Multics- The First Seven Years

Emil Tsalapatis

September 11, 2018

Table of contents

Background

The Paper

Memory Management

Aftermath and Legacy

State of the Art in 1965

- Efforts towards interactive computing - CTSS
- Software-Hardware Codesign - Burroughs 5500
- Virtual Memory, Interactive Computing etc. introduced but not common

The Multics Project

- (1965) Introduction and Overview of the Multics System, by Corbato and Vyssotsky
- Computing power as a utility
- Reliability
- Built to last, i.e. portability and scalability

Functional Requirements

- Usability
  - Terminal Access
  - Hierarchical Storage
- Reliability
  - Continuous Operation
  - Data Integrity
- Versatility
  - Scalability wrt Hardware (across both machine classes and generations)
  - Scalability wrt Client Size
  - Different Programming Environments

Design Process

- Unprecedented Undertaking, except maybe for CTSS
- Standard engineering methodologies
- ...are what led to the project falling behind schedule
- Most requirements partially fulfilled at the time of the paper's publication

User View

- One user - one process
- User calls system-provided PL/I routines
- Usability Enhancements:
  - The compiler as a command, complete with error messages
  - Hierarchical File System

Technical Details

- Hot swapping hardware
- Virtual environments (e.g. APL, Dartmouth BASIC)
- Implementation almost fully in PL/I, in full PL/I
- Hardware Protection Mechanisms (Permission bits, Ring-Based Security Model)

Most interestingly:

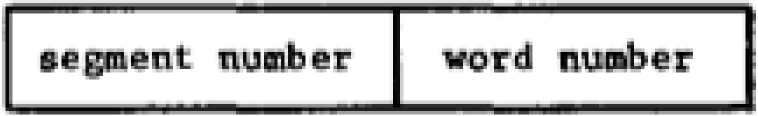
- Two-level Memory Management, Segmentation + Demand Paging

Physical Memory

- Up to 75 MB of Physical Memory
- Exorbitant Price per MB
- 36-bit words, 24-bit physical addresses

Credit: https://pdos.csail.mit.edu/6.828/2007/lec/l14.txt

Address Format

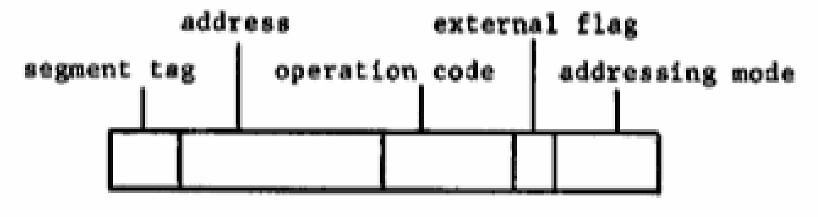


segment number	word number

Credit: Virtual Memory, Processes, and Sharing in Multics by Daley et al, retrieved from

https://multicians.org/daley-dennis.html

Instruction Format



Credit: Virtual Memory, Processes, and Sharing in Multics by Daley et al, retrieved from

https://multicians.org/daley-dennis.html

Segmentation

- Explicit Memory Management
- Everything is a segment, even files!
- IPC / Shared Libraries Made Easy

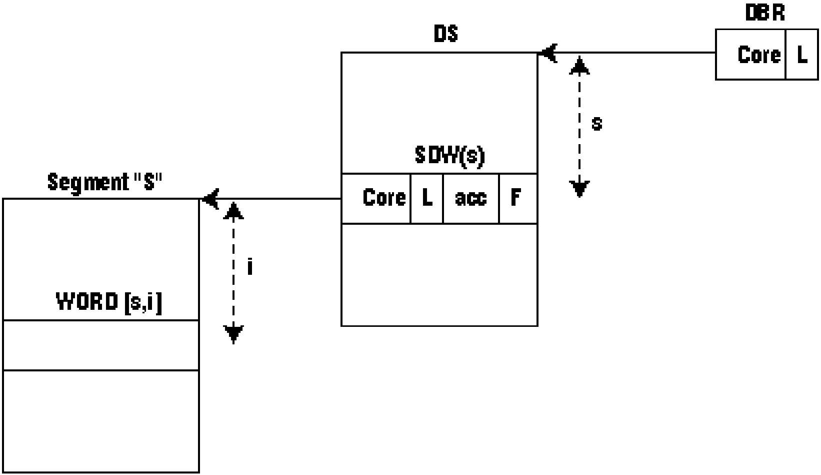
Segmentation (Cont.)

- Variable sized
- Addressable by symbolic name
- Name resolution can be deferred to runtime

Honeywell 645 Segmentation

- Variable Sized Segments
- Addressable By Name

Honeywell 645 Segmentation



Credit: The Multics Virtual Memory: Concepts and Design by Bensoussan et al, retrieved from https://multicians.org/multics-vm.html

Paging

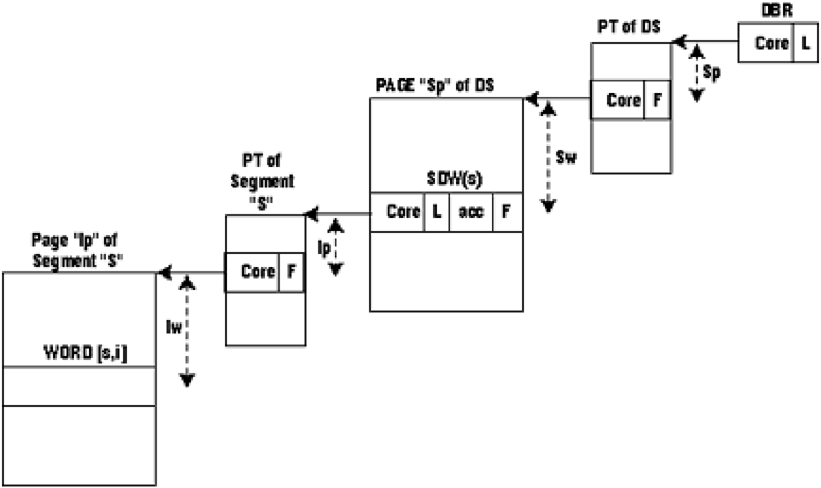
- Supervisor-initiated Memory Management
- Even the Supervisor can be paged out (partly)
- No sophisticated memory management algortithms yet
  - OPT algorithm for page replacement found in 1966
  - Belady's anomaly recorded in 1969
- Accelerated using a TLB

Honeywell 645 Paging

- No access bits, permissions are segment-based
- Per-segment page tables
- 64-entry page tables
  - 1K word pages
  - Segments constrained to 64K words

Credit: https://pdos.csail.mit.edu/6.828/2007/lec/l14.txt

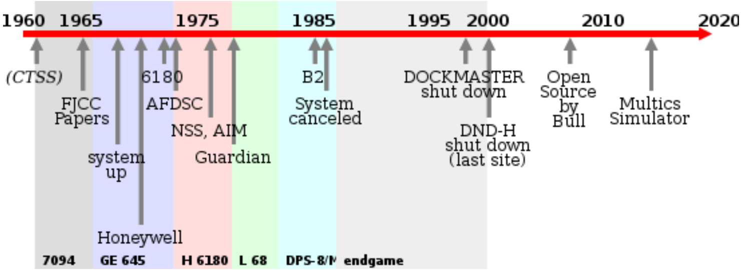
Honeywell 645 Paging



Credit: The Multics Virtual Memory: Concepts and Design by Bensoussan et al, retrieved from

https://multicians.org/multics-vm.html

Later Developments



Credit: Multicians Website, https://multicians.org/history.html

Intel x86

- Support for Multics Software Concepts
- 4 Protection Rings
- Call/Interrupt/Trap Gates (i.e. segment juggling)

Unix

- Different Design Goals, Different Target Platforms
- Technical e.g. IO Streams
- Nontechnical e.g. Communal Computing

Questions?