5. [7 points]: A pipe as described in the UNIX paper (Reading #9) can provide **(Circle all that apply)**

[✓ **A.**] Sequence coordination

B. Belady's anomaly

[✓ **C.**] Interprocess communication

[✓ **D.**] A technique to achieve modularity

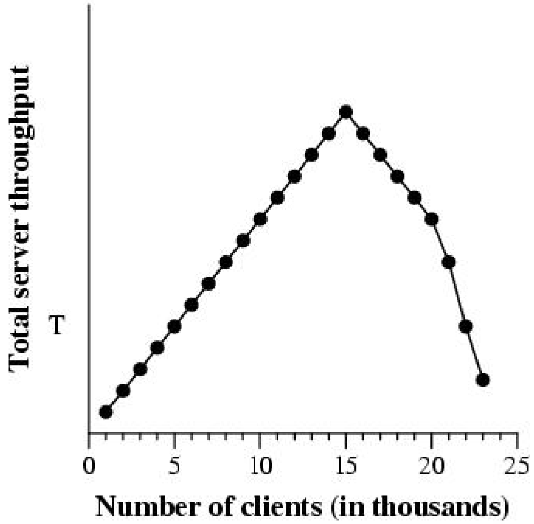


Figure 1: Total throughput versus number of clients.

How to review the papers

- What is the main points of each paper?
- What are the key terms and concepts?
- Consider for each systems paper:
  - structure of system
  - design decisions made and why
  - where are examples of the abstract topics we discussed in the system (modularity, sharing, resource management, hierarchy, client server, “worse is better” vs. “the right thing”)

“The Right Thing”

- Simplicity-the design must be simple, both in implementation and interface. It is more important for the interface to be simple than the implementation.
- Correctness-the design must be correct in all observable aspects. Incorrectness is simply not allowed.
- Consistency-the design must not be inconsistent. A design is allowed to be slightly less simple and less complete to avoid inconsistency. Consistency is as important as correctness.
- Completeness-the design must cover as many important situations as is practical. All reasonably expected cases must be covered. Simplicity is not allowed to overly reduce completeness.

“Worse is Better”

- Simplicity-the design must be simple, both in implementation and interface. It is more important for the implementation to be simple than the interface. Simplicity is the most important consideration in a design.
- Correctness-the design must be correct in all observable aspects. It is slightly better to be simple than correct.
- Consistency-the design must not be overly inconsistent. Consistency can be sacrificed for simplicity in some cases, but it is better to drop those parts of the design that deal with less common circumstances than to introduce either implementational complexity or inconsistency.
- Completeness-the design must cover as many important situations as is practical. All reasonably expected cases should be covered. Completeness can be sacrificed in favor of any other quality. In fact, completeness must sacrificed whenever implementation simplicity is jeopardized. Consistency can be sacrificed to achieve completeness if simplicity is retained; especially worthless is consistency of interface.

The Architecture of Complexity

True or False?

All truly complex systems have some hierarchical structure.

Hierarchy

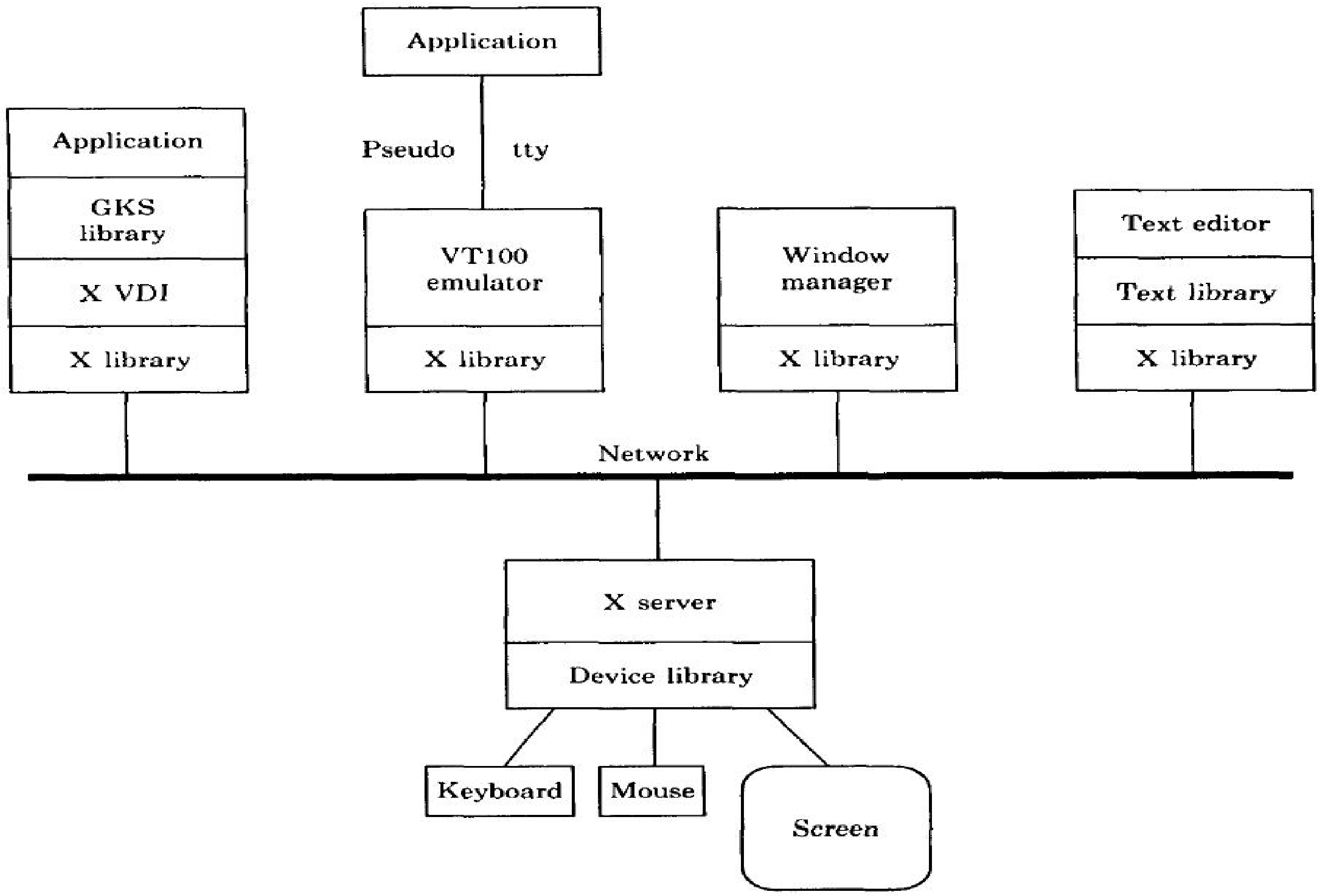
- Complex systems tend to evolve in the direction of hierarchy.
- Complex systems are more likely to survive if they are organized hierarchically.
- Questions to ponder:
  - What are examples of hierarchy in computer systems?
  - What are the arguments against building a system hierarchically?

Therac-25 Paper

“Complex systems fail for complex reasons.”

- Familiarize yourself with the structure of the system. What did each thread do? What were the specific failures that occurred?
- Compile a list of the faults that contributed to the disaster.
  - Operator interface problems, faults in software, system design approach, design iteration and problem handling approach, etc…

X-Window System



X Window System Provides:

- A device independent display applications programming interface
- A client/server boundary between the application and the display
  - Why do these actually matter?

Finally, look up the terms you don’t know in this paper and all the papers we read.