

1 Summary

Virtual Memory, Processes, and Sharing in MULTICS (1968): The authors introduce the concept of virtualized memory (VM), processes, and address space. They describe the mechanisms by which segmented memory works in a multiplexed information and computing service (MULTICS) OS, and how it works so that resources can be shared among a community of users from remote terminals. Their three main objectives for developing VM are 1) providing machine-independent VM managed by system software, 2) making programming more general by allowing procedure calls with symbolic names, and 3) permitting processes to interact closely and efficiently through procedure and data sharing.

2 Strengths

- Each figure the paper references does a good job of illustrating the different mechanisms they are describing, like the formation of generalized addresses, how segment addressing and descriptor segment lookup work, and how intersegment linking works. Since the paper introduces a lot of new concepts at the time of its writing, this is an important aspect of their paper which explains some of the VM implementation and decisions made.
- At the time of the writing, the concepts introduced were completely novel. Not only that, but they were implemented in a working computer system.

3 Weaknesses

- The way they explain the different mechanisms is confusing and verbose. For example, it is not very clear how the different addressing modes work without the context for which they would be used because they stick to high-level references when explaining new concepts.
- Their description of how paging works is very brief with no visual representation of how the page table is implemented, so it was difficult to figure out if what they describe is close to what might be modern paging. Compared to how the segments and linkages are described, this appears to be an outlier in the paper, despite the authors recognizing that paging would be necessary for when segments became too large for efficient storage.
- The paper does not evaluate any performance metrics so it is not clear if what they implemented for MULTICS was sufficiently efficient.

4 Rating: 3

5 Comments

A lot of the concepts introduced in the paper are familiar because they are still somewhat relevant today. Most of what they discussed either remained somewhat consistent with modern VM or have since been redesigned or replaced. For example, symbolic references are probably referring to the linking stage of compilation for unresolved procedure calls. While use of segmented memory is not as prevalent due to issues from external fragmentation, memory is still thought of as an address space. The paper also references use of procedure and data segments, which are likely now instruction and data memory, but with the similar read/write protections as the respective segments. And generalized addresses are what we call virtual addresses today, which are translated (quite differently) to the actual (physical) addresses in main memory. While not every feature of MULTICS made it to modern systems, it is very evident that this early system introduced a lot of new ideas and was an important influence in the development of VM and modern operating systems.