The Michigan Terminal System (MTS) is a large time-sharing operating system that runs on System/370 and System/370-compatible hardware. It was developed at the University of Michigan at Ann Arbor particularly to support the academic, research, and administrative requirements of a university. It features a friendly and easy-to-learn command language, extensive file sharing capabilities, support for a wide variety of terminal types and attachments, and a batch processing mode. In 1989 it is in use at a dozen or more sites worldwide.

History

In 1967, the University of Michigan was one of the first installations of the System/360 Model 67, the first IBM machine to support virtual storage. The Time Sharing System, or TSS/360, was the operating system designed and marketed by IBM to accompany the 360/67 and the University of Michigan expected to run it. However, TSS was late begin delivered, and while waiting UM began the development of MTS, using as a basis a small System/360 supervisor developed at the MIT Lincoln Laboratory. Ultimately TSS proved to be very late, and by the time it was delivered MTS was a full-fledged and entrenched operating system at the university. It remains so to this day.

About 1970, the University of British Columbia was shopping for an operating system for their central computing service, and chose MTS. Since then, UM and UBC have been the primary development sites for MTS, both contributing about equally to its development.

Following UBC's entry into the MTS arena, another four universities in North America and two in England also chose MTS. As of 1986 in addition to UM and UBC, the University of Alberta (Edmonton), Simon Fraser University (Vancouver), Wayne State University (Detroit), University of Newcastle Upon Tyne (Newcastle, England), University of Durham (Durham, England), and Rensselaer Polytechnic Institute (Troy, New York) use MTS as their primary central-site operating system. MTS sites also exist in Brazil, at Hewlett-Packard, and at NASA.

Over the span of years MTS has run on a variety of hardware. When they outgrew the 360/67, UM converted MTS to run on the dual-processor System/360 Model 67-2 (duplex). This configuration was later used at UBC as well. Later, UBC ported MTS to the System/370 architecture when they installed a System/370 Model 168 in 1972. A 370/168 was in use at the University of Newcastle as late as 1985.

For the period from 1972 until the announcement of the IBM 3033 in 1977, the 370/168 was the largest uniprocessor offered by IBM. MTS only supported uniprocessors in the System/370 architecture, even though the 360/67-2 was a multiprocessor. MTS has always run in large shops on the largest available mainframes. By 1975, UM, and shortly thereafter, UBC and University of Alberta, began to outgrow their 370/168 uniprocessors, and turned to Amdahl Corporation, which was just beginning to market the 470/V6 product line, for relief. Amdahl 470s were eventually installed at all three sites, and Wayne State University as well. These machines also continued to grow, first into 470/V7s and later 470/V8s, through the latter half of the 1970s and into the 1980s. By 1980, Amdahl was the predominant mainframe vendor at all of the large state- and provincially-run universities running MTS.

In 1977, Rensselaer Polytechnic Institute, RPI, became an MTS site, replacing CP/67 and OS/MVT on a 360/67 with MTS. Shortly thereafter, in a major computing upgrade, the 360/67 was replaced with one of the first-shipped IBM 3033Us. Countering the by then established trend, RPI chose IBM over Amdahl due to competitive pricing and a long-standing research relationship with IBM.

Over the period from 1977 to 1983, the MTS community, as it is now called, followed industry trends by installing larger and larger machines. By the end of 1983, both UM and University of Alberta had installed Amdahl 580 processors; in fact, UM was the first customer installation of an Amdahl 5860. MTS had become increasingly adapted to the Amdahl products, taking advantage, for example, of

31-bit addressing on the Amdahl 470 and 580 to address more than 16 megabytes of storage.

Also in 1983, RPI had upgraded the 3033U several times and was looking for a successor machine. The IBM product line presented a problem. As was the case in 1975 with the 370/168, there was no IBM uniprocessor larger than the 3033U to which to migrate. IBM bid the dyadic 3081D, while Amdahl and National Advanced Systems both bid uniprocessors. RPI might have ceased to be an IBM account at that time except that the decision was made to convert MTS to again support a multiprocessor, and to do so under the (at the time) just-announced System/370 Extended Architecture (XA).

The 3081D was installed at RPI in December 1983 and initially ran MTS on a single processor in 370 mode. The other processor sat unused. Some changes were made to support the 4K storage key size and 64K-only segment size of the 3081D. Work commenced on the conversion of MTS to XA, and by the beginning of the following academic year in September 1984, MTS/XA was running on both processors of the 3081D.

Since then, the conversion to XA has proceeded apace at virtually all MTS sites, and the System/370 mode is little-used. The system currently operates on the following hardware:

- IBM 3090-600E

- IBM 3090-200S

- IBM 3090-180E

- IBM 3081GX

- IBM 3083B

- IBM 4381-Q13

- IBM 4361 Model Group 5

- Amdahl 5890

- Amdahl 5870

- Amdahl 5860

Current MTS Capabilities

Currently MTS is distributed as a single system that can run either in System/370 or 370-XA modes. The switch between modes can be accomplished in about one-half hour through reassembly of about a dozen critical modules. Therefore it is not the practice to refer to separate versions of MTS/370 or MTS/XA as they are both the same operating system.

In System/370 mode, MTS can support the following features:

1. Only a single central processor (i.e., no AP or MP support)

2. Real storage up to 16 megabytes, unless on an Amdahl 470 or 580 machine, in which case up to 64 megabytes of real storage are supported using the Extended Real Addressing feature

3. Both 2K-block and 4K-block storage keys

4. Both 1 Mb and 64 Kb segment sizes

5. The 3082 processor controller.

6. The 3092 processor controller.

7. The 3090 Vector Facility.

8. Expanded storage.

The latter support was developed to accomodate the 308X processors which support only the 64K segment size in System/370 mode and only 4K storage keys.

In 370-XA mode, MTS can support the following features:

1. From 1 to 32 central processors, dynamically determined at IPL time

2. TOD clock synchronization

3. Real storage up to 256 megabytes, using 31-bit addressing

4. XA 4K storage keys

5. XA virtual address translation format (1M segment size, segment tables, and page tables)

6. XA dynamic channel subsystem with full architected number of subchannels and channel paths

7. Channel subsystem monitoring, including device-connect-time measurement mode and measurement-block-update-mode.

8. Dynamic path reconnection on supported control units, such as IBM 3880 Model 3 with attached IBM 3380 Model AA4 or Model AD4 direct access storage

9. The 3082 processor controller.

10. The 3092 processor controller.

11. The 3090 Vector Facility.

12. Expanded storage.

An MTS Primer for VM/SP Users

A user of MTS typically accesses the system through an attached 3270-type or ASCII-type terminal. The user establishes a terminal session using the $SIGNON command which supplies the MTS userID and password (analogous to the VM LOGON command) and concludes the session with the $SIGNOFF command (analogous to the VM LOGOFF command).

Once signed on to MTS, the user can create, edit, and destroy disk files, compile and run programs, send mail and messages to other users, or direct printed output to the spooling system for eventual transcription to a real printer.

MTS provides extensive and economical facilities for creating and sharing files. Each MTS userID receives an allotment of disk space which represents the limit of that user's ability to consume DASD storage. Disk space is measured in 4096-byte pages which are similar to CMS file blocks. Unlike VM minidisks, the MTS disk allotment is not a pre-allocated commitment of certain cylinders on a disk; it is simply a limit representing the maximum number of disk pages that user may create.

As a file is created or expanded, new disk pages are automatically allocated to it by the MTS file system from the user's allotment. An attempt to create or expand a file beyond this limit is aborted with an error message. Physically, the disk volume is page formatted (by a system program similar to the CMS FORMAT program) and pages from all users are freely intermixed, according to the order in which they were allocated.

It is easy to share files in MTS, which is a major strength it has compared to CMS, and one that is critical in the university environment. Sharing occurs by designating an access type: read, write, destroy, or permit; and an accessor: a single MTS userID or collection of userIDs. For example, if userID ABCD wishes to grant read access to his file PHYLE to userID XXXX and write access to the same file to userID YYYY, he simply issues the command

```
  $PERMIT PHYLE READ XXXX, WRITE YYYY
```

This command takes effect immediately. There is no directory update process, as the file-sharing information is stored with the file. In addition to read and write permissions, destroy access allows the accessor to rename or destroy the file, while permit access grants the accessor essentially the same privileges over the file as the file's original owner; that is, the ability to issue the $PERMIT command against it to grant still further access.

Programs available under MTS include a wide variety of language translators for every conceivable programming language, text processors and formatters for document production, utilities to update and manage source and object code libraries, to list or copy files or compare two files and produce a list of differences, a sort program, and thousands of others.

Many IBM program products have been converted to run under MTS. They include MPSX/370, Assembler H, VS FORTRAN, FORTRAN H Extended, PL/I Optimizing Compiler, VS PASCAL, and GPSS.

The file editor bears some resemblence to XEDIT (which it predates), including a full-screen mode. The latter mode is available both on 3270 and ASCII terminals. Lines can be matched, altered, deleted, scanned for strings, selectively overlayed or blanked, moved, and copied. The editor incorporates the string-matching ability of the SNOBOL programming language which provides great flexibility in pattern matching and string searching commands.

Here is an annotated typical MTS terminal session to compose, debug, and run a program written in VS FORTRAN.

```
  $SIGNON ABCD                    (user signs on to MTS)
  password                        (password display is suppressed)
# User ABCD signed on at 12:34:56 Mon Feb 15/86.
# $CREATE FORTRAN_SRC             (create file for FORTRAN source)
# $EDIT FORTRAN_SRC               (invoke the file editor)
: VISUAL                          (enter full screen editing mode)
  ...                             (user types in source program)
: STOP                            (exit the editor)
# $RUN *FORTRANVS ...             (invoke VS FORTRAN compiler)
# Execution begins
  IFY020I ERROR IN LINE ...       (compiler error messages)
# Execution terminated
# $EDIT FORTRAN_SRC               (correct errors with editor)
: VISUAL
  ...
: STOP
# $RUN *FORTRANVS ...             (recompile corrected program)
# Execution begins
  NO ERRORS DETECTED              (no errors, object placed in
# Execution terminated             file -LOAD)
# $RUN -LOAD ...                  (user runs object program)
# Execution begins
  ...                             (results of program are displayed
                                   on the terminal)
# Execution terminated
# $SIGNOFF                        (user signs off terminal session)
```

The character at the extreme left is an MTS "prompt character". It tells the user what portion of the system he is communicating with. The character is "#" for MTS command mode, ":" for the file editor, and blank for a running program.