1. The design and implementation of a system often holds a number of implicit as-
sumptions. What are some of the assumptions that Dijkstra made in the THE system
that the Hydra authors rejected? What aspects of Dijkstra’s approach persist in the
design of operating system components? Do you think Dijkstra would still have
advocated for strict layering given more complex hardware and usage scenarios?
Why?

2. Multics describes five design principles. How well does Hydra meet each of the five
principles? How about UNIX? Multics also describes a few major shortcomings, one
related to the complexity of the user interface. How do Hydra and TENEX fare with
their own user interfaces? How might the user interact with capabilities in Hydra?
What makes each interface more or less complex?

3. A fundamental aspect of protection in operating systems is rights amplification.
Rights amplification enables a more privileged protection domain to perform an
operation on behalf of a less privileged protection domain in a controlled fashion
without violating protection in the system. For each of the following operating sys-
tems, state (a) the protection domain that they support, (b) the mechanism for cross-
ing protection domains, (c) how rights are represented, (d) how rights are amplified
crossing domains, and (e) how the OS determines whether to allow the domain
crossing.

Support your answers with material from the papers (e.g., a quote of a phrase or
sentence), or with your own critical arguments, as appropriate (the Lampson pro-
tection paper is fair game). I am interested in your justifications as much as the
answer itself and the more often you can quote from a paper, the less you have to
justify. For instance, two possible answers to part (a) for Hydra are:

A protection domain in Hydra is the “local name space” (LNS). An
LNS represents the current set of objects and rights to which a process has
access, and those objects and rights change when a process moves from
one LNS to another.

A protection domain in Hydra is the “local name space” (LNS): "At
any instant, the execution environment (domain) of a program is defined
by an LNS object associated with it...the rights lists in each capability define the permissible access rights of this program at this instant." (Hydra p. 341).

- Medusa
- Multics
- Hydra
- Unix

4. The concept of a process is fundamental in operating systems, but the particular details vary widely. What makes the task-force organization of Medusa similar to the job structure in TENEX? What makes them different?