MULTICS STAFF BULLETIN-12

TO: Distribution

FROM: Melanie Weaver

DATE: February 3, 1972

SUBJECT: Changes to Multics Standard Tape DSM

In attemtping to make the Multics standard tape DIM publicly useable, several bugs and deficiencies have been uncovered. Given below is a list of these, along with their planned corrections. "interim" refers to those things to be done in the next few weeks. It does not mean temporary unless there is also a "target" specification. (Note that status refers to the 72 bit argument to ios_.)

I. EOT (end of tape) marker reached while writing

now	set end_of_data bit in status; set code to error_table_$device_end; write out set of buffers currently being processed
interim	set end_of_data and device_end bits in status (device_end bit is newly defined); set code to error_table_$device_end; write out set of buffers currently being processed

target	set device_end bit in status; set code to 0; write out set of buffers currently being processed

II. Writing beyond EOT marker

now	DSM does not restrict write requests after it has reached marker
interim	allow only detach request after marker is reached; set end_of_data and device_end bits in status; set code to error_table_$device_end
target	allow only detach request after EOT marker is reached; set device_end bit in status; set code to 0

III. Record header of tape trailer record (written on detach call)

now	eor (end of reel) admin bit is set
interim	if byond EOT marker, set both admin bits end_of_data (now called eor) and device_end (now called eot but not used); otherwise, just set end_of_data admin bit
target	if not on last reel of logical (possibly

multi-reel) tape, set device_end bit;
otherwise, if beyond EOT marker, set
end_of_data and device_end bits;
else just set end_of_data bit

IV. tape_trailer record encountered on read

now
a) if nelem words have already been read in, no code or status is returned;
b) otherwise set code part of status to error_table_$device_end set end_of_data bit in status

interim
set status bits to reflect end_of_data and device_end admin bits in tape trailer record header;
set code part to error_table_$device_end

target
status bits set to reflect end_of_data and device_end admin bits in EOR record header;
set code to 0

V. Reading blank tape (concerns attach entry on read)

now
attach forward spaces file to skip tape label, which causes the whole tape to be "read" and takes about 3 minutes

interim	try to read tape label, and if get hardware status for blank tape, set end_of_data bit in status; set code part to error_table_$blank_tape; detach tape (since it would not be fully attached)

VI. On read, if unique id read < first unique id on tape

now	set end_of_data bit in status
interim	set end_of_data bit in status; set code part of status to error_table_$ data_improperly_terminated

VII. Reading partially blank tape

now	tape_ tries to read 64 records and then sets end_of_data bit
interim	set end_of_data bit in status; set code part of status to error_table_$ data_improperly_terminated

VIII. On read, when get >64 consecutive records not in Multics standard tape format

now	set end_of_data bit in status
interim	set end_of_data bit in status; set code part to error_table_$improper_ data_format

IX. On read, when get >64 consecutive data alerts, or mixture of data alerts and bad formats

now	set end_of_data bit in status

interim	set end_of_data bit in status; set code part of status to error_table_$ device_parity; if another read request is made, attempt reading of next logical record (this is what happens now for some kinds of hardware status)
target	set code part of status to error_table_$ device_parity; if another read request is made, attempt reading of next logical record

X. Reading partially or fully blank tape through nstd_

now	when it gets to blank portion, it back- spaces and tries to re-read a record 10 times
interim	set code part to error_table_$blank_tape; set end_of_data bit in status

The following list concerns miscellaneous deficiencies which we will correct.

1) Implement "seek", but only for setting read and write to 0 (rewinding), so order call need not be used (do not allow change of mode).
2) Before attaching, check to see if the caller is highly privileged; if so, use hphcs_$tdcm_priv_attach; otherwise, use hcs_$tdcm_attach.
3) Create temporaries with intelligible names (tape_temp_1, etc.); delete temporaries when detach.
4) When an error is detected in the attach entry, call hcs_$tdcm_detach to detach the drive (when relevant), in addition to detaching the steam (currently, only stream is detached).
5) Maintain error count of rewrite attempts (currently omitted); enforce maximum of 64 rewrite attempts per record to correspond with reading strategy (currently no maximum is enforced).