Questions

1. (30 points) Consider the rule of transitive confinement. Suppose a process needs to execute a subprocess in such a way that the child can access exactly two files, one only for reading and one only for writing.
   
   (a) Could capability lists be used to implement this? If so, how?
   (b) Could access control lists implement this? If so, how?

2. (30 points) In the Janus system, when the framework disallows a system call, the error code EINTR (interrupted system call) is returned.
   
   (a) When some programs have read or write system calls terminated with this error, they retry the calls. What problems might this create?
   (b) Why did the developers of Janus not devise a new error code (say, EJAN) to indicate an unauthorized system call?

3. (40 points) Kernighan and Plauger argue a minimalist philosophy of tool building. Their thesis is that each program should perform exactly one task, and more complex programs should be formed by combining simpler programs. Discuss how this philosophy fits in with the principle of economy of mechanism. In particular, how does the advantage of the simplicity of each component of a software system offset the disadvantage of a multiplicity of interfaces among the various components?