TO: Distribution

FROM: Gary C. Dixon

DATE: April 22, 1974

SUBJECT: ANSI Standard Tape DIM Project Status

I.   SIGNIFICANT EVENTS

Janice Phillipps has completed her work on the ANSI DIM's file positioning and label handling code. This code is a large fraction of the total DIM and its completion is a noteworthy milestone of the project. Work will begin now on merging this code with Ross Klinger's logical record blocking package.

II. TASK STATUS

Attach/Detach - File Positioning -
Label Handling (Janice Phillipps)

As stated above, Janice has reached a major milestone in the ANSI DIM implementation. Tape interchange between Multics and OS/370 has been demonstrated, using an ANSI standard labelled tape with multiple files recorded in U-format at 800 bpi. An ANSI volume created on Multics has been read under OS/370, and one created by OS/370 has been read through the ANSI DIM on Multics.

lrec_io_ (Ross Klinger)

lrec_io_, the logical record blocking/deblocking module for the ANSI DIM, has been divided into two modules: ansi_lrec_io_, which handles blocking for the ANSI Tape DIM; and ibm_lrec_io_, which handles blocking for an IBM OS/DOS/370 Tape DIM. The IBM DIM is a planned extension of the ANSI DIM. We had originally intended that it would be invoked by using special control arguments in the attachment modes given to the ANSI DIM. However, the restrictions on logical record length and block length, imposed by OS's handling of physical records which have been padded out to word boundaries, has necessitated a division of the interfaces. The ANSI and IBM DIMs will share all code, except for the processing of attachment modes and the blocking/deblocking of physical blocks. However, each DIM will

---

be invoked separately and will have its own user documentation to avoid confusion between their somewhat different requirements.

Testing the tape information interchange capabilities between Multics and OS/370-MVT has been completed. OS is able to read all of the tape formats generated by the IBM DIM on Multics, as long as the restrictions built into the DIM requiring 0 mod 4 logical record and block lengths are satisfied. When V-, VB-, and U-format records are used, some logical records must be padded with blanks to make the physical records an integral number of words. For VBS-format, only the final logical record need be padded. Of course, F- and FB-format logical records must be padded out to the logical record length, which must be 0 mod 4. Multics has no problems reading IBM format tapes generated on OS/370, irregardless of their record length, since the Multics tape hardware pads byte-aligned records read from a tape out to word boundaries and the IBM DIM on Multics ignores this padding. (It is superior to OS's tape handling in this respect.)

Testing of ANSI F-, D-, S-, and U-format record interchange between Multics and OS/370 is also complete. IBM does not support Version 3 of the ANSI Standard for Magnetic Tape Interchange, which allows physical blocks to be padded out to word boundaries with ANSI pad characters ($\tilde{\ }$). IBM only supports Version 1 of the interchange standard (except for a programmed product which can be used to deblock Version 3 records). Rather than put special code in the ANSI DIM to support OS/370's idiosyncrasies, we have elected to faithfully follow the Version 3 ANSI Standard (in the hopes that OS and other systems will support it someday), and we will recommend that information interchange between Multics and OS be carried out through the IBM DIM. Of course, because the Multics tape reading hardware pads physical records out to word boundaries, reading ANSI labelled tapes generated under OS/370 is no problem.

merging the code (Janice Phillipps and Ross Klinger)

The next step in the development of the ANSI DIM is to merge Janice's file-positioning, label-handling code for the DIM with Ross's logical record blocking/deblocking code. The first step in this merger is to surround Jan's sync_tape_io_ program by a subroutine whose interfaces simulate the operation of Frank Canali's tapeio_ program. These tapeio_ interfaces are already used by Ross's code, and will eventually be used by Jan's code.

To avoid delays in testing the merged code, Jan has coded a write-around for the tape mount interfaces which does not depend upon the switchable gate environment created by Dennis Capps to debug the tape mount package.

tape_in/tape_out (Ross Klinger)

Coding of the installation-maintained commands, tape_in and tape_out, is complete and in the final debugging stage. Non-labelled, multi-file volumes have been successfully read and written in sequential order. The first version snould be complete early in May. It will permit non-sequential access to files on tape, and will use PL/I I/O through the vfile_ DIM to access files in the Multics Storage System. Only non-labelled tapes will be supported initially (although labelled tapes can be read by considering each label group as a separate file).

tapeio_ (Frank Canall)

The design review has been completed for tapeio_, the new interface to tdcm_ which will be used by the ANSI DIM (and by other DIMs in the future), and coding is well underway. Coding of the user ring interfaces is nearly complete, and coding for the ring 1 interfaces to be used by the tape mount package will be complete in early May. Then the tape mount package will be modified to use these interfaces, and debugging of the entire package will commence, using Dennis Capp's switchaole gate testing environment. Completion is expected in late May, followed by documentation during June. An MCR for tapeio_ will be submitted in the very near future.

Implementation plan (all)

This week, the implementation plan for the ANSI DIM will be completed, and an MTB describing the plan and outlining the features of the ANSI DIM (and the IBM DIM) will be published in May. An MCR will be submitted shortly thereafter.

TASK DESCRIPTION	PERSONNEL	START	FINISH	M-M	CHANGES-STATUS
ATTACH: attach (write) a new volume and write a single file	JP	02/01/74 01/28/74	02/07/74 02/05/74	.5	Checked out.
ATTACH: attach (read) a file of a single-file volume	JP	02/08/74 02/05/74	02/14/74 03/06/74	.5	Checked out.
ATTACH: attach (write) to append a file to a multi-file volume	JP	02/15/74 02/01/74	02/19/74 02/05/74	.25	Checked out.
ATTACH: attach (read) a file of a multi-file volume after label chain has been constructed	JP	02/19/74 02/19/74	02/21/74 03/11/74	.25	Checked out.
ATTACH: test for bad attachment modes	JP	03/11/74 03/11/74	03/13/74 03/12/74	.25	Checked out.
ATTACH: attach (read) a file not found on tape	JP	02/26/74 03/19/74	02/28/74 03/07/74	.25	Checked out.
ATTACH: attach (write) a file already on tape	JP	03/14/74 03/22/74	03/20/74 03/29/74	.5	Checked out.
DETACH: test detaching file with unload and leave disposals.	JP	03/06/74 02/05/74	03/08/74 02/08/74	.13	Checked out.
DETACH: Test detaching file with disposals rewind and reread.	JP	03/20/74 03/18/74	03/21/74 03/29/74	.13	Checked out.
ORDERS: test order calls	JP	03/21/74 03/18/74	03/25/74 03/29/74	.25	Checked out.
WRITE: write file (ASCII, 9-mode, U-format)	JP	03/13/74 02/06/74	03/23/74 02/08/74	.75	Checked out.
READ: read file (ASCII, 9-mode, U-format)	JP	03/23/74 02/07/74	04/02/74 03/06/74	.75	Checked out.

TASK DESCRIPTION	PERSONNEL	START	FINISH	M-M	CHANGES-STATUS
Create and run test sequence to attach and detach several different tapepositions and combinations of positions.	JP	03/25/74 03/25/74	04/01/74 04/02/74	1	Checked out.
MULTICS-OS/370: write multi-file volume on Multics, read into OS; write on OS, read into Multics	JP	04/02/74 04/01/74	04/09/74 04/04/74	.5	Checked out.

TASK DESCRIPTION	PERSONNEL	START	FINISH	M-M	CHANGES-STATUS
lrec_io_: ANSI Standard Tape DIM read/write interface for U, F, D, V, and S record formats	Klinger	12/10/73	12/18/73	1.4	Done
file_ DIM simulator for asynchronous tape io	Klinger	12/19/73	12/21/73	.6	Done
testing with file_ DIM	Klinger	12/26/73	01/09/74 / 01/07/74	1.4 / 1.1	Done
nstd_ simulator for asynchronous tape io	Klinger	01/10/74 / 01/08/74	01/14/74 / 01/11/74	.6 / .8	Done
testing with nstd_	Klinger	01/15/74 / 01/14/74	01/23/74 / 01/30/74	1.0 / 2.0	Done
cv_char_: alm subroutine for ascii - ebcdic / ebcdic - ascii conversion	Klinger	01/18/74	01/22/74	.6	Done
tape interchange testing: Multics tapes on OS; OS tapes on Multics	Klinger	02/04/74 / 02/04/74	02/22/74 / 04/05/74	3.0 / 8.0	Done +
Interim lrec_io_: bridge between ios_ and iox_ using tapeio_ interface	Klinger	03/11/74 / 03/11/74	03/22/74 / 03/18/74	2.0 / 1.0	Done +
Research and recode cv_char_ conversion tables	Klinger	03/12/74	03/13/74	.4	Done
split lrec_io_ into ansi_lrec_io_ and ibm_lrec_io_	Klinger	03/25/74 / 03/19/74	04/05/74 / 03/27/74	2.0 / 1.4	Done +
nstd simulator for tapeio_	Klinger	03/25/74 / 03/28/74	03/28/74 / 03/29/74	.8 / .4	Done +
sync_tape_io_ simulator for tapeio_	Klinger	04/24/74	05/08/74	2.0	+
SPS documentation of lrec_io_ and cv_char_	Klinger				ongoing task

TASK DESCRIPTION	PERSONNEL	START	FINISH	M-H	CHANGES-STATUS
inclusion of ansi_irec_io_ in ANSI DIM	Klinger	04/23/74			+
irec_io_ testing with asynchronous tape io package	Klinger				when asynchronous tape io package becomes available

TASK DESCRIPTION	PERSONNEL	START	FINISH	M-W	CHANGES-STATUS
tapeio_: design asynchronous tape I/O interface for ANSI DIM	Canall	12/25/73	01/14/74	1.5	Design Review complete
		12/25/73	03/12/74	3.0	
tapeio_: code subroutine	Canall	04/01/74	04/30/74	4	Ring 4 interfaces almost coded
		03/13/74	05/03/74		
trm_ (tape mount package): modify to call tapeio_'s ring 1 interfaces	Canall	05/06/74	05/10/74	1	
tapeio_: debug ring 4 and ring 1 interfaces, and trm_ mods	Canall	05/13/74	05/31/74	3	
tapeio_: PLM documentation	Canall	06/03/74	06/30/74	3	

TASK DESCRIPTION	PERSONNEL	START	FINISH	M-W	CHANGES-STATUS
Design phase: extension of ostape to a general tape I/O command.	Klinger	02/07/74 02/07/74	02/15/74 02/20/74	1.4 2.0	Done
Coding phase: general tape I/O command.	Klinger	02/18/74 02/21/74	03/01/74 03/14/74	2.0 3.1	Done
Testing phase: general tape I/O command.	Klinger	03/04/74 04/01/74	03/15/74 04/12/74	2.0 2.0	Done: reads and writes multi-file volumes in sequential order in all formats
Implementation Step I: storage system io using vfile_; absolute file positioning	Klinger	04/22/74	05/01/74	1.6	