p.L

TO:
A. Bensoussan
F. J. Corbató
C. T. Clingen
S. Dunten
R. J. Feiertag
C. Garman
J. W. Gintell
N. I. Morris
J. H. Saltzer ✓
S. Webber
D. Vinograd

FROM: Mosley Meer

DATE: April 23, 1971

SUBJECT: KST Modification

This is a proposal to modify the KST so that a "normal" user references fewer pages when using it.

To accomplish this the entry array and hash tables are broken into small chunks and allocated as needed. For a 128_{10} entry array, 256_{10} entry uid hash table and 512_{10} entry name hash table all three tables plus the header can be fitted on one page.* A second page is required for the names and possibly a third page or more if segments are made known with many names.

A header is shown in Figure 1 which will allow up to eight small entry arrays and hash tables.

A modification to the KST entry is shown in Figure 2. This cuts the entry size in half (to 4 words) and contains a count of null names rather than using name structure for them.

* This assumes a small array size of 128 entries with an average of two names per entry.

The name struture will remain the same but an additional size of 64 characters (besides the 32 and 168) will be allowed. Pathnames will be stored in the smallest size name structure which will hold them.

Hash Tables will be allocated when all existing hash tables are half full. Hash tables will be searched successively, and a "not found" code will be returned only after all hash tables have been searched.

Inability to allocate name structures, hash tables or entry arrays will return a KST full code.

Can you estimate how many entries will cause overflow?

HEADER

highseg	highest segment number used
hcscnt	hard core seg count
npiece	number of arrays and hash tables which may be allocated
asize	small array size
esize	entry size
hcname	small name hash table size
hcid	small uid hash table size
huname(0)	name hash entries used
huname(7)	
huid(0)	uid hash entries used
huid(7)	
pad18	areap
freep	reservp
kstap (0)	kstap (1)
kstap (6)	kstap (7)
hpname(0)	hpname(1)
hpname(6)	hpname(7)
hpid(0)	hpid(1)
hpid(6)	hpid(7)
srulep(0)	srulep(1)
srulep(6)	srulep(7)

ONE KST ENTRY

0		NULL NAME COUNT				
1	NAME RELATIVE POINTER	TUS	TMS	TSW	DIR SW	INFERIOR COUNT
2	PARENT SEG NUM	BRANCH ENTRY OFFSET				
3	UID					

Word 0 contains forward and backward relative pointers for threading the free or reserved list when the entry has been used but is now empty.

HEADER	$56_{10}=70_8$ words
AREA INFO	
UID HASH	$128_{10}=200_8$ words
NAME HASH	$256_{10}=400_8$ words
ENTRY ARRAY	$512_{10}=800_8$ words for 128 entries
REST OF SEGMENT IS AREA FOR ALLOCATION OF NAMES, HASH TABLES AND ENTRY ARRAYS	