

Addendum to "Using Type Extension to Organize Virtual-Memory Mechanisms"

by P.A. Janson

When the original manuscript of this paper was written in 1977, two referees made some very constructive suggestions for revision. While it would have been easy, four years ago, to shorten the introduction, to move some of the conclusion material to a discussion section, and to improve style, as recommended, there is little incentive and no time to do so today. However, some specific remarks from the referees call for comment, and prompted the material in this Addendum.

First, it was recommended that the paper present one or two additional examples of use of type extension on the Multics without memory. Interested readers will find such information in other work published since 1977 [1].

Second, it was argued that the definition of type extension suggested in the paper is not standard. This is a valid remark. Usually, the operations defining an abstract data type are viewed as separate procedures that can be applied to objects of that type; the type manager is then only a syntactic entity. We have chosen to define the type manager as the set of procedures and data structures implementing the type to stress the identity between a type manager and a strict module, as used in the paper.

Finally, it was suggested that the concept of the map is confusing, is really an implementation question, and does not belong in the canonical

representation of an object. We have probably misstated the fact that the proposed canonical representation applies specifically to operating system objects, where it is fundamental. Indeed, the components (fields) of a record of structure, in the programming language sense, need not be pointed to by a map: they are stored directly in the record or structure itself. However, we are dealing with operating system objects such as disk records, core blocks, pages, segments, etc. These cannot be stored "in" the representation of the abstractions they compose. They must be pointed to by the representation (map) of the abstractions they compose. For instance, if segments are composed of pages, pages are not stored "in" the segments, which would imply that the pages of segments must be contiguous. Instead, they are pointed to by the map (page tables) of the segments they represent.

[1.] Schroeder, M.D., Clark, D.D., and Saltzer, J.H., The Multics Kernel Design Project, Proc. Sixth Symposium on the Operations System Principles, ACM Oper. Syst. Rev., vol. 11, no. 5.

Acknowledgements

I wish to thank J. Saltzer and M. Schroeder for their comments on the original version of the paper and for their encouragement and help to submit it to OSR in spite of the long delay. Many thanks also to the referees whose comments were so much appreciated though not implemented.

June 29, 1981