Outline for April 6, 2004

1. Principle of Economy of Mechanism
   a. KISS principle
   b. Enables quick, easy verification
   c. Example of complexity: *sendmail*
2. Principle of Complete Mediation
   a. All accesses must be checked
   b. Forces system-wide view of controls
   c. Sources of requests must be identified correatly
   d. Source of problems: caching (because it may not reflect the state of the system correctly); examples are race conditions, DNS poisoning
3. Principle of Open Design
   a. Designs are open so everyone can examine them and know the limits of the security provided
   b. Does ***not*** apply to cryptographic keys
   c. Acceptance of reality: they can get this info anyway
4. Principle of Separation of Privilege
   a. Require multiple conditions to be satisfied before granting permission/access/*etc*.
   b. Advantage: 2 accidents/errors/*etc*. must happen together to trigger failure
5. Principle of Least Common Mechanism
   a. Minimize sharing
   b. New service: in kernel or as a library routine? Latter is better, as each user gets their own copy
6. Principle of Psychological Acceptability
   a. Willingness to use the mechanisms
   b. Understanding model
   c. Matching user's goal
7. ACM and primitive operations
   a. Go over subjects, objects (includes subjects), and state (S, O, A) where A is ACM
   b. Transitions modify ACM entries; primitive operations
      i. **enter** r **into** $A[s, o]$
      ii. **delete** r **from** $A[s, o]$
      iii. **create subject** s' (note $A[s', x] = A[x, s'] = ø$ for all x)
      iv. **create object** o' (note $A[x, o'] = ø$ for all x)
      v. **destroy subject** s'
      vi. **destroy object** o'
8. Commands
   a. **command** $c(s_1, ..., s_k, o_1, ..., o_k)$
      if r_1 **in** $A[s_1, o_1]$ **and**
      r_2 **in** $A[s_2, o_2]$ **and**
      ...
      r_m **in** $A[s_m, o_m]$
      then
      op_1;
      op_2;
      ...;
      op_n;

end.

b. Example 1: creating a file
command *create_file*(*p, f*)
    create object *f*;
    enter *Own* **into** *A*[*p, f*]
    enter *Read* **into** *A*[*p, f*]
    enter *Write* **into** *A*[*p, f*]
end.

c. Example 2: granting one process read rights to a file
command *grant_read*(*p, q, f*)
if *Own* **in** *A*[*p, f*]
then
    enter *Read* **into** *A*[*q, f*]
end.

9. What is the safety question?
    a. An unauthorized state is one in which a generic right *r* could be leaked into an entry in the ACM that did not previously contain *r.* An initial state is safe for *r* if it cannot lead to a state in which *r* could be leaked.
    b. Question: in a given arbitrary protection system, is safety decidable?
    c. Mono-operational protection systems: decidable
    d. Theorem: there is an algorithm that decides whether a given mono-operational system and initial state is safe for a given generic right.