[CS 5600] Computer Systems
Hour 1, Lecture of Oct 20th

1. Project 1 review.

1.1 Alarm clock

1.1.1 How efficient the code dealing with interrupts.

1.1.2 The field we need in the thread structure

Similar in IP v4 header, one of the fields is ‘Link’; and it would be weird if the ‘link field’ is shorter than header.

1.2 Priority scheduling.

The difficult part is implementing priority donation.
Think and model the problem before you write any code;
Multiple threads waiting for one lock => tree structure is an intuitive option.
There are still other possible implementations.

1.3 Advanced Scheduler.

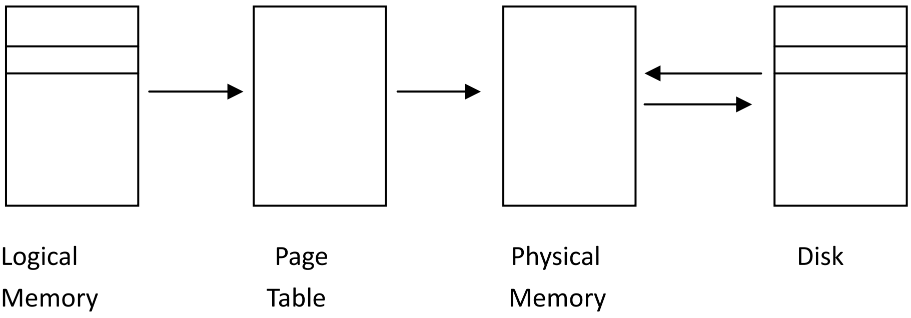
Most team didn’t have a problem.

On Coding Style: Not required now, but will be in the future.
On Questions: Always ask in the blackboard forum before sending emails to the TA.

2. Continuing the topics of Virtual Memory

It appears to have more memory while using virtual memory;
Leverage of virtual memory: file access, where the ‘calls’ actually means system call.
More efficient way to access files, and most modern operating systems support this.

Memory mapping files:



Detailed diagram in the page 349 of textbook, fig 9.23;

3. Mid-Term Information.

25% of the total credit.
The first 2 hours of Oct 27’s lecture;
Open notes, open books (textbook only), NO internet access during exam;
Pintos-related programming questions will be covered in the exam;
Other topics will be covering:

1. Computer Organization basics
   e.g., what an Operating System does, memory protection, resource allocator, hardware support, timer, memory hierarchy-speed of each memory, microkernel and module kernel, the way kernel interacts with upper and lower level, system call, interrupts, etc.
2. Process
   What processes are;
   Process state: running, waiting…
   How items are stored; PCB;
   How context switch are implemented
   Basic Scheduling issues;
   How process are created; fox() definition, what JOIN is;
   How process are waiting for their children;
   IPC;
   Shared memory
3. Thread
   Definition;
   Implementing threads: many to one model; many to many model;
   Multi-threads concepts, advantages and disadvantages;
   Scheduling algorithms, and evaluating scheduling.
4. Synchronization
   What Critical Section is;
   Can critical section be solved in software?
   What hardware use to solve critical section?
   Higher level system primitives;
   Transactions: check-points, etc.
5. Deadlock
   Prevention and protection;
   MMU;
   Run out of physical memory—swap out;
   Contiguous memory allocation;
   Paging, page table and their hardware support.
   What TLB is: Translation Lookaside Buffer;
   Flush the TLB;
   Hierarchical paging;
   Valid and invalid page tables;
   Hashing talbes;
   Options other than paging?
6. Virtual Memory
   How to deal with page faults; pure demand paging and its cost;
   Implementation of Copy-on-Write;
   Page replacement algorithms: FIFO, Least Recent Used(LRU), Second-chance..
   Frames, etc.