1) Some operating systems have a feature that allows for encrypting the filesystem, either in whole or in part. This encryption is done at the filesystem level, and programs running on the system never notice it. The encryption key is derived from a password that the user enters at login. What sorts of attackers does this protect against, and what sorts of threats does this not protect against?

2) Which of these is a an example capability system, and which is an ACL-based approach? Explain each answer with a sentence or two.

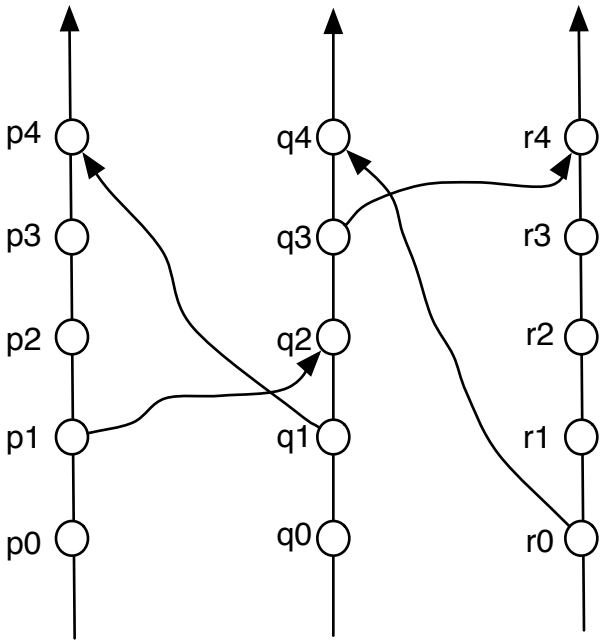
a) You give your friend a key to your apartment

b) A fancy club has a list of approved guests

c) Some dorms at Cornell have card-swipe access, where the magnetic code on the card is matched against a list of residents

d) Your car has a parking permit, listing where you're allowed to park.

3) In any distributed coordination problem, it is necessary to be able to determine the order in which two events occurred. In a distributed system it is sometimes impossible to say which of two events occurred first. The happened-before relation is used in obtaining a total ordering of events in a distributed system. The happened-before relation (denoted by ->) is defined as follows: first, if A and B are events in the same process, and A was executed before B, then A -> B. Second, if A is the event of sending a message by one process and B is the event of receiving that message by another process, then A -> B. Third, if A -> B and B -> C then A -> C .



a) In a centralized system with a single core processor, it is always possible to determine the order in which two events occurred, but in a distributed system this is not always possible. Why?

b)From the diagram below of relative time for three concurrent processes, what can you say about the relationship between q0 and p2? What can you say about the relationship between p0 and r4? (diagram attached)

c) For each of the three rules defining happened-before, substantiate why the rule must hold. (Hint: assume the property does not hold and discuss the consequence.)

4) You have a public-key private-key pair, with the public key known to the world, and the private key known only to you. You want to prove to a friend that you have the private key corresponding to the known public key. (Note. Assume that you are using a protocol like RSA, where either key can be used to encrypt or decrypt, with the other key doing the inverse operation).

a) Can you do this?

b) Can you do it in such a way that friend cannot subsequently use a transcript of the conversation to prove to a skeptical third party that you have they private key?

5) On older unix systems, passwords were stored hashed, salted, and in a publicly readable file. On more modern systems, the file is readable only by root.

a) Given that the file can't be read by any normal users, is there any benefit in storing only the hashed passwords?

b) Does this prevent root from impersonating users? If so how, if not why not?