



## Bell Laboratories

subject: **Alternate Implementation Proposal for Unix/370**

date: **January 2, 1979**

from: **C. B. Haley**  
**MH 1271**  
**2C515 x3583**

**T. B. London**  
**HO 1353**  
**4E624 x2006**

**J. F. Maranzano**  
**MH 2524**  
**2C256 x3816**

**D. M. Ritchie**  
**MH 1273**  
**2C517 x3770**

Mr. B. G. Prieve:

A previous memorandum (B. G. Prieve, 8224-780928.01MF) proposes an implementation of Unix on the IBM System/370 using a modified version of the TSS resident supervisor. We suggest that a better approach would be to follow the path of the Unix implementations on the the PDP-11, the VAX 11/780, and the Interdata 8/32, using a virtual machine provided by VM/370. Our beliefs are based on the following considerations.

1. The VM approach is much simpler to implement because the same commonality of code (about 90%) can be expected in the IBM version as obtains in the other three. By contrast, using TSS will require essentially a complete rewrite of most of the system. Much of the new code will be of the sort that is most difficult to write correctly (and even very informally "verify"), i.e. process synchronization and mutual exclusion. We do not doubt that it is possible to make it work, but only point out that it won't be easy.
2. As a consequence, the VM version will be much easier to maintain. The current system is more or less continually changing as bugs are repaired, inefficient routines recoded, and new functions added. Today it is possible (with some effort) to keep the various versions in correspondence, but this will be much harder (in most cases impractical) with a totally unique version.
3. There is no particular reason to believe that a TSS-based Unix will be any more efficient than a VM-based version. It seems to be agreed that the problem of "shadow paging" would be the major source of inefficiency in VM-Unix, but the actual degradation to be expected is unknown. Moreover, the problem has evidently been solved, or reduced to manageable degree, for systems like CMS under VM; at Amdahl, where the approach advocated here is being followed, the implementors believe the problems can be controlled.
4. The TSS approach seems to have some potential performance problems of its own. For example, the synchronous nature of the IO interface to the kernel apparently

prevents read-ahead on disk files. Perhaps more important, the TSS-based system will be utterly dependent on a complex set of interlocks on the Unix system data that serve to keep the various processes from interfering with each other. In other operating systems for multi-processor machines, the overheads caused by such interlocks can be quite acceptable; however, all such systems have only a few (say fewer than four) processors. The proposed TSS-Unix, however, logically has tens or hundreds of processors (one for each active process). It seems quite likely to us that the queueing delays and other overheads introduced by this multitude of virtual processors may prove very burdensome indeed.

5. Everyone seems to agree that the only plausible way to get a system running in the first place is by using VM. However, the TSS-Unix proposal does not take into account the continuing advantages of VM. For example, TSS-Unix requires, in production use, a multi-million dollar machine totally dedicated to Unix: there can be no system development or testing, no on-line diagnostics, no IBM-supplied error-log processing programs. (It is true that one can conceive of running Unix under TSS under VM but this idea combines the worst of both proposals). Already there is one potential customer for Unix/370 within AT&T; their enthusiasm is conditional on running Unix under VM. There is a lesson to be learned from the number of MTS and TSS installations, which is almost negligible in spite of the fact that both systems seem to be vastly superior in many ways to the major IBM systems. This lesson is that it is very hard to compete head-on with OS and its descendants. VM-Unix offers the chance to have some of the cake and eat a bit of it too.
6. Although on the surface it appears that a direct adaptation of Unix to the 370 via the same path as was followed for the Interdata and the VAX is not a very interesting job, we feel there will be ample opportunities for invention in adapting even portable Unix for the 370. For example, the 370 C compiler could be improved quite significantly (e.g. by optimizing the use of base registers).

These arguments seem sufficiently strong to convince us that use of VM is the best way to implement Unix on the 370. In only one way does the TSS approach seem definitely superior: because a TSS-Unix *requires* the interlocks necessary to make multiple virtual processors work correctly, it permits operation with multiple actual processors. This advantage, although real, is not enough to make us change our preference.

C. B. Haley

T. B. London

J. F. Maranzano

D. M. Ritchie

MH-1271-CBH/TBL/JFM/DMR-troff

Copy to  
See next page

Copy to  
T. A. Dolotta  
A. G. Fraser  
W. A. Felton  
R. C. Haight  
S. C. Johnson  
C. A. Mast  
N. A. Martellotto  
M. D. McIlroy  
S. P. Morgan  
J. F. Reiser  
W. C. Ridgway  
C. S. Roberts  
J. M. Scanlon  
C. T. Schlegel  
N. R. Sinowitz  
B. A. Tague