

- pthread_create,_cancel,_exit,_join
- Passing data to threads. Gotchas (race-condition gotchas)
- Why are some functions not thread-safe

In addition to the handout(see above link), see Pthreads-Part-2

Lecture 06. Malloc,Calloc,Realloc. Intro to VM. Intro to threads

- Malloc,Calloc,Realloc
- Short introduction to Virtual Memory.
- Intro to threads (pthread_create, pthread_join) In addition to the handout, see Pthreads-Part-1 and Heap memory introduction

Lecture 05. Thanks for the memory.

- Intro to how malloc works. Fragmentation. In addition to the handout, see Implementing-a-memory-allocator and Heap memory introduction

Quiz 02 (Mon 9/8)

- Results are posted in your subversion directory and emailed. Median score 16.

Lecture 04 (Fri 9/5). Fork bombs and zombies - When good processes go bad

- Fork-exec-wait. exec. waitpid. Zombies. Fork bombs.

Lecture 03 (Wed 9/3)

- Signals intro (SIGALRM, SIGINT). Environment variables. Program arguments)

Quiz 01 (Fri)

Lecture 02 (Wed 8/27). C Intro Crash Course

Todos

1. Attend section tomorrow
2. Spot the mistakes in your friends' C code. You can bring along your friends printouts to share
3. C Programming (please fix up typos; add more examples etc etc)
4. Tricky Extra Bonus Code-Review Credit: I wrote some C code for your startup. Is there anything underhanded in it? (pdf,txt,docx versions)
5. Bring your ICARD and pencil to Friday's class. Wednesday&s class, your committed code in subversion, C Gotchas wiki page and your section will be my inspiration for the quiz.

Lecture 01 (Mon 8/25). Welcome.

Lecture handouts (and past-semester ppts) are in subversion here.

Post lecture Todos:

```
Login into your favorite UIUC linux machine e.g.
ssh netid@linux.ews.illinois.edu

// (If you do this on your own linux laptop, replace '$USER' with your netid if your user id != netid)
mkdir cs241
cd cs241
svn co https://subversion.ews.illinois.edu/svn/fa14-cs241/$USER
//     (your svn area will exist 12-24 hours after course registration)
cd $USER;
// Create 5 awesome programs that are bad C programs.
//Show that you know some of the C Gotchas by deliberately creating some bad programs
// See my C Programming gotcha's here (I'm sure there are many other gotchas...)-
// System Programming Book
svn add myfile.c
svn commit -m 'Awesome work by me'
Before lecture.... Open https://subversion.ews.illinois.edu/svn/fa14-cs241/$USER
..(replace $USER with your netid) to print your work and bring it to lecture #2.
```

```
Q. Help I'm stuck! These instructions are insufficient!
//Ans. Don't panic. You're not in the course alone. Use Piazza. Use Google.
//You're not a freshwoman/freshman any more. You're from UIUC - you can do this!


A / A+ students ... Got some spare time ?

Play with the linux-in-the-browser project
```