

Abstractions in Computer Systems

Tuesday, November 27, 2007
12:14 AM

- Three fundamental abstractions
 - o The memory, the interpreter, and the communication path
 - o Supply functions of recall, processing, and communication.
- Names are used in the interfaces of all these abstractions

- The memory
 - o Operations
 - WRITE(name, value)
 - value <- READ(name)
 - o Read/write consistency
 - The result of READ of a named object is always the value of the most recent WRITE.

- Interpreter
 - o Components
 - Instruction reference = tells the interpreter where to find the next instruction.
 - Context reference = tells the interpreter in which context to interpret the instruction reference and the names in the instruction itself
 - Repertoire = the set of actions the interpreter can perform
 - o Interpreters are always organized in layers, with the lowest being the machine, and the upper layers provide richer and specialized repertoire.

- Communication Path
 - o Let information flows between physically connected components.
 - o Operations
 - SEND(outgoing_message_buffer, destination)
 - RECEIVE(incoming_message_buffer, source)
 - o Can be thought of as a specialized form of memory. However, a communication path requires interpreters at both the receiving and sending ends.
 - o Semantics of SEND and RECEIVE can be different from READ and WRITE.