J.H. Saltzer

Identification

fsm

S. H. Webber

I must [illegible] with reset call.

# of pages ceiling, AST size?

Purpose

This command is used to meter certain file system variables and functions. It can be invoked from any user process (by using differences stored in internal state) and directs I/O to the device specfied for 'user_output'.

There are three entry points which perform as follows (from command level):

1. fsm

   This entry must be called before either of the other two entries (for each new process in which fsm is invoked). It sets up necessary variables in internal static. It may be called more than once.

2. fsm$reset

   This entry 'resets' the file system meters kept in this process's internal static area. It is called at the time when it is desired that metering <u>begin</u>. It is effectively called at bootload time so that initialization may be metered.

3. fsm$interpret

   This entry prints out the metering results for the period of time starting at the last call to fsm$reset (or to bootload time if fsm$reset has never been called). The meters are <u>not</u> reset so that cummulative metering may be done. An explicit call to fsm$reset must be done to reset the meters.

   The following is a list of the meters printed out as well as other file system information also printed out.

   <u>Total time</u>: The total time of the metering run. (Not accurate unless fsm$reset [illegible] called not at least one once.)

AST grace: This is the average number of microseconds an AST entry remains after the last page for that segment has been paged out and before the AST entry is relinquished.

Used ASTs: This is the number of AST entries on the AST used list.

Free ASTs: This is the number of AST entries on the AST free list.

Held ASTs: This is the number of AST entries with the 'entry hold switch' on.

Steps: This is the total number of steps taken by 'page' around the used core list. A 'step' consists in moving the 'used pointer' to the next entry on the list.

Needs: This is the total number of times a page of core was needed. It is a count of the number of page faults which were of a non-trivial type (e.g., the page was already on its way into core).

Invert

Ave. Steps: This is the average number of steps taken in handling a page fault.

Ceiling: This is a count of the number of times 'page' hit the 'ceiling'. The 'ceiling' is a certain maximum number of writes that may be queued simultaneously.

Print its value at reset time.

Ctwait: This is a count of the number of times the 'write counter' in a page table word was full. This occurs when too many write requests have been queued for the same page. (Note: this is very unlikely.)

Laps: This is a counter of the number of times the used list pointer has gone around the used list.

Skip wired: This is a count of the number of times a page was skipped over (while scanning the used list) because the page was temporarily wired down.

Skip used: This is a count of the number of times a page was skipped. because it was used in the last cycle.

Skip mod: This is a count of the number of times a page was skipped because it was 'out of service' because i/o was going on.

Drum reads: This is a count of the number of reads from the drum.

Drum writes: This is a count of the number of writes on the drum.

Disk reads: This is a count of the number of reads from the disk.

Disk writes: This is a count of the number of writes on the disk.

Drum errors: This is the number of drum errors since bootload time.

Disk errors: This is the number of disk errors since bootload time.

not reset?

Drum left: This is the number of unused 1024 word blocks remaining on the drum.

Disk left: This is the number of unused 1024 word blocks remaining on the disk.

A No. Pages: This is the total number of pages in the paging environment.

B Wired pages: This is the number of pages temporarily wired down.

Do I subtract B from A?

How about total # 1
AST entries made?