The Structure of "THE"-Multiprogramming System

1968
Edsger W. Dijkstra

Presenter: Hao Ren

History

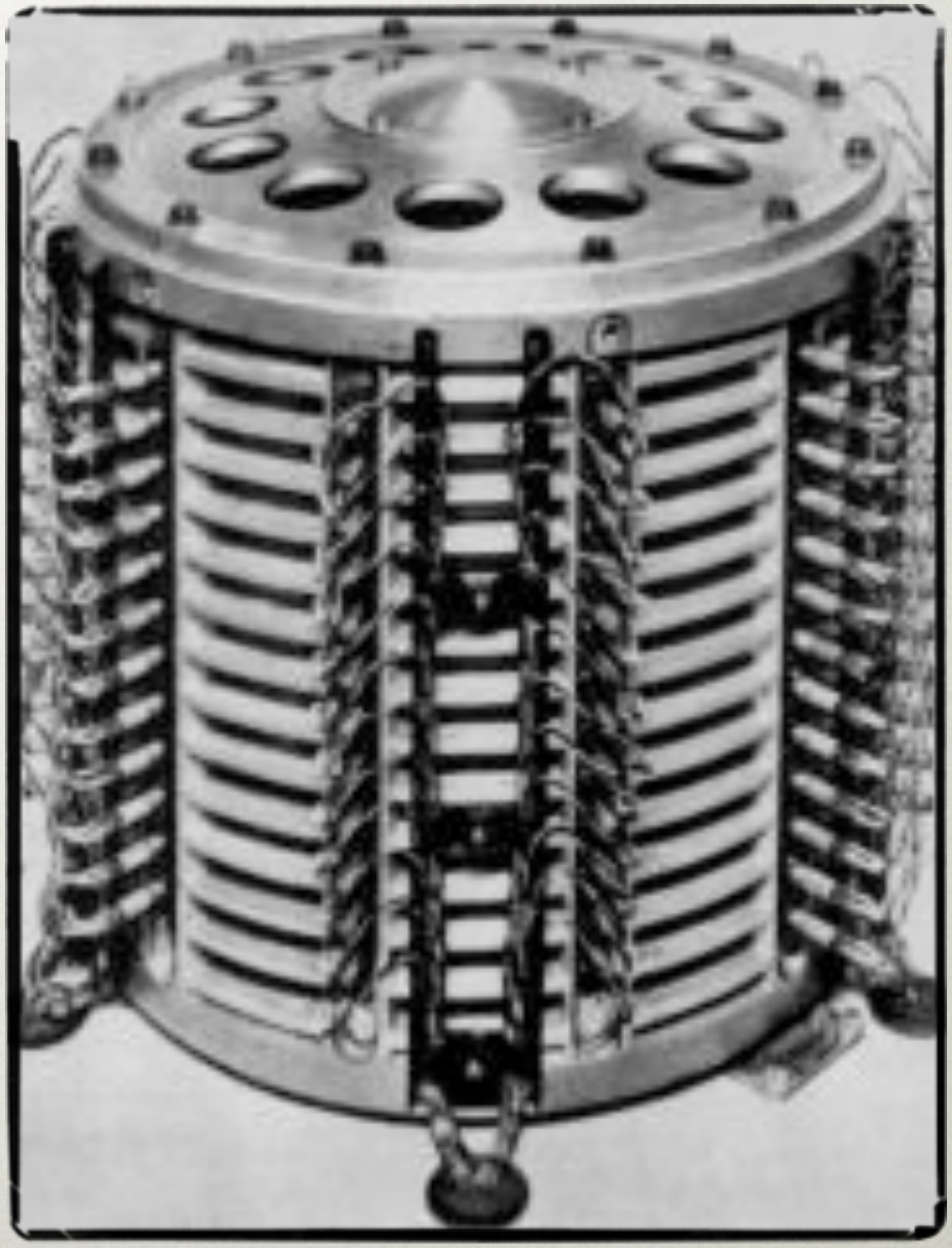
- 1955 - IBM introduces the 704, floating-point
- 1960 -Honeywell 800, with hardware support for timesharing between eight programs.
- 1962 -Atlas computer,first machine to use virtual memory and paging
- 1965 - Electrologica X8
- 1966-Control Data Corp. CDC 6500, containing two CDC 6400 processor

THE multiprogramming System

- THE:
  Technische Hogeschool Eindhoven:
  Eindhoven University of Technology
- Software-based memory segmentation
- Batch system system that supported multi-tasking
- not multi-user
- Use of structured Layer

Tools Comparison

- 32k of Core memory
- 512K drum for backing store
- paper tape readers
- Paper Tape punches



Goal

- Turn-around time
- Economic use of peripheral devices
- Automatic control of backing store
- Economic feasibility of a general purpose computer

Progress Report

- “Pathology”, management rules should be less refined.
- Irreproducible bug
- Formal Design process
  only trivial bugs

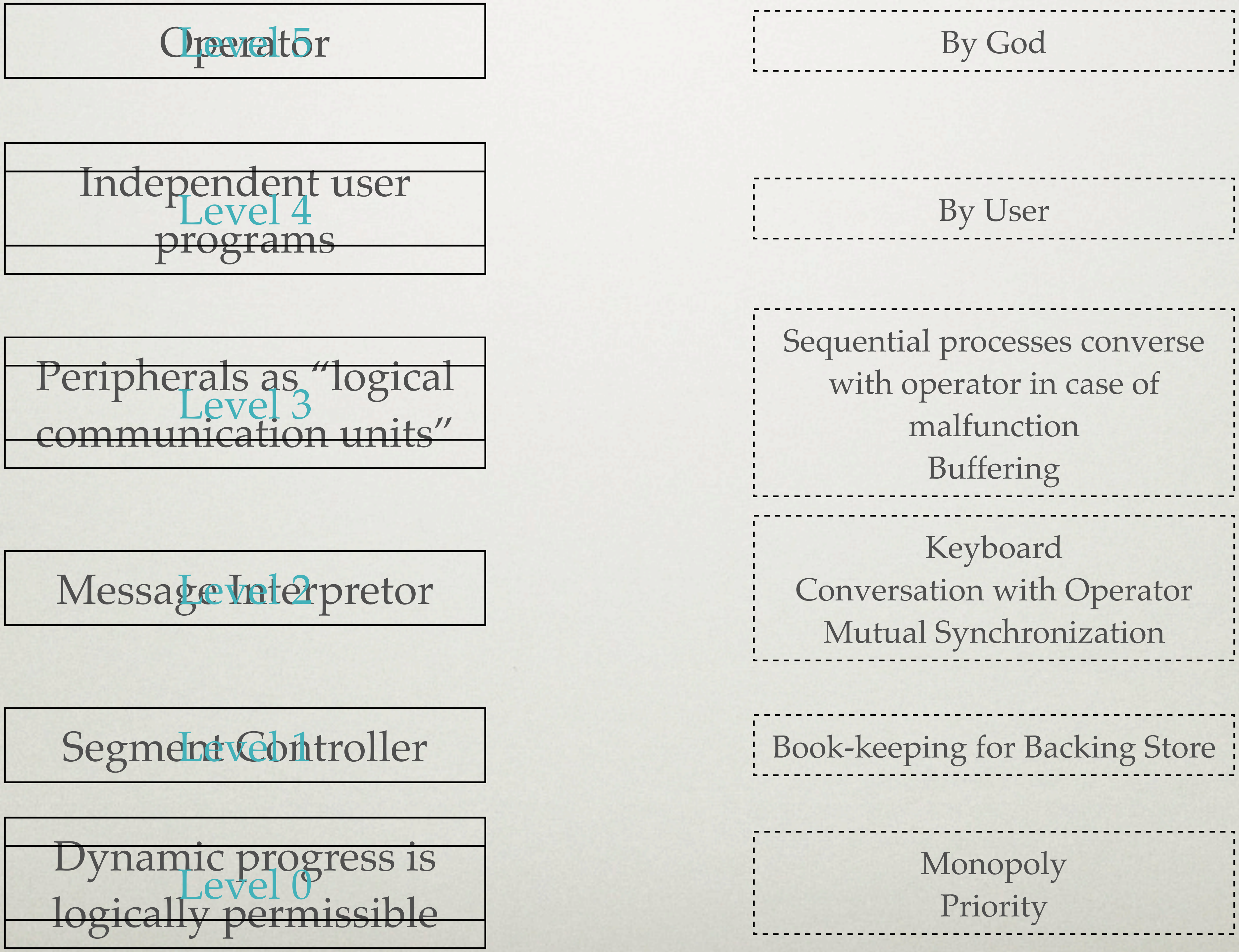
Formal Verification

- Model Checking: Exhaustive exploration of the mathematical model: e.g. state space enumeration.
- Logical inference: HOL theorem prover

System Structure

- Storage Allocation: secondary storage, virtual memory, segmentation
- Processor Allocation:
    - Undefined speed ratios:
      Not harmful to interior logic
      Ignores Responsiveness

System Structure System Hierarchy



Level 5
Operator

By God

Level 4
Independent user programs

By User

Level 3
Peripherals as “logical communication units”

Sequential processes converse with operator in case of malfunction
Buffering

Level 2
Message Interpretor

Keyboard
Conversation with Operator
Mutual Synchronization

Level 1
Segment Controller

Book-keeping for Backing Store

Level 0
Dynamic progress is logically permissible

Monopoly
Priority

System Hierarchy

- Make reasoning about the system more tractable (quasi-formal methods)
- Building and testing the system incrementally
- Subsequent OS that used layering: Windows NT, MAC OS X

Computer Multitasking

- Multiprogramming
  - Cooperative multitasking
    Microsoft Windows prior to Windows 95 and Windows NT, and Mac OS prior to Mac OS X
  - poorly designed program
- Real time
- Multithreading

Reference:

- Wikipedia: http://en.wikipedia.org/
- The History of the Development of Parallel Computing http://ei.cs.vt.edu/~history/Parallel.html

Questions

- More of a comment than a question, as I found it rather humorous that the author indicates
- "At the time this was written the testing had not yet been completed, but the resulting system is guaranteed to be flawless."
- and then later in the paper indicates:
- "I shall not deny that the construction of these testing programs has been a major intellectual effort: to convince oneself that one has not overlooked "a relevant state" and to convince oneself that the testing programs generate them all is no simple matter. The encouraging thing is that (as far as we know!) it could be done."
- How can one guarantee a flawless system yet openly admit that you *believe* you (as far as you know), that you have not overlooked a given state to test. The latter just doesn't have the same sense of confidence. I guess it all hinges on the definition of "flawless". ;-)

Questions

- Prior to the introduction of multiprogramming, what was the reigning operating system design?

Questions

- It seems like there is a lot of book-keeping (i.e. which process belongs to which task, etc), and there are also the five or so logical levels that are introduced in the system. Is that a major problem,performance wise?

Questions

- One of the author’s self reported mistakes is that they “conceived and programmed the major part of the system without giving more than scanty thought to the problem of debugging it” (page 2). We have touched on the point of debugging in events vs threads a few times this year. How much does this play a role in design and development of systems and/or programs