1.0 Introduction

This document describes the hardware configuration specification for the TYMBASes based on LSI-11 computers.

It is assumed that the reader is familiar with the architecture of the LSI-11 and that the appropriate DEC hardware manuals are at hand and available for reference.

2.0 Components

The following is a list of components which can be integrated into TYMBAS configurations.

2.1 Processors

Two processor boards can be used in a TYMBAS configuration, both are revisions of the KD11 module. These are the M7264 quad height board with on board memory and the M7270 dual height board, both mentioned in the 1978-1979 edition of "Microcomputer Processors".

2.2 Bootstraps

The bootstrap board used in a TYMBASe configuration is the BDV11-AA.

The BDV11 board is described in the 1978-1979 edition of "Memories and Peripherals".

The bootstrap has a prom which resides on the board. This bootstrap allows the down line loading of the TYMBAS from TYMNET using LOADII or equivalent program, or from a connected host. Hosts currently supported are:-

1. Pdp-20 Tymcom-X
2. Pdp-11 Rsx-11m
3. Pdp-11 Rsts
4. F3 Tenex

The BDV11 is a quad height board.

2.3 Memories

The TYMBASe is provided with either a 64K byte or 32K byte dual height MSV11-D or MSV11-E 18 bit MOS memory board.

The MSV11 memory is described in the 1978-1979 edition of "Memories and Peripherals" and both the MSV11 and the MXV11 are described in the 1979-1980 edition of "Microcomputer processor handbook".

2.4 Asynchronous Interfaces

Two serial asynchronous port options can be used in a TYMBAS. These are the DLV11-J and the DZV11.

The DLV11-J is a dual height module with 4 separate serial line units (SLU). A DLV11 SLU interfaces one asynchronous serial line I/O device (either 20ma or EIA) to the LSI-11 bus (without modem control). One DLV11 port must be configured as a console terminal (cty).

A DZV11 multiplexes asynchronous serial line I/O for 4 devices. It is a quad height module and has full modem control.

2.5 Synchronous Interfaces

Only one synchronous serial line interface is available for the TYMBAS configuration, the DUV-11. This is a quad height board described in the 1978-1979 edition of "Memories and Peripherals".

2.6 Backplanes

Two boxes can be used for the TYMBAS configuration, the BA11-N or large 9 slot, and the PDP-11/03 or small 4 slot box with an optional expansion box. These boxes are described in the 1979-1980 edition of "Microcomputer Interfaces Handbook".

For the 9 slot box, the slots serviced by the Q-bus, are the leftmost two (as viewed from the rear of the box). The rightmost two slots do not interface to the bus.

All slots in the 4 slot box (and the expansion box), are serviced by the Q-bus.

3.0 Configurations

There are two possible configurations that can be supported by the TYMBAS software. These are detailed below.

3.1 Large Box

The modules in the large box are inserted as follows;-

1. KD11 processor
2. MSV11 memory board
3. 0-2 DUV11 synchronous serial interfaces
4. 1-5 DLV11-J asynchronous serial interfaces
5. 0-4 DZV11 asynchronous multiplexors
6. 1-2 DRV11 parallel interfaces
7. BDV11 bootstrap board (with prom)

```
+-------------+-------------+
|    KD11     |             | 1.
+-------------+-------------+
|    MSV11    |             | 2.
+-------------+-------------+
|           DUV11           | 3.
+-------------+-------------+
|       DLV11-J/DUV11       | 4.
+---------------------------+
|     DLV11-J / DZV11       | 5.
+---------------------------+
|  DLV11-J / DZV11 / DRV11  | 6.
+---------------------------+
|  DLV11-J / DZV11 / DRV11  | 7.
+---------------------------+
|      BDV11 / DRV11        | 8.
+---------------------------+
|           BDV11           | 9.
+---------------------------+
```

The DUV11(s) must be installed in the bus closest to the processor because this is the most time critical device. The BDV11 acts as bus terminator, so it must always be inserted in the last slot after the other boards. (i.e. there must not exist any 'holes' in the bus).

3.2 Small box

The insertion of boards in the small box (with expander) is as follows:-

1. KD11 processor

2. MSV11 memory module

3. 1-2 DUV11 synchronous serial ports

4. 1-5 DLV11-J 4 SLU module

5. 0-4 DZV11 asynchronous multiplexors

6. 1-2 DRV11 parallel interface

7. 1 Q bus expander

```
+-------------------------------+
|      KD11      |    MSV11     | 1.
+----------------+--------------+
| (2 x DLV11-J)/ DZV11 / DUV11  | 2.
+-------------------------------+
| (2 x DLV11-J)/ DZV11 / DUV11  | 3.
+-------------------------------+
| ::::::::::::   |   DLV11-J    | 4.
+-::::::::::::------------------+
  ::Expander::
+-::::::::::::------------------+
| ::::::::::::   |   DLV11-J    | 5.
+-------------------------------+
|        DZV11 / DRV11          | 6.
+-------------------------------+
|    DZV11 / BDV11 / DRV11      | 7.
+-------------------------------+
|             BDV11             | 8.
+-------------------------------+
```

All boards must be contiguous from slot 1. The DUV11(s) must be installed in the slot(s) closest to the processor because it is the most time critical device. If no bus expander is provided, then the BDV11 must be inserted into slot 4, and the DUV11 in slot 2.

4.0 Configuring the Boards

This section describes the way in which the component boards must be configured to make them work in a TYMBAS.

It is assumed that the reader has nearby a copy of the relevant DEC reference handbook.

4.1 External Interfaces

The following is a list of 'standard' addresses and vectors that can be assigned to various interface boards in a TYMBAS configuration.

Device	Address	Vector
DRV11 1	167770	330
DRV11 2	167760	40
DRV11 3	167750	50
DRV11 4	167740	70
DUV11 1	160010	400
DUV11 2	160020	410
DUV11 3	160030	420
DUV11 4	160040	430
DZV11 1	160100	440
DZV11 2	160110	450
DZV11 3	160120	460
DZV11 4	160130	470
DLV11 1	176500	300
	176510	310
	176520	320
(cty)	177560	60
DLV11 2	176540	340
	176550	350
	176560	360
	176570	370
DLV11 3	176600	140
	176610	150
	176620	160
	176630	170
DLV11 4	176640	200
	176650	210
	176660	220
	176670	230
DLV11 5	176700	240
	176710	250
	176720	260
	176730	270

These addresses and vectors MUST be observed for consistency.

4.2 Bootstrap

1. Remove jumper W5

2. Install jumper W13

   (If W13 does not exist on your board, carefully install the jumper as indicated in the DEC instalation guide).

3. Install the EPROM in the socket indicated (49).

4. Configure both switches as indicated.

```
                                      Halt Run
 \:/                                     | /   /          \:/
-----------------------------------------|/---/---------------
| --- --- --- --- --- ---                [] []  0 0 0 0 0   |
| | | | | | | | | | | | |                S1  S2             |
| | | | | | | | | | | | |                        ---        |
| | | | | | | | | | | | |                        |E|        |
| --- --- --- --- --- ---                        |1|        |
|                                                |5|        |
| --- --- --- --- --- ---                        ---        |
| | | | | | | | | | | | |                                   |
| | | | | | | | | | | | |                                   |
| | | | | | | | | | | | |                                   |
| --- --- --- --- --- ---                                   |
|                    o....o<--- Install a Jumper here        |
| --- --- --- --- --- ---       (W13)                        |
| | | |D| |*| | | | | | |                                   |
| | | |E| |*|<----------------- Install EPROM here           |
| | | |C| |*| | | | | | |                                   |
| --- --- --- --- --- ---       ---                          |
|                               |E|                          |
| --- --- --- --- --- ---       |3|                          |
| | | | | |D| | | | | | |     o-|2|                          |
| | | | | |E| | | | | | | .-->: ---                          |
| | | | | |C| | | | | | | |   :         ---                  |
| --- --- --- --- --- --- | o-o         |E|                  |
|         Remove this     | | |         |2|                  |
|           Jumper (W5) --'     ---     |1|                  |
|                               |E|     ---                  |
|                               |3|                          |
|                               |1|                          |
|                               ---                          |
--           ----           --------         ----           --
  |         |    |         |        |       |    |         |
  |         |    |         |        |       |    |         |
   ---------      ---------          -------      ---------
```

E21 switches are set as follows:-

```
                 E21
          +-------------------+
          | 0 | 0 | 0 | 1 | 0 |
          +-------------------+
            1   2   3   4   5
```

On = 1, Off = 0

E15 switch settings are as follows:-

```
                    E15
          +-------------------------------+
          | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
          +-------------------------------+
            1   2   3   4   5   6   7   8
```

4.2 Synchronous Interface

The DUV11 is the only synchronous serial interface supported. The following defines the switch settings that must exist.

1. Option switches are set to allow single character synchronizing unless the connection is node to node (Remote nodes).

```
                 E55
+-------------------------------+
| 1 | 0 | 0 | x | 0 | 0 | 0 | 0 |
+-------------------------------+
  1   2   3   4   5   6   7   8
 x <= 1(on) if connection is to TYMNET else 0 (off).
```

2. Address and vector switches are set to the standard values.

```
                 E38
+-------------------------------+
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | x |
+-------------------------------+
  1   2   3   4   5   6   7   8
                 E39
+-------------------------------+
| x | x | 1 | 0 | 0 | 0 | x | x |
+-------------------------------+
  1   2   3   4   5   6   7   8
E38-8  => 1 for DUV11 number 4
else   => 0
E39-1  => 1 for DUV11 number 2 and 3
else   => 0
E39-2  => 1 for DUV11 number 1 and 3
else   => 0
E39-3  => 1 for all DUV11s
else   => 0
E39-7  => 1 for DUV11 number 3 and 4
else   => 0
E39-8  => 1 for DUV11 number 2 and 4
else   => 0
```

4.3 Parallel Interfaces

To set up the standard addresses and vectors on the DRV11, it is necessary to install and/or remove several wire straps.

4.3.1 Address Selection

The DRV11 address is encoded in 10 address straps A12 through A3, A12 being the most significant.

```
     A3 o    o      o    o A9
     A4 o    o      o    o A10
     A5 o    o      o    o A11
     A6 o    o      o----o A12
     A7 o    o
     A8 o    o
```

o A3 => install jumper for DRV11 2 and 4
o A4 => install jumper for DRV11 3 and 4
o A12 => install jumper for all DRV11s

4.3.2 Vector selection

Vector selection is accomplished by installing and/or removing straps between v7 through v3.

```
     V4 o    o     o---o V5


     V3 o    o     o   o V6
                   o   o V7
```

o V3 => install jumper for DRV11 2
o V4 => install jumper for DRV11 2 and 3
o V5 => install jumper for DRV11 1
o V6 => install jumper for DRV11 2, 3 and 4
o V7 => install jumper for DRV11 2, 3 and 4

4.4 Asynchronous Interfaces

4.4.1 DLV11-J

To set up the standard addresses and vectors for the 4 ports on the DLV11-J, it is necessary to jumper the board accordingly.

1. Address selection

   o A12 => X to 1

   o A11 => X to 1

   o A10 => X to 1

   o A9 => X to 0

   o A8 => X to 1

   o A7 => remove jumper for DLV11 1 and 2 else install jumper.

   o A6 => remove jumper for DLV11 3 and 4 else install jumper.

   o A5 => X to 0 for DLV11 1, 3 and 5 else X to 1.

2. Vector selection

   o V7 => Installed for DLV11 1, 2 , 4 and 5 else removed

   o V6 => Installed for DLV11 1 , 2 and 3 else removed

   o V5 => X to 0 for DLV11 1

   V5 => X to 1 for DLV11 2 , 3 and 5 else removed

3. Console selection

   o C1 => X to 1 for DLV11 1 else X to 0

   o C2 => X to 1 for DLV11 1 else X to 0

   o Break selection (B X H) => Remove jumper

4. Channel parameters

   For channels 0 through 3;

   o E => X to 0

- o D => X to 1
- o S => X to 0
- o P => X to 1

5. Speed selection

- o U = 150 Baud
- o T = 300 Baud
- o V = 600 Baud
- o W = 1200 Baud
- o Y = 2400 Baud
- o L = 4800 Baud
- o N = 9600 Baud
- o K = 19200 Baud
- o Z = 38400 Baud

Jumper from 0, 1, 2 or 3 to one of the above to select appropriate clock rate.

4.5.2 DZV11

Addresses and vectors for DZV11s are set up as follows:-

1. Address selection

```
                E30
+---------------------------------------+
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | x | x |
+---------------------------------------+
  1   2   3   4   5   6   7   8   9   10
E30-9  => 1 for DZV11 number 3 and 4
else   => 0
E30-10 => 1 for DZV11 number 2 and 4
else   => 0
```

2. Vector selection

```
               E2
+-------------------------------+
| 1 | 0 | 0 | x | x | x | 0 | 0 |
+-------------------------------+
  1   2   3   4   5   6   7   8
E2-4  => 1 for all DZV11s
else  => 0
E2-5  => 1 for DZV11 number 3 and 4
else  => 0
E2-6  => 1 for DZV11 number 2 and 4
else  => 0
```

MICRONODE TYMBASE

FAILURE RECOVERY PROCEDURES

March 1980
Journal Number 80272

Office Automation Division
Tymshare, Inc.
20705 Valley Green Drive
Cupertino, California 95014

1 INTRODUCTION

This document tells you how to bring up a micronode TYMBASE, also called an LSI11, that is down. The procedure documented here is very simple, but very important, for all information passed to and from the System XXV and the other AUGMENT hosts, as well as all interaction with individual users at their terminals, must go through the micronode. System XXV calls the micronode "TYMBASE" and uses a program called "TYMSRV" to connect to it. Thus, when you see TYMBASE in a system message, it means the micronode; the term TYMSRV means the System XXV program that communicates with the micronode.

There are two exceptions to the rule that the micronode must handle all communication with the outside world. The first exception is all communication with the system via the ARPANET. Communication via the ARPANET is handled by a different device and is completely independent of the micronode. If your System XXV is equiped to interact with the ARPANET as well as the micronode, you should be careful in bringing up the micronode. Do not do anything that might harm ARPANET users still working on the system. The second exception to the use of the micronode to communicate with the sustem is the System XXV's operator's terminal. Because operators must be able to use this terminal even when the micronode is down, it is wired directly to the System XXV and and does not go through the micronode. Thus this terminal and the system can continue to interact even when the micronode is not functioning correctly.

In some installations the System XXV and the micronode share one operator's terminal. In other installations, the System XXV and the micronode have separate terminals. As you work though the procedures in this document, you will notice steps labeled "[one-terminal installations]". These steps apply only to installations with one operator's terminal; they direct you to switch this terminal between connecting with the System XXV and the micronode. If your installation has separate terminals, you will not need to do this and should skip these steps. Other steps in the procedures will tell you to give a commmand on a particular terminal, either the system terminal or the micronode terminal. If your installation has separate terminals for the System XXV and the micronode, simply use the appropriate one. If your installation has only one terminal, do not worry; if you follow the steps for one-terminal installations your terminal will always be connected correctly.

2 HOW TO TELL IF THE MICRONODE IS DOWN

There is no one sure way to tell if the micronode is down, but the following symptoms mean it is not functioning properly and should be brought up with the procedure in section 3.

1) The System XXV operator's terminal repeatedly prints messages indicating problems with TYMBASE or TYMSRV. For example, "TYMBASE APPARENTLY DISABLED" or "TYMSRV: FAILED TO RESYNC WITH NODE".

2) You are getting irate calls from users who cannot reach the System XXV or have been detached.

3) The micronode's RUN light is off.

3 BRINGING UP THE MICRONODE

3.1 Introduction

Because the micronode connects users to the System XXV, before bringing it up and opening the system to users, you should always make sure the system is operating correctly. (See "System XXV Failure Recovery Procedures" for instructions on bringing the system up after a crash.) Once you know the system is in good shape, follow the procedure documented here to bring up the micronode. This procedure is not complicated; however, it does not always work the first time. If recovery fails, simply try again. If, after trying several times, you still do not succeed, see section 3.4 below, Errors and Recoveries.

Once the micronode is up, if your System XXV is connected to TYMNET, you need to make sure TYMNET and the micronode are communicating with each other and that TYMNET is OK. The second to last step in this procedure will allow you to do this. Skip it if it does not apply to you.

3.2 Summary

1) [one-terminal installations] Put the switch on the side of the operator's terminal to the position marked "Foonly" (usually up).

2) On the system terminal, type a few carriage returns. You should see the prompt "@" or "!". If you get "@", type "ena<CR>"; you should then get "!".

3) On the system terminal, type "<CTRL-E>tymnet<SP>off<CR>".

4) [one-terminal installations] Put the switch on the side of the operator's terminal to the position marked "micronode" (usually down).

5) Check the micronode's power light, the right light on the front. If the light is lit, the power is on. If the light is off, turn on the power switch on the back of the micronode.

6) Put off (down) the three switches on the front of the micronode.

7) Put on (up) LTC, the switch on the right.

8) Put on (up) HALT, the switch on the left.

9) Put on (up) INIT, the middle switch. The RUN light (the left light on the front of the micronode) should light.

10) Type a few carriage returns on the micronode terminal. When you see a "$", the prompt for the micronode's Command Processor, type "D0<CR>" (0 here is a zero).

11) [one-terminal installations] Put the switch on the side of the operator's terminal to the position marked "Foonly".

12) On the system terminal, type a few carriage returns. When you get the prompt "!", type "run<SP><system>nodebo<ESC><CR>".

13) When you see "MICRONODE IMAGE FILENAME >", type "<ESC><CR>".

14) The System XXV will report that it is trying to reset, load, and start the micronode. If this process is successful, you will get the message "NODE BOOTSTRAP COMPLETED SUCCESSFULLY" followed by some system messages.

15) When you get the "!" prompt on the system terminal, type "<CTRL-E>tymnet<SP>on<CR>". You may get some more system messages. After a few minutes, you should see your last message, "TYMBASE UP".

16) If you do not get "TYMBASE UP" or you get the message "MUST REPEAT ENTIRE REBOOT PROCEDURE", something has gone wrong. Turn the micronode's power off and back on and begin again with step 1.

17) [one-terminal installations] Put the switch on the side of the operator's terminal to the position marked "micronode".

18) Type a few carriage returns on the micronode terminal. If you then see "Please log in:", the micronode is OK. If you are NOT asked to log in, the micronode has come up wrong; start recovery again from step 1.

19) [for systems on TYMNET] When you see "Please log in:", wait a few minutes and then try to log in to another host on TYMNET. If you succeed, TYMNET is OK; log out from the other host. If you do not succeed, try again. If you have not succeded after five minutes, something may be wrong with TYMNET. Call the TYMNET Network Control Center and report the problem.

20) [one-terminal installations] Put the switch on the side of the operator's terminal to the position marked "Foonly".

3.3 Discussion

The micronode TYMBASE connects the System XXV (or Foonly) to the outside world, the users' terminals, networks, and other computers and funnels information to and from the System XXV, allowing users and other hosts to interact with the system. To do this, the micronode runs a program stored in its memory. When the micronode goes down, this program is lost and must be reloaded from the System XXV where it is permanently stored. The micronode recovery procedure first prepares the micronode to receive this program and then instructs the System XXV to send it over. Once the program has been successfully transmitted and the micronode is running it, the System XXV must resynchronize with the micronode so they can pass information back and forth. If the System XXV succeeds in synchronizing with the micronode, then the micronode has probably come up correctly.

You begin a micronode recovery by making sure that you are at the AUGUST EXEC and that you are enabled. Type a few carriage returns on the system terminal. (At installations with one terminal for the System XXV and the micronode, first connect the terminal to the System XXV by putting the switch on the side of the terminal to the position marked "Foonly" and then type the carriage returns). If, when you type the carriage returns, you get the prompt "!", you are enabled. If your prompt is "@", you are not enabled. Since you must be enabled to bring up the micronode, if you get the "@" prompt, enable yourself by typing "ena<CR>" at the EXEC. When you enable, you tell the system that you are a person with special powers who should be allowed to do things normal users cannot do. After you enable, you will be prompted with "!".

You now prepare the System XXV and the micronode for recovery. Since you do not want the System XXV to use the micronode as you bring it up, you begin by typing "<CTRL-E>tymnet<SP>off<CR>" on the system terminal. This tells the System XXV that the micronode is no longer available. Next, if you are at an installation with one operator's terminal, connect the terminal to the micronode instead of the System XXV by putting the switch on the operator's terminal to the position marked "micronode" (usually down). Now you need to check the micronode itself to make sure it has power. The right light on the front of the micronode is lit when the power is on. If this light is off, turn on the power switch at the back of the micronode. Finally, to reset the micronode before bringing it up, put off (down) the three switches on the front of the micronode.

You now are ready to begin bringing up the micronode. You do this by putting on (up) the three switches on the front of the micronode. Make sure that you put on these switches in the proper order; putting them on in the wrong order, can cause serious damage. First, put on the right switch on the micronode labeled "LTC". This switch turns on the line clock timer, an internal timer that makes sure computer procedures happen at the correct intervals and are synchronized. Second, put on the left switch on the micronode labeled "HALT". Third, put on the middle switch on the micronode labeled "INIT".

When you put on the "INIT" switch, the RUN light (the right light on the micronode) should come on. The micronode is now running a small Command Processor. Type a few carriage returns on the micronode terminal; you should get a "$", the Command Processor's prompt. If you get an "@" prompt instead of the "$", you may have accidentally entered the micronode's debugger. If you are at a one-terminal installation, check to make sure the switch on the terminal is pushed to the position marked "micronode". If the switch accidentally has been left at the position marked "Foonly", then the "@" you are getting is the prompt for the AUGUST EXEC. Push the switch to the position marked "micronode", and again try typing a few carriage returns; you should now get the "$" prompt. If the switch was already set correctly when you got the "@" prompt or you are at an installation with separate terminals, you have a serious problem. Immediately start the recovery procedure again with step 6.

Once you get the "$" prompt, the micronode is ready and waiting to be brought up. On the micronode terminal, type "D0<CR>". Note that "0" is zero, and that all commands to the micronode must be capitalized. The D0 command tells the micronode that it should prepare to take a new program from the System XXV. To tell the System XXV to deliver the program, type "run<SP><system>nodebo<CR>" on the system terminal. (At one-terminal installations, put the switch on the side of the terminal to the position marked "Foonly" before typing "run<SP><system>nodebo<CR>".) When you give this command the system will begin to run a micronode rebooting program which will ask "MICRONODE IMAGE FILENAME >". An image file is a file that can be read directly into memory. In this case, it is like a snapshot of the memory as it should be for the micronode to operate correctly. To tell the program what image file to use, type "<ESC>". <ESC> here means "use the standard file"; the program will fill out the name of this file followed by the comment "[Old Version]". Type a carriage return to confirm this.

The System XXV will now deliver the micronode image file to the micronode and bring it up. As it does this, the system will print various messages about its progress, such as, "resetting the micronode", "loading the micronode", or "starting the micronode". If and when this process is finished, you will be told "NODE BOOTSTRAP COMPLETED SUCCESSFULLY".

Once System XXV has successfully finished bringing up the micronode can communicate with it, on the system terminal type "<CTRL-E>tymnet<SP>on<CR>". This tells the System XXV to try to synchronize with the micronode and attempt to send and receive information. If it sucedes, you will see some more system messages and, after a while, "TYMBASE UP". If, after waiting several minutes, you do not get this message or, if you get the message "MUST REPEAT ENTIRE REBOOT PROCEDURE", something was wrong with the micronode's recovery. Either the image file was not delivered sucessfully or the System XXV could not resynchronize with the micronode. In any case, you must restart this procedure at step 1.

Once the micronode does come up and you see the message "TYMBASE UP", you still need to make sure the micronode really is functioning correctly. To do this, you use the micronode terminal as if you were a regular user trying to log in to the System XXV. (A one-terminal installations, first connect the terminal to the micronode by putting the switch on the side of the terminal to the position marked "micronode".) When you type a few carriage returns on the micronode terminal, the micronode should notice the existence of the terminal, treat it just like any other terminal, assume a user is waiting to log in, and say "please log in:". If this happens, the micronode has come up correctly. If you are not asked to log in, there is still something wrong. Begin the recovery again at step 1.

If the micronode does ask you to log in, you know that users whose terminals are connected to the micronode can use it to reach the System XXV. If your System XXV is connected to TYMNET, you now should make sure that this micronode can communicate with the rest of the TYMNET network. After waiting three minutes (to give TYMNET time to notice that the micronode is back on the network), try to log in to another host on TYMNET. If, after trying for five minutes, you cannot reach another TYMNET host, call the TYMNET Network Control Center and report the problem to them. If you can log in to another TYMNET host, you know that the micronode is fine, that the connection with TYMNET is good, and that information is successfully passing between the micronode and the network. Log out from the other TYMNET host and, if you are at an one-terminal installation, push the switch on the terminal back to the position marked "Foonly". Once you have done this, you have completed the procedure of bringing up the micronode and testing it.

3.4 Errors and Recoveries

"Must Repeat Entire Reboot Procedure"

If you get this error message, repeat the entire procedure.

Repeated Failure to Bring up the Micronode

Introduction

If, if after trying several times, you cannot bring up the micronode, this may mean that the micronode itself has a memory problem. To determine if this is the case, you run a short program that checks the memory of the micronode.

Summary

1) Repeat steps 1 though 9 of the recovery procedure.

2) At step 10, where you would normally type "DO<CR>", instead type "XM<CR>". Note that all commands to the micronode must be capitalized.

3) [one-terminal installations] Put the switch on the side of the operator's terminal to the position marked "Foonly".

4) The micronode will now examine its memory. If it prints a series of numbers on the micronode terminal, there is something wrong with the memory. Call Tymshare Maintenance.

5) If nothing is printed, the memory is good but you may still have a hardware problem. Try the entire recovery procedure from step 1 twice more. If you do not succeed, notify Tymshare Maintenance.

MICRO-NODE NETWORK MANAGER'S GUIDE

January 23, 1981

TYMSHARE Distributed Systems Group
Cupertino, Ca.

Table of Contents

1 Introduction

This manual provides the specified information needed by the manager of a MICRO-NODE network. Topics covered include general rules for configuring a network and detailed information on specifying software and hardware configurations of network nodes.

2 Node configuration

The configuration of each node in a MICRO-NODE network must be specified in a configuration file for that node. In addition, the global configuration of the entire network must be specified in a map file for the network. Also, a link command file must be produced for each node. To generate the program for each node, the module COMMON.MAC is assembled with the configuration file and the module ROUTE.MAC is assembled with the configuration and map files.

The resulting object files are linked along with the pre-assembled node object files which are independent of the configuration using the linker program to produce a "image" file which can be loaded from a host into the node using the NODLOD program. The image file can be input to the program NIBTRN to produce a "NIB" format file for loading over TYMNET using LOADII.

A GENBAS program is planned to generate the required configuration and command files for building a network node. Until this program is available, the configuration and command files must be edited manually to indicate the desired configuration.

Following is a complete typical node configuration file:

```
*******************************************************************
*                                                                 *
*       .SBTTL  TEST   - NEXILIS/JRG                              *
*       .SBTTL  TEST   - CONFIGURATION FILE FOR NODE NEX01        *
*                                                                 *
*       .MACRO  CONFIG                                            *
*                                                                 *
*;DEFINE GENERAL PARAMETERS                                       *
*                                                                 *
*       DEFINE  JIFSEC,60.      ;60 TICKS PER SECOND              *
*                                                                 *
*;DEFINE THE NAME OF THIS NODE                                    *
*                                                                 *
*       NAME    NEX01                                             *
*                                                                 *
*;DEFINE LINKS BETWEEN THIS NODE AND OTHER NODES AND HOSTS        *
*                                                                 *
*       LINK    NEXA  ,000,TF,10.,S,D,<170000,120,5>              *
*       LINK    NEX02 ,000,TF,10.,P,C,<170010,130,5>              *
*                                                                 *
*;DEFINE ALL DEVICES ON THIS NODE                                 *
*                                                                 *
*       DEV     ST,200,<300,377>                                  *
*         SOC   1.                                                *
*         SOC   2.                                                *
*         SOC   3.                                                *
*                                                                 *
*       DEV     LP                                                *
*         SOC   1.,A,<177554,204,177550>                          *
```

```
*                                                                 *
*;DEFINE THE HOST CONNECT LISTS                                   *
*                                                                 *
*       HCL     A,<NEXA>                                          *
*                                                                 *
*;DEFINE THE DEFAULT HOST                                         *
*                                                                 *
*       DHOST   NEXA                                              *
*                                                                 *
*;DEFINE TERMINAL LINES                                           *
*                                                                 *
*       LINX    A,<177560,60>                                     *
*         LINE  1.,F,300, ,NEX,CRT,ASC,RSTS,N,N,Y,Y,HW,A,200      *
*                                                                 *
*       LINX    B,<160130,340>                                    *
*         LINE  2.,A,  3, ,NEX,CRT,ASC,RSTS,N,N,Y,Y,HW,A,200      *
*         LINE  3.,A,  3, ,NEX,CRT,ASC,RSTS,N,N,Y,Y,HW,A,200      *
*         LINE  4.,A,  3, ,NEX,CRT,ASC,RSTS,N,N,Y,Y,HW,A,200      *
*         LINE  5.,A,  3, ,NEX,CRT,ASC,RSTS,N,N,Y,Y,HW,A,200      *
*                                                                 *
*       .ENDM                                                     *
*                                                                 *
*******************************************************************
```

Following is a description of each macro used in the configuration file:

```
**********
* DEFINE *
**********
```

The DEFINE macro is used to assign non-standard values to node parameters. It normally does not need to be used. Its format is:

```
        DEFINE  parameter,value

Where:  parameter = name of node parameter to be defined
        value     = value for parameter
```

```
**********
* NAME   *
**********
```

The NAME macro is used to specify the name of the node. It must be included in the configuration file.

```
        NAME    node
```

```
Where:  node   = name of this node
```

```
**********
* TYMSAT *
**********
```

This macro specifies that this node is a Tymsat only node. It has no parameters.

```
**********
* LINK   *
**********
```

The LINK macro is used to specify links between this node and other nodes or hosts or gateways.

```
        LINK    name,level,auxcap,conn,prisec,type,<xxxxxx>

Where:  name   = name of node, host, or gateway on other end of link
        level  = minimum priviledge which allowes connection if host or
                 gateway; maximum priviledge allowed across link if link
                 to another node
        auxcap = sets link capabilities as follows:
                   TF = connections to and from this link are allowed
                   T  = connections allowed only to this link
                   F  = connections allowed only from this link
                   N  = no connections allowed to or from this link
        connno = number of connection sockets allocated for this link
        prisec = indicates if this end of the link is primary or
                 secondary (in terms of socket allocation)
        type   = indicates type of link as follows:
                   A = not used
                   B = NSP/a (synchronous line protocol) (one end of
                       each link must be primary; the other end must be
                       secondary)
                   C = DRV11/DR11-C parallel interface link (one end of
                       each link must be primary; the other end must be
                       secondary)
                   D = DRV11/DR11-C RSX or RSTS host interface (must
                       always be secondary)
                   E = X.25 gateway (byte framing) (primary = DCE,
                       secondary = DTE)
                   F = Tymcom-X (2020 and F3) and TENEX host interface
                       (must always be secondary)
                   G = Tymnet extended X.25 gateway (primary = DCE,
                       secondary = DTE)
                   H = Tymnet native mode (T201) gateway (must always be
                       secondary)
        xxxxxx = specifies link type dependent parameters.

        For types B, C, and D, the format is:
```

```
          <reg,vec,level>

where:  reg   = address of first device register
        vec   = address of first device vector
        level = interrupt priority level for
                interface
```

For link types E and G the format is:

```
          <reg,vec,level,dst,adst,window,T1,N1,N2>

where:  reg    = address of first device register
        vec    = address of first device vector
        level  = interrupt priority level for
                 interface
        dst    = destination name for incoming
                 connections
        adst   = alternate destination name for
                 incoming connections
        window = frame level window size
        T1     = no response timer value (ticks)
        N1     = maximum number of bytes in frame
                 information field
        N2     = maximum number of retransmissions
```

Format for type F link is:

```
          <reg,vec,level,rngsiz,KMC>

where:  reg    = address of first device register
        vec    = address of first device vector
        level  = interrupt priority level for
                 interface
        rngsiz = size of host input ring (in PDP-10
                 words)
        KMC      indicates that a KMC is used in
                 the 2020 interface
```

Format for type H link is:

```
          <nodnum,nsph,kind>

where:  nodnum = node number as seen by Tymnet
        nsph   = number of ports available for each
                 host - specifies how many ports
                 are really available to each host
                 out of the maximum number defined
                 (256 if 1 host, 128 if 2 hosts, or
                 64 if 3 or 4 hosts).  Ports not
                 available to hosts are available
                 for circuits originating from
                 terminals.  If all ports are
                 specified as available to hosts,
                 no terminal circuits can be built!
```

kind = kind of link - must be one of the following:

SAT = TYMsat

BAS = TYMbase without terminal protocol conversion

TPC = TYMbase with terminal protocol conversion

```
**********
* LNKH   *
**********
```

The LNKH macro is used follow a LINK macro for a type H gateway to specify the "hosts" which are to be known to Tymnet. If this node is not to support any hosts, i.e., if it is to be a "TYMSAT" only node, this macro should not appear. In this case the LKHST modules must be loaded. If it does appear, this node will be a "TYMBAS" and the LKHBS modules must be loaded. The format is:

```
        LNKH     TYMHST,NEXHST
```

Where: TYMHST = Tymnet host number

NEXHST = MICRO-NODE host name which corresponds to TYMHST

There may be up to 4 hosts specified for each type H link.

```
**********
* LNKL   *
**********
```

The LNKL macro is only used following a LINK macro for a type H gateway. It specifies a synchronous line connecting to Tymnet. The format is:

```
        LNKL     chn,nbr,speed,sat,type,xxxxxx
```

Where: chn = Number of channels on line (number of groups times 16)
nbr = number of neighbor node
speed = speed of line as follows:

1 = 2400 baud
2 = 4800 baud
3 = 56000 baud
4 = 9600 baud

```
sat    = S if  this is  a satellite  link with  long propigation
           delay, N if it is a normal link
type   = line driver type - currently must always be A
xxxxxx = line driver type dependent parameters as follows:
                    <reg,vec,level>
           where:   reg    = address of first device register
                    vec    = address of first device vector
                    level  = interrupt    priority    level    for
                             interface
```

A type H link must always specify at least 1 Tymnet line using a LNKL macro. Up to 8 lines may be specified. All LNKL macros must follow all LNKH macros.

```
**********
* DEV    *
**********
```

The DEV macro specifies all non-terminal devices on the node. It is only needed if devices are present. Its format is:

```
        DEV     dv,level,<xxxxxx>

where:  dv     = two letter mnemonic for device
        level  = minimum priviledge level for connection to device
        xxxxxx = device dependent parameters
```

Valid devices are:

ST — Status device - this is a software device which is used to obtain status information from the node and to control the node. The status device is optional but should usually be included to allow access to data stored in the node. The two device dependent parameters are the priviledge level for examining and writing memory.

WD — Load device - this is a special device which is used to down-line load adjacent nodes. It only be used with type B or C links. When included, it must always be specified with one socket.

LP — Line printer

```
**********
* SOC    *
**********
```

The SOC macro is used with the DEV macro to specify devices. Each DEV macro must be followed by one or more SOC macros which specify each socket on the

device as follows:

```
SOC     sn,type,<xxxxxx>
```

Where: sn = socket number (must start at 1 and increase by 1)

type = type of device driver for this socket (not used for ST or WD)

xxxxxx = device driver dependent parameters (not used for ST or WD - LP has unknown device dependent parameters)

```
**********
* HCL    *
**********
```

The HCL macro specifies any "host connect lists" which are to be used by terminals on this node. Each host connect list specifies an ordered list of host systems which is used when a terminal is initially connected to the node or when a "connect" command is issued without an argument. Format of this macro is:

```
HCL     list,<host1,host2,...>
```

Where: list = a single letter which identifies this list. First list must be "A", second "B", etc.

host1 = name of first host system in list

host2 = name of second host system in list

```
**********
* DHOST  *
**********
```

The DHOST macro specifies a single "default host" which will be used as the destination whenever an unknown host is specified. The host specified will be used as the destination extension. This allows hosts connected to a foreign network connected to this network through a gateway to be accessed in a transparent manner. The gateway is specified as the default host. Any requests for an unknown host will thus be routed to the gateway, which will then use the destination extension as the destination on the foreign network. Format of this macro is:

```
        DHOST       hstnam

where:   hstnam = name of default host (or gateway)
```

```
**********
* LINX   *
**********
```

The LINX macro specifies the asynchronous terminal interfaces on the node. Format is:

```
        LINX     type,<xxxxxx>

where:  type   = type of interface as follows:
                   A = DLV11/DL11 single line interface
                   B = DZV11/DZ11 4 or 8 line interface
                   C = DH11 16 line interface
                   D = Infotron 780 "Super-mux" driver
                   F = Nexilis Systems DCP board
        xxxxxx = interface dependent parameters
                 (usually <address,vector,priority>)
```

```
**********
* LINE   *
**********
```

Each line on each terminal interface must be specified in a LINE macro. The LINE macros must immediately follow the LINX macro which specifies the interface. Its format is:

```
LINE num,fa,or,ir,cmd,type,code,pro,ask,snm,cm,acd,rob,dunw,hcl,priv

where:     num  = socket number for port (must be in ascending order)
           fa   = indicates if fixed or auto-baud rate or if special
                  line; F = fixed, A = auto, C = special line used as
                  high speed clock, L = special line used as serial
                  printer port. If C or F, no parameters beyond this one
                  are used. All C or L lines must follow all A or F
                  lines (even if on different interfaces). F and A lines
                  may be freely intermixed as may C or L lines.
           or   = output baud rate if fa = F or auto rate detect type if
                  fa = A
           ir   = input baud rate if fa = F and it is different from the
                  output rate - otherwise should be null
           cmd  = name of network command decoder - should be "TYM" for a
                  TYMNET base and "SAT" for a TYMSAT
           type = default terminal type (usually "TYMSHR")
           code = default character code for terminal - valid values are
```

ASC (Ascii), CRS (selectric correspondence), CAL (Selectric Call-360), BCD (Selectric BCD), and EBC (Selectric EBCD)

pro = default terminal profile (based on host; TYMSHR, RSTS, RSX)

ask = Y if should ask for Tymshare terminal ID; N if should not

snm = Y if should suppress all network messages, I if should suppress only informational network nessages, N if should output all network messages

cm = Y if should allow network command mode and command mode should be default when not connected; R if should allow network command mode but only as a result of typing the command interrupt character; N if network command mode is not allowed

acd = Y if should do auto-connect when a disconnect is received; N if should not (not used with TYMNET version)

rob = Y if should release (hang up) the terminal when a break is detected, N if should not

duhw = indicates if dial-up(answers ring indicator and watches carrier detect) or hard-wired line; DU = dial-up, HW = hard-wired

hcl = host connect list - single letter which must have been used as a host connect list name in a HCL macro

priv = network privilege for terminal (a octal number in the range 0-377)

3 Routing configuration

For all configurations of MICRO-NODES some routing information is necessary. For an isolated node the routing is simple and straightforward. As the the number of nodes in a network increases, the routing become more complicated. The routing information is contained in the ROUTE macro. Generally the ROUTE macro is placed in a file of the form RUxxxx.MAC, where xxxx is the node number. The ROUTE macro has three parts: node identification information, local resource information, and connectivity information for all other resources in the network.

An example of a routing macro is:

```
; ru552.mac - network map for node 552 (and friends)
.macro ROUTE
        node    552
            NPATH   554,554
            NPATH   553,553
           hpath    118,554,553
            HPATH   930,*
           hpath    TYMNET,*
            HPATH   BNET,554
        NODE    553
            NPATH   552,552
           NPATH    554,552
            HPATH   118,*,552
            HPATH   TYMNET,552
           HPATH    930,552
            HPATH   BNET,552
        NODE    554
           NPATH    553,552
            NPATH   552,552
           HPATH    930,552
            HPATH   118,*,552
            HPATH   TYMNET,552
            HPATH   BNET,*
.endm
```

The NODE statement specifies which node the information that follows is associated with. Information which follows a NODE statement for a node other that the one being assembled is ignored. Thus a NODE statement is REQUIRED in order for routing information to be assembled for a node.

The first NPATH parameter specifies the primary path for reaching other nodes in the network. The second parameter specifies a secondary path to be used if the first is down.

The HPATH statement specifies how to reach resources (hosts and gateways) in the network. The "*" indicates that the resource is in the node being assembled. It specifies a primary and a secondary path line the NPATH statement.

4 Node modules

The following table summarizes the modules required for various node configuration options. Each module should only be loaded once and all modules may be loaded in any order after the COMMON, ROUTE, and XDT modules except that END must always be the last module loaded.

The base for the loaded image should normally be 1000. This is the default for the PDP-11 LINK11 linker. It must be specified for the PDP-10 cross-linker. If the node uses an MXV-11 multi-function card with the ROM area configured for bank 0, the base address must be 21000. This must be specified for both the PDP-11 linker and the PDP-10 cross-linker.

The following modules must always be loaded:

CGxxxx,RUxxxx,EXEC,ERROR,PKT1,PKT2,PKT3D,ONCE,END

If the node includes a Tymshare "watch-dog" timer, the DOG module must be included unless the AUTO7 module is included.
This table shows modules required for each type of link.

Type	Notes	Modules required
B	using DUP11	LNK1,LNK2,LNKFLG,LNKB,LNKBS,LKBDUP
B	using DUV11	LNK1,LNK2,LNKFLG,LNKB,LNKBS,LKBDUV,CRCBYT
B	using DCP	LNK1,LNK2,LNKFLG,LNKB,LNKBS,LKBDCP
C		LNK1,LNK2,LNKRSP,LNKC
D		LNK1,LNK2,LNKRSP,LNKD
E		LNK1,LNK2,LNKE,LKEPKT,X25Q,CRCTAB
F	using KMC	LNK1,LNK2,LNKF,LNKF2,TYMA,TYMB
F	without KMC	LNK1,LNK2,LNKF,LNKF1,TYMA,TYMB
G		LNK1,LNK2,LNKG,LKGPKT,LKGQAL,CRCTAB
H	using DU11/DUV11 (Tymsat only)	LNK1,LNKHA,LKHST1,LKHST2,LKHST3,TYMA (see note 1)
H	using DUP11 (Tymsat only)	LNK1,LNKHAP,LKHST1,LKHST2,LKHST3,TYMA (see note 1)
H	using DU11/DUV11 (Tymsat or Tymbas)	LNK1,LNKHA,LKHBS1,LKHBS2,LKHBS3,TYMA,TYMB
H	using DUP11 (Tymsat or Tymbas)	LNK1,LNKHAP,LKHBS1,LKHBS2,LKHBS3,TYMA, TYMB
H	using DU11/DUV11 (Tymbas with TPC)	LNK1,LNKHA,LKHTP1,LKHTP2,LKHTP3,LKHTPC, TYMA,TYMB (see note 2)
H	using DUP11 (Tymbas with TPC)	LNK1,LNKHAP,LKHTP1,LKHTP2,LKHTP3,LKHTPC, TYMA,TYMB (see note 2)

note 1: For a Tymsat only node, the following standard modules must be assembled with the parameter file SAT.MAC in addition to XP.MAC: TRM, TRMF, TRMT, TRMP, CMDS.

note 2: For a node with Tymnet terminal protocol conversion, the module TRMP must be assembled with the parameter file TPC.MAC in addition to XP.MAC.

The following table specifies which modules are required for devices:

Device	Type	Modules required
WD		LOAD
ST		STATUS
LP	A	LPTA

If any terminal lines are specified for the node, the following modules must be included:

TRM,TRMF,TRMP,TRMT,CMDS

If any terminal lines specify the "TYM" command decoder, the CMDTYM module must be included.

If any teminal lines specify the "SAT" command decoder, the CMDSAT module must be included.

If any terminal specifies auto-baud, the modules corresponding to the auto-baud detect type specified must be included:

Detect type	Module	Description
1	AUTO13	110, 150, 300 baud ascii and 134.5 baud correspondence
2	AUTO13	110, 300, 1200 baud ascii and 134.5 baud correspondence code
3	AUTO13	300, 600, 1200, 1800, 2400, 4800 and 9600 baud ascii
4	AUTO45	TYMNET style detect for 110, 150 and 300 baud ascii terminals
5	AUTO45	TYMNET style detect for 110, 300 and 1200 baud ascii
6	AUTO6	TYMNET style detect for 110, 300 and 1200 baud ascii and 134.5 baud correspondence or ebcdic (using high speed clock)
7	AUTO7	TYMNET style detect same as type 6 (but with DOG builtin)

The AUTO6 and AUTO7 modules cannot both be loaded in the same node. The AUTO7 module includes Tymshare watchdog timer functions, so if it is loaded, the module DOG must not be loaded.

If any terminal codes other than Ascii are specified, the modules specified in the following table must be included:

Code Module

CRS	XLTCRS
EBC	XLTEBC
CAL	XLTCAL
BCD	XLTBCD.

Note that some auto-baud detect tables specify codes other than ASCII. Use of any of these auto-baud tables also require that the corresponding translation tables be loaded.

The following table specifies which modules are required for terminal interfaces:

Type	Notes	Modules required
A		LINA
B	no C or L lines	LINB
B	C or L lines	LINBX
C		LINC
D		LIND,LIND1
F		LINF

5 Module assembly

The configuration dependent modules are assembled using MACRO-11 or MACN11 as follows: (node name is assumed to be "xxxx")

```
CGxxxx=XP,CGxxxx,COMMON

RUxxxx=XP,CGxxxx,RUxxxx,ROUTE
```

If sources are available, the remaining modules are assembled as follows:

```
For machines with EIS           For machines without EIS

EXEC=XP,EXEC                    EXEC=NOEIPR,XP,EXEC
ERROR=XP,ERROR                  ERROR=NOEIPR,XP,ERROR
PKT1=XP,PKT1                    PKT1=NOEIPR,XP,PKT1
PKT2=XP,PKT2                    PKT2=NOEIPR,XP,PKT2
DOG=XP,DOG                      DOG=NOEIPR,XP,DOG
STATUS=XP,STATUS                STATUS=NOEIPR,XP,STATUS
TRM=XP,TRM*                     TRM=NOEIPR,XP,TRM*
TRMF=XP,TRMF*                   TRMF=NOEIPR,XP,TRMF*
TRMP=XP,TRMP**                  TRMP=NOEIPR,XP,TRMP**
TRMT=XP,TRMT*                   TRMT=XP,TRMT*
CMDS=XP,CMDS*                   CMDS=NOEIPR,XP,CMDS*
CMDNEX=XP,CMDNEX                CMDNEX=NOEIPR,XP,CMDNEX
CMDTYM=XP,CMDTYM                CMDTYM=NOEIPR,XP,CMDTYM
AUTO13=XP,AUTO13                AUTO13=NOEIPR,XP,AUTO13
AUTO45=XP,AUTO45                AUTO45=NOEIPR,XP,AUTO45
AUTO6=XP,AUTO6                  AUTO6=XP,AUTO6
AUTO7=XP,AUTO7                  AUTO7=XP,AUTO7
XLTCRS=XP,XLTCRS                XLTCRS=XP,XLTCRS
XLTEBC=XP,XLTEBC                XLTEBC=XP,XLTEBC
XLTBCD=XP,XLTBCD                XLTBCD=XP,XLTBCD
XLTCAL=XP,XLTCAL                XLTCAL=XP,XLTCAL
LINA=XP,LINA                    LINA=NOEIPR,XP,LINA
LINB=XP,LINB                    LINB=NOEIPR,XP,LINB
LINBX=XP,LINBX,LINB             LINBX=NOEIPR,XP,LINBX,LINB
LINC=XP,LINC                    LINC=NOEIPR,XP,LINC
LIND=XP,XM,LIND                 LIND=NOEIPR,XP,XM,LIND
LIND1=XP,XM,LIND1               LIND1=NOEIPR,XP,XM,LIND1
LINF=XP,LINF                    LINF=NOEIPR,XP,LINF
LNK1=XP,LNK1                    LNK1=NOEIPR,XP,LNK1
LNK2=XP,LNK2                    LNK2=XP,LNK2
LNKB=XP,LNKBPR,LNKB             LNKB=NOEIPR,XP,LNKBPR,LNKB
LNKBS=XP,LNKBPR,LNKBPS          LNKBS=NOEIPR,XP,LNKBPR,LNKBPS
LKBDUP=XP,LNKBPR,LKBDUP         LKBDUP=NOEIPR,XP,LNKBPR,LKBDUP
LKBDUV=XP,LNKBPR,LKBDUV         LKBDUV=NOEIPR,XP,LNKBPR,LKBDUV
CRCBYT=XP,CRCBYT                CRCBYT=XP,CRCBYT
LNKC=XP,LNKC                    LNKC=NOEIPR,XP,LNKC
LNKD=XP,LNKD                    LNKD=NOEIPR,XP,LNKD
LNKE=XP,LNKE                    LNKE=NOEIPR,XP,LNKE
LKEPKT=XP,LKEPKT                LKEPKT=NOEIPR,XP,LKEPKT
X25Q=XP,X25Q                    X25Q=NOEIPR,XP,X25Q
LNKF=XP,LNKFPR,LNKF             LNKF=NOEIPR,XP,LNKFPR,LNKF
```

```
LNKF1=XP,LNKFPR,LNKF1               LNKF1=NOEIPR,XP,LNKFPR,LNKF1
LNKF2=XP,LNKFPR,LNKF2               LNKF2=NOEIPR,XP,LNKFPR,LNKF2
LNKG=XP,LNKG                        LNKG=NOEIPR,XP,LNKG
LKGPKT=XP,LKGPKT                    LKGPKT=NOEIPR,XP,LKGPKT
LKGQAL=XP,LKGQAL                    LKGQAL=NOEIPR,XP,LKGQAL
CRCTAB=XP,CRCTAB                    CRCTAB=XP,CRCTAB
LNKHA=XP,LNKHA                      LNKHA=NOEIPR,XP,LNKHA
LNKHAP=XP,DUPPR,LNKHA               LNKHAP=NOEIPR,XP,DUP,LNKHA
LKHST1=XP,SATPR,LNKHPR,LKHPK1       LKHST1=NOEIPR,XP,SATPR,LNKHPR,LKHPK1
LKHST2=XP,SATPR,LNKHPR,LKHPK2       LKHST2=NOEIPR,XP,SATPR,LNKHPR,LKHPK2
LKHST3=XP,SATPR,LNKHPR,LKHPK3       LKHST3=NOEIPR,XP,SATPR,LNKHPR,LKHPK3
LKHBS1=XP,LNKHPR,LKHPK1             LKHBS1=NOEIPR,XP,LNKHPR,LKHPK1
LKHBS2=XP,LNKHPR,LKHPK2             LKHBS2=NOEIPR,XP,LNKHPR,LKHPK2
LKHBS3=XP,LNKHPR,LKHPK3             LKHBS3=NOEIPR,XP,LNKHPR,LKHPK3
LKHTP1=XP,TPC,LNKHPR,LKHPK1         LKHTP1=NOEIPR,XP,TPC,LNKHPR,LKHPK1
LKHTP2=XP,TPC,LNKHPR,LKHPK2         LKHTP2=NOEIPR,XP,TPC,LNKHPR,LKHPK2
LKHTP3=XP,TPC,LNKHPR,LKHPK3         LKHTP3=NOEIPR,XP,TPC,LNKHPR,LKHPK3
LKHTPC=XP,LNKHPR,LKHTPC             LKHTPC=NOEIPR,XP,LNKHPR,LKHTPC
LOAD=XP,LOAD                        LOAD=NOEIPR,XP,LOAD
ONCE=XP,ONCE                        ONCE=NOEIPR,XP,ONCE
END=XP,END                          END=XP,END
```

Notes:

* If for Tymsat only node, these files must be assembled with the parameter file SAT in addition to XP.

** If for Tymsat only node, this file must be assembled with the parameter file SAT in addition to XP; if for node with Tymnet terminal protocol converter (LKHTPC loaded) this file must be assembled with the parameter file TPC in addition to XP.

6 RSTS host configuration

In order to use a RSTS system as a host in a MICRO-NODE network, several network related parameters must be defined at SYSGEN time. These parameters specify how the system is connected to the network and how K.B. numbers are mapped to network terminals or devices. The parameters associated with the connection to the network specify how many nodes are connected to the host and give the device addresses for each interface to a node. Note that, unlike other devices on a RSTS system, the device addresses for the network interfaces must be specified at SYSGEN time. The correspondence between K.B. numbers and network terminals or devices is also specified at SYSGEN time. These specifications establish a one-to-one correspondence between a RSTS K.B. number and a terminal or device on the network. This serves two purposes: first, it allows an "outgoing connection" to be established from the host to the terminal or device by simply "initing" the corresponding K.B. by a program; second, it guarantees that whenever the specified device or terminal connects to the host system it will always be mapped as the corresponding K.B. number. If the specified K.B. number is not available, the connection will be refused. This should normally not happen since K.B. numbers which are specified to correspond to a special network device or terminal are not used in the general pool of network K.B.'s.

The configuration of each RSTS host in a MICRO-NODE network is specified in the normal manner for RSTS with the addition of the information necessary to specify the connection to the network. An extended SYSGEN program will be provided, but, until it is available, the following proceedure should be used. The standard DEC SYSGEN program should be run to specify the RSTS system configuration except for the connection to MICRO-NODE. Any direct terminal connections to the RSTS system are specified at this time. Note that the "multi-terminal" and "echo control" features are not supported by this release of MICRO-NODE and must not be included in the RSTS configuration. The output of the SYSGEN program consists of two files: SYSGEN.MAC and SYSGEN.CTL. Both of these files must be edited to include the specification of the MICRO-NODE connection. In the CONFIG.MAC file, the following lines must be inserted just before the line which reads "; TERMINAL OPTIONS"

```
NX     =    1       ;1 IF NEXILIS NETWORK SUPPORT; ELSE 0
NXN    =    n       ;NUMBER OF INTERFACES TO NETWORK NODES
N0VEC  =    vec0    ;VECTOR ADDRESS FOR FIRST INTERFACE
N1VEC  =    vec1    ;VECTOR ADDRESS FOR SECOND INTERFACE
N0CSR  =    csr0    ;ADDRESS OF CSR FOR FIRST INTERFACE
N1CSR  =    csr1    ;ADDRESS OF CSR FOR SECOND INTERFACE
N0SOC  =    soc0    ;NUMBER OF SOCKETS ON FIRST INTERFACE
N1SOC  =    soc1    ;NUMBER OF SOCKETS ON SECOND INTERFACE
NT11   =    m       ;NUMBER OF NETWORK TERMINAL DDB'S
```

```
;DEFINE "OUTGOING" NETWORK SOCKETS
;  FORMAT IS:
;       KBN     K.B.NO.,DEV,CLASS,PRIORITY,TYPE,SUBTYPE,WRN,MAX

        .MACRO  OUTNET
        KBN     specification for outgoing device
        KBN     specification for outgoing device
        .ENDM

;DEFINE NETWORK MAP
;       FORMAT IS:
;       NODE    NAME,PRIMARY INTERFACE,SECONDARY INTERFACE

        .MACRO  NETMAP
        NODE    nodea,npa,nsa
        NODE    nodeb,npb,nsb
        .ENDM
```

In the above code, upper case letters indicate text that must be included exactly as given; lower case letters indicate text that will vary according to the configuration. Additional (or fewer) lines may be required in some places depending on the configuration. For each network interface, each of the symbols NnVEC, NnCSR, and NnSOC must be defined, where n is the number of the interface. The first interface is always numbered 0. Also, a separate KBN line must be included for each "outgoing" network device. The parameters in these lines must be defined as follows:

```
CLASS    = 200
PRIORITY = 0
TYPE     = 0
SUBTYPE  = 0
WRN      = 5
MAX      = 7
```

The DEVICE parameter is the full network name of the device; for a terminal this is:

TRnNODE

where "n" is the port on the node for the terminal and "NODE" is the name of the node. The K.B. number given must fall with the range of network K.B.'s in this configuration. Remember that network K.B.'s are assigned above all other K.B.'s in the configuration.

The network map section must include a "NODE" line for each node in the network, including nodes which are not directly connected to this host. The number following the node name indicates which directly connected node will be used to reach the indicated node for outgoing connections. This is the number which is used the NnVEC, NnCSR, and NnSOC symbols. If a node is not included in this table, it will still be able to connect to the host, but no outgoing connection from the host to that node will be possible.

The SYSGEN.CTL file must also be edited to include the required network modules. These edits are the same regardless of the configuration. Preceeding the line which contains only the text "TTDINT/C", a line must be inserted which reads "NXDINT/C". Preceeding the line which contains only the text "TTDVR", a line must be inserted which reads "NXDVR/C". Preceeding the first line which reads "$R LINK.SAV", the following text must be inserted:

After the above changes have been made, the RSTS monitor is built in the normal way using SYSBAT.

```
$R MACRO.SAV
NXDVR,NXDVR/C=IN:COMMON,KERNEL,DK:CONFIG,IN:CHECK,NXDEF,NXDVR
$R MACRO.SAV
NXDINT,NXDINT/C=IN:COMMON,KERNEL,DK:CONFIG,IN:CHECK,NXDEF,NXDINT
```

The files NXDINT.MAC, NXDVR.MAC, AND NXDEF.MAC must be present on the input device. Also, the versions of COMMON.MAC, KERNEL.MAC and TBL.MAC provided with MICRO-NODE must be used.

Appendix A

DEC 9301 Bootstrap Module

Nexilis Systems provides a set of PROM chips for the DEC 9301 bootstrap module which allow a Unibus PDP-11 node to be loaded through either a local (DR-11) or remote (DU-11 or DUP-11) connection or from paper tape (for diagnostics). A DEC compatible front-panel emulator is also provided for use with PDP-11/04 and PDP-11/34 machines equipted with the operators panel. A simple memory diagnostic routine is included.

Startup characteristics are determined by setting the appropriate switches on the 9301 card during installation. The bootstrap can be set up to expect commands from the console terminal or to boot automatically from any one of 9 devices without operator intervention.

The bootstrap is initiated by the operator when the node is powered up or when the "boot" button (programmers panel) or "init" switch (operator's panel) is pressed.

Console terminal commands

The console terminal startup option is selected during installation by setting the 9301 switches as per table A. The bootstrap then displays the contents of R0, R4, SP, R5 and the 's' prompt. Initially the registers contain garbage. Now any of the commands from the console command summary can be entered.

Console command summary

TT	Absolute papertape loader for DEC diagnostics
D0	DR11 unit 0 boot (local node)
D1	DR11 unit 1 boot (local node)
D2	DR11 unit 2 boot (local node)
D3	DR11 unit 3 boot (local node)
U0	DUx unit 0 boot (remote node)
U1	DUx unit 1 boot (remote node)
U2	DUx unit 2 boot (remote node)
U3	DUx unit 3 boot (remote node)

L <address>	load address
E	examine location
D <data>	deposit data
S	start program

Note - all commands must be in upper case.

After loading the node the bootstrap waits several seconds before starting the node.

The memory diagnostic option is selected with the appropriate switch settings from table A. This diagnostic tests only the lower 28K of memory. Any error detected will immediately halt the CPU at address 173742. The bad memory location can then be determined by examining and recording the contents of these registers:

R0 = expected data
R4 = received data
SP = failing address

This kind of error is not solved by skipping the memory diagnostic - your computer is sick! Call your DEC representative for assistance.

*** Warning ***

All normal boots should be done with the memory diagnostic option. This may be skipped but only at the risk of your system's integrity as bad memory can cause any number of mysterious problems.

The NX9301 bootstrap is capable of booting a MICRO-NODE node through DR11's, DU11's and DUP11's. These devices must be strapped for the standard MICRO-NODE address listed below.

Device	Unit	Address	Vector
DR11	0	767770	300
"	1	767760	310
"	2	767750	320
"	3	767740	330
DUx11	0	760010	340
"	1	760020	350
"	2	760030	360
"	3	760040	370

TABLE A

Automatic startup options	Starting address	Switch settings 34567890
Keyboard entry		
with memory diagnostics	773000	xxxxxxxx
w/o " "	773030	xxxxooxx
DR0 entry		
with memory diagnostics	773426	oxxxoxoo
w/o " "	773430	oxxxooxx
DR1 entry		
with memory diagnostics	773436	oxxxoooo
w/o " "	773440	oxxoxxxx
DR2 entry		
with memory diagnostics	773446	oxxoxxoo
w/o " "	773450	oxxoxoxx
DR3 entry		
with memory diagnostics	773456	oxxoxooo
w/o " "	773460	oxxooxxx
DUx0 entry		
with memory diagnostics	773470	oxxoooxx
w/o " "	773472	oxxoooxo
DUx1 entry		
with memory diagnostics	773500	oxoxxxxx
w/o " "	773502	oxoxxxxo
DUx2 entry		
with memory diagnostics	773510	oxoxxoxx
w/o " "	773512	oxoxxoxo
DUx3 entry		
with memory diagnostics	773520	oxoxoxxx
w/o " "	773522	oxoxoxxo
DL paper tape entry		
with memory diagnostics	773532	oxoxooxo
w/o " "	773534	oxoxooox

x = on
o = off

Switch 1 must always be on
Switch 2 should be on to enable automatic boot on power-up, off otherwise

Appendix B

DEC BDV11 bootstrap module

Nexilis Systems provides a set of PROM chips for the DEC BDV11 bootstrap module for the LSI-11 which allow a node to be loaded through either a local (DRV11) or remote (DUV11) connection or from paper (for diagnostics). A simple memory diagnostic routine is included.

Since there are no option switches on the BDV11 module, the bootstrap routine scans all devices for activity and loads from whichever one becomes active. The bootstrap is entered whenever the node is powered up or whenever the reset switch in operated.

Complete information on the operation of the BDV11 bootstrap will be provided in a later edition of the document.

Appendix C

Memory allocation

The following specifies all reserved PDP-11 memory locations. This includes the recommended device register and vector addresses for network devices and interfaces. Generally, only those devices which are to be used for booting must be at the specified addresses since all devices are specified in the config file for each node. It is strongly recommended that the addresses given here be used, however, to eliminate the possibilty of conflicts.

The following specify single words:

0/	Not used
2/	Not used

The following specify word pairs which contain the new PC in the first word and the new PSW in the second word.

4/	Bus time out trap
10/	Illegal instr trap
14/	BPT instr trap
20/	IOT instr trap
24/	Power fail trap
30/	EMT instr trap
34/	TRAP instr trap
40-54	Reserved to software
60/	CTY input interrupt
64/	CTY output interrupt
70/	
74/	
100/	LSI-11 clock interrupt
114/	Memory parity
120/	First DR11 input interrupt
124/	First DR11 output interupt
130/	Second DR11 input interrupt
134/	Second DR11 output interrupt
140/	Third DR11 input interrupt
200/	LP/11/TI810 printer interrupt
204/	RF11/RS11 RJS04/RWS04/RS04 RJS03/RWS03/RS03 disk interrupt
210/	RC11 disk interrupt
214/	TC11/TU56 DECtape interrupt
220/	RK11/RK05 disk interrupt
224/	TMA11/TU10 TS03 TJU16/TWU16/TU16 tape interrupt
230/	CR11/CD11 card reader interrupt
234/	
240/	PIRQ interrupt

244/	FIS trap
250/	Memory management trap
254/	RP11/RP03 RJP04/RWP04/RP04 disk interrupt
260/	TA11 cassette interrupt
264/	First RXV11 interrupt
270/	Second RXV11 interrupt
274/	
300-364	Asynchronous line interfaces or additional link or host interface interrupts (asynchronous interface vectors are assigned downwards from 364)

The following specify single words:

770200-770366	Processor dependent locations
770370-772036	
772040-772072	RJS04/RWS04/RS04 RJS03/RWS03/RS03 disk
772074-772436	
772440-772476	TJU16/TWU16/TU16 tape
772500-772516	
772520-772532	TMA11/TU10/TS03 tape
772534-772776	
773000-773376	BDV11 first window or REV11 first half
773400-773776	BDV11 second window or REV11 second half
774000-776676	
776700-776752	RP11/RP03 RJP04/RWP04/RP04 disk
776754-777156	
777160-777166	CR11/CD11 card reader
777170/	First RXV11 disk RXCS
777172/	First RXV11 disk RXDB
777174/	Second RXV11 disk RXCS
777176/	Second RXV11 disk RXDB
777200-777336	
777340-777350	TC11/TU56 DECtape
777400-777416	RK11/RK05 disk
777460-777476	RF11/RS11 disk
777500/	TA11 cassette status
777502/	TA11 cassette data
777504-777506	
777510/	TI810 printer control CONTROL
777512/	Not used
777514/	LP11/TI810 printer status
777516/	LP11/TI810 printer data
777520/	BDV11 base registers
777522/	BDV11 scratchpad
777524/	BDV11 switches
777526-777544	
777546/	KW11-L clock status
777550-777556	
777560/	CTY RCSR
777562/	CTY RBUF
777564/	CTY XCSR
777566/	CTY XBUF
777570-777776	Processor dependent locations

Index

SYSTEM XXV

FAILURE RECOVERY PROCEDURES

March, 1981 DRAFT
Version 2
Journal Number 75400

Office Automation Division
Tymshare, Inc.
20705 Valley Green Drive
Cupertino, California 95014

TABLE OF CONTENTS

1 INTRODUCTION

1.1 The System XXV

The brain of the System XXV is a Foonly computer and its memory. Associated with this are three disk drives, which store and retrieve information on the disks, and a tape drive for reading information into the system from tape. In addition, the system is connected to one or more networks which allow it to communicate with users and other computers. These various parts which make up a System XXV are run by a monitor (or operating system) called "AUGUST". AUGUST talks to the network, makes sure that all the parts of the system are working properly and in harmony, and oversees users' interaction with the programs run by the system. AUGUST also communicates directly with System XXV users through a program called "EXEC" and carries out many user commands.

1.2 Purpose and Structure of the Manual

This manual provides the information necessary to bring up a System XXV that has crashed. It is divided into several major sections, each of which covers a different situation you might encounter after a system crash. Their order is the same as the series of questions you might ask yourself when faced with a system that is not operating correctly. We hope that reading through the sections in order will enable you to step through the process of determining what is wrong with a system, deciding how to bring it up, and finally actually doing the recovery procedure you have decided on. Because the manual is arranged in this working order, its first several sections deal with error conditions and hung systems. Only after these problems are dealt with can we turn in section 5 to what to do if the system has crashed. This section discusses recovery procedures in general, the various types of recovery available on the System XXV, and when to use each of them. Section 5 is very important; do not skip it.

Where applicable and helpful, each large section of this manual is divided into four parts:

Introduction. A quick look at the current section. This will tell you such things as what the section is about, how the information is organized, and where to go if it is not what you need.

Summary. An outline that briefly presents exactly what you need to know or do. This is meant to be used as a working document or for quick reference; no explanation is included.

Discussion. A detailed explanation of the information and procedures outlined in the Summary.

Errors and Recoveries. A list of common problems and suggested solutions.

1.3 Conventions of the Manual

This manual has a set of conventions that will make it, we hope, clear and easy to read.

Program names always appear in capital letters, for example, CHECKDISK.

Special keys on the terminal are indicated by the abbreviations:

<SP> for space

<CR> for carriage return

<ESC> for altmode or escape

<LF> for line feed

Control characters are indicated with the notation "CTRL" and surrounded by angle brackets, for example, <CTRL-X>. To type a control character, hold down the CTRL key while typing the letter. To type <CTRL-X>, for instance, you would hold down the control key and at the same time type an X (in uppercase or lowercase).

The manual will refer to switches on the control panel by function in capital letters. When you look at the control panel, you will see that the switches are in rows and that different rows of switches are labeled by what they control; for example, there is a row of switches labeled "micro processor". Inside these rows of switches, individual switches are named by what they do: for instance, in the row of switches labeled "micro processor", there is a switch named "stop". This switch, which is used to stop the microprocessor, is called MICRO PROCESSOR STOP.

You put control panel switches "on" by pushing them up, and you put them "off" by pushing them down. Some switches are momentary, which means that after you put them on (up), they will return to the off (down) position when you release them. If you read "Put MICRO PROCESSOR STOP on", this means push up the switch labeled "stop" in the row of switches labeled "micro processor".

Commands appear in two ways. When the command is discussed in the text, the first letter of the command is capitalized and there are no quotation marks. When, on the other hand, you are directed to enter a specific command, for example in the Summary of a section, the command and its argument(s) are lowercase and enclosed in quotation marks. In this case, type exactly what you see, excluding, of course, the quotation marks. If the operator's terminal is uppercase only, you may type the commands in uppercase: however, the reverse is not true. All uppercase commands must be given that way; do not type them lowercase. Some prgrams cannot recognize uppercase letters.

2 ERROR CONDITIONS -- HOW TO IDENTIFY THEM

2.1 Introduction

When you are faced with a system experiencing some problem, the first thing you need to do is determine what this problem may be. There are three important aids in this process: error messages, BUGHLT numbers, and error lights. Always read and record the error messages, BUGHLT numbers, and error lights before you attempt to bring the system up.

2.2 Error Messages and BUGHLT Numbers

When the system crashes, it usually provides an error message on the operator's terminal specifying what caused the crash. This is followed by a BUGHLT number. Always read and record error messages and BUGHLT numbers when you have a system that is down. A list of the various BUGHLT numbers and what they mean comes with this manaul.

The system may be set so that it does not print the BUGHLT number, but only prints the word "BUGHLT" followed by the location of the BUGHLT. When this happens, type ".[" to force the system to print the number. The BUGHLT number will be the second, or right, half of the number printed.

2.3 Error Lights

When the system is functioning normally, certain lights on the control panel are on, others off. When the system crashes, these lights change. Lights that indicate the system is functioning correctly are replaced by error lights, lights indicating some error has occurred. This section will help you tell the difference between lights lit during normal operation and error lights.

Normal Lights

When the system is operating normally, a pattern consisting of four lights will be cycling among the address lights on the control panel. Unless the system is very heavily loaded, these lights should be moving. If they do not move for a reasonable period of time, the system is probably hung or down.

Error Lights

The following lights on the control panel are lit steadily only when an error has occurred and the system has crashed or will crash.

MEM PAR ERR light indicates a memory parity error.

MI PAR ERR light indicates a microcode parity error.

PROG HALT light indicates that the computer has encountered a halt instruction in AUGUST, the operating system. Systems programmers occasionally install halt instructions in AUGUST to help them trace problems.

3 IS THE SYSTEM HUNG OR HAS IT CRASHED?

Before you can deal with a system that is not operating correctly, you must determine whether it is hung or has crashed. Learning to recognize a hung system is a matter of practice. There is no one sure test that will determine if a system is hung, but hung systems do have the following common symptoms.

1) Lights on the control panel appear static or are immobile and pulsing in some kind of regular pattern.

2) There is no response when you type <CTRL-C> or <CTRL-T> on the operator's terminal.

3) You cannot log in from another terminal.

4) You are receiving irate calls from users who are unable to do anything.

5) In spite of all this, there is no BUGHLT indicated on the operator's terminal.

If a system is not functioning and does not have one or more of these symptoms, then it has crashed. See section 5, What to do if the System has Crashed.

4 WHAT TO DO IF THE SYSTEM IS HUNG

4.1 Introduction

The procedure documented in this section will force a hung system to crash. This may seem brutal, but it is necessary. The hung system is in limbo; only after it crashes can you bring it back up. When you finish this procedure and the system is down, use the disk recovery procedure to bring it up.

4.2 Summary

1) Put address switch 31 on (up).

2) Put data switch 2 on.

3) Put CONSOLE DEPOSIT THIS momentarily on.

4) Put data switch 2 off.

5) Put data switch 0 on.

6) Put CONSOLE DEPOSIT THIS momentarily on.

7) Wait until activity (the flickering of the lights, etc.) stops.

8) Bring the system up with the disk recovery procedure.

4.3 Discussion

When the system is hung, it is trapped in the execution of some process. The procedure outlined in the above Summary is designed to bring the system out of this cycle and cause it to crash. This is desirable because crashing is the normal response to abnormal conditions. When it crashes, the system tries to take care of itself -- to save files, to protect the monitor, to print an error message indicating what the problem may be, and so forth. Furthermore, only after it has crashed can the system be brought up.

Because a hung system ignores commands entered on the operator's terminal, to work with one, you must enter the data and commands manually from the control panel. Put on address switch 31 by pushing the switch up. Then, put on data switch 2. Finally, momentarily put on CONSOLE DEPOSIT THIS. This process turns on bit 2 at address 20 octal in the computer's memory. When you turn on this bit, you tell the system that everything that is stored in the temporary storage area should be read back into its permanent location. Temporary storage contains all new information the system has not read out to its real disk location and also the intermediate results from processes being performed but not yet completed. Before forcing the system to crash, you need to make sure that all this information is safely stored in the right place on the disk.

Now put data switch 2 off, put on data switch 0 and then put on CONSOLE DEPOSIT THIS. By doing this, you turn on bit 0 at address 20 octal. This bit is used by the system to record and check its status. When bit 0 is off, the system knows it is running successfully; when bit 0 is on, it means the system has encountered a dangerous situation and should crash. Thus, when you turn on bit 0 manually from the control panel, you trigger a system crash. Once all the flickering of the lights stops, the system is down. Bring it up with the disk recovery procedure.

5 WHAT TO DO IF THE SYSTEM HAS CRASHED

5.1 Introduction

When a System XXV crashes, before you can bring it up you must decide which recovery procedure to use. This section will help you do this. It is divided into three parts. The first part explains what the System XXV's recovery prodecures do, the second part briefly describes the recovery procedures available, and the third part will help you decide which procedure you need to use. In addition, we give advice about what to do if you cannot bring up the system. Once you know which recovery procedure to use, for specific instructions, go to the major section describing it.

5.2 What is a Recovery Procedure?

The System XXV is operated by a very large program called the "monitor". The monitor is basically what makes a machine into a computer. It is responsible for checking the system to make sure it is running correctly, transferring information within the computer and between it and the outside world, overseeing all the various programs run by the users, keeping the users' jobs separate and allocating resources to them, and so forth.

In keeping with its two functions, running the system and overseeing the users' programs and requests, the monitor is divided into two parts. The most important part is the "resident", or "kernel", monitor. The two names of this part of the monitor reflect its two major characteristics. "Kernel" monitor indicates that this part of the monitor is the core of the system. It contains basic instructions and information necessary for the system to function. For this reason, it must always remain, or reside, in central memory; thus, the name "resident" monitor.

The second part of the monitor, the "swappable" monitor, contains information and procedures related to users' needs rather than system functions. It is called the "swappable" because, unlike the resident monitor, this part of the monitor is not always present in central memory. Instead its various parts are copied or "swapped" into central memory only when they are needed. The Copy File to File process is an example of the type of procedure located in swappable monitor. When you enter a Copy command, the system begins by looking for this process in the parts of swappable monitor present in core. If it discovers that the Copy process is no longer im memory, it recalls it from disk and then executes your command.

The System XXV's entire monitor is called "AUGUST". AUGUST thinks it is running on a PDP10. Since it is not, AUGUST depends on another part of the system called the "microprocessor". The microprocessor is what translates the monitor's instructions into something the System XXV can understand. It consists of a memory containing information called "microcode", and a "microcontroller" that uses this information. When AUGUST, the monitor, gives a PDP10-like instruction, the microcontroller takes the instruction and uses the microcode to translate it into the equivalent instruction for a System XXV.

In most System XXV crashes the problem is an error in the monitor. The System XXV's various recovery procedures are designed to replace the old copy of the monitor with a new one from disk or tape where copies are permanently stored. Some crashes, however, destroy not only the monitor but also the microcode. Since the microprocessor cannot function without the microcode, this means that the microprocessor can no longer translate the instructions you or the monitor try to give it. In this case, before copying the monitor, the recovery procedure must also provide a new copy of the microcode. After the microprocessor has this copy of the microcode, the system is given the resident monitor. Once the resident monitor is safely stored in central memory, the system starts running and then copies the swappable monitor.

After the new monitor is in memory, in most recovery procedures, the system checks the file system with a program called CHECKDISK. If everything is OK, the system reports "August in operation". This means the system is ready to come up and open itself for normal use. If CHECKDISK discovers something wrong with file system, it will not come up. Instead, it will wait for you to correct the problem. After correcting the problem, you will have to halt the system and bring it up again. (As you will learn in the next section, some recovery procedures allow you to avoid this system checking.)

It should be emphasized that the reocvery procedures are simply programs like the monitor and everything else that runs on the system. They will not fix any hardward problems, and, in fact, cannot work if the system has something physically wrong with it. If you suspect that the system has a hardware problem, or you cannot bring it up after trying repeatedly, you may need to contact your manager and Tymshare maintenance.

5.3 Recovery Procedures Available

Introduction

The System XXV has five recovery procedures. None of them is particularly difficult, but they do have substantial differences. They are divided into two groups: those procedures which return the system to normal use, and those procedures which should be used only after very serious system error and which do not return the system to normal use. The Summary below lists all five procedures and mentions one or two of their most important features. Following this is a general discussion of what each procedure does and of the procedures' relationship to each other. To decide which procedure to use after a crash, see the next section. For details on exactly how each procedure works, see the individual section which discusses it. To learn about recovery procedures in general, see section 3.2, What is a Recovery Procedure.

Summary

Each of the following three procedures returns the system to normal operation and opens it to users.

Disk Recovery. You instruct the system to look on the disk for the information it needs to come up. Discussed in section 6.

Tape Recovery. You provide the information the system needs to come up from tape: providing a new copy of the microcode is an optional part of this procedure. Discussed in section 8.

Automatic Recovery. After crashing, the system immediately copies the information it needs from disk and tries to bring itself up. This is done automatically, without waiting for an operator. Discussed in section 7.

Both the next two procedures brings the system up closed to normal users and allows systems programmers to investigate what is going on. Use them only as a last resort, after serious system errors, and under supervision

Standalone Recovery. You instruct the system to come up without checking the file system or running the system jobs. Discussed in section 9.

Disk Rebuild. Before bringing the system up, you wipe out and then rebuild the entire file system, reading copies of every file from tape. Discussed in section 10.

Discussion

When the System XXV crashes, in most cases you bring it up by replacing the old copy of the monitor with a new one. You can provide this new copy either by copying it from disk, in which case you are doing a "disk recovery", or by reading it from tape for a "tape recovery". Both of these procedures are begun by an operator after the system has crashed. As part of the tape recovery procedure you may also read in new copy of the microcode the information used by the microprocessor. Replacing the microcode is usually necessary only after crashes due to power failure.

In addition to the tape and disk recovery procedures, there is another procedure that the system itself can start up after a crash. Since the system begins this procedure without waiting for anyone to instruct it, this third type of recovery is called "automatic recovery". Automatic recovery is very much like disk recovery. Upon crashing, the system immediately copies the current contents of central memory to disk, copies in a new monitor from disk, and starts to bring itself up. Automatic recovery never occurs unless the system was already set for it before crashing. To learn how to set a system to recover automatically, see section 12, Recovery Switches.

Disk, tape, and automatic recovery have substantial differences -- they are either automatic or not and the new monitor comes from either tape or disk. However, all three have the same result: they all end with the system checking itself and the file system and then being opened for normal use. The next two recovery procedures do not have this convenient result.

Both standalone recovery and disk rebuild allow the system to skip important parts of the normal recovery procedure. For this reason they are very risky; do not attempt them without being specifically instructed to do so and without supervision of a systems programmer or manager. In standalone recovery, the monitor is read from tape and you then direct the system to bypass its normal self-checking procedures and come up CLOSED to users. This means that only the operator's terminal has access to the system; no other users may log in. When a system will accept imput only from the operator's terminal, it is said to be "standalone". The standalone recovery procedures takes its name because it has this effect.

Even more serious than standalone recovery is disk rebuild. Disk rebuild allows you to do just what you might suspect from its name -- rebuild the file system stored on the disk. As in standalone recovery, you begin by reading in a tape containing a new monitor. Then, before the system needs to use any information from the disk, you begin the disk rebuild procedure. A disk rebuild involves destroying all the current versions of every file, and returning to the version stored on dump tapes; normally, it should NEVER be used.

5.4 Deciding Which Recovery Procedure to Use

Introduction

When a System XXV is down, whether for the first time in a month or minutes after a previous crash, the first step in bringing it up is deciding which recovery procedure to use. This section will help you make this choice. It is divided into two parts. The Summary contains a table showing types of crashes and their recovery procedures. The Discussion explains the logic behind the table; it will tell you why a particular recovery procedure is used in a certain set of circumstances. Once you determine which recovery procedure you need, go to the section discussing it to learn how to use it.

Summary

The table below shows when to use each of the System XXV's five recovery procedures. The left column lists different situations you might encounter; the right column shows the recovery procedure you should use. Note that you cannot decide to use automatic recovery after the crash has occurred. (Automatic recovery means that the system will try to bring itself up after a crash without waiting for an operator.) For automatic recovery to occur, the system must be set for it before the crash.

Situation	Recovery Procedure
Crash NOT DUE to power failure	Disk Recovery
Hung system is forced to crash	Disk Recovery
Any "normal" crash	Disk Recovery
After CHECKDISK problems corrected	Disk Recovery
Recovery begins automatically	Automatic Recovery
Crash DUE to power failure	Tape Recovery
Disk recovery fails	Tape Recovery
Automatic recovery fails	Tape Recovery
Tape recovery fails repeatedly	Standalone Recovery FIRST, contact manager or systems programmmer
Entire file system destroyed	Disk Rebuild FIRST, contact manager or systems programmmer

Discussion

The System XXV has three standard recovery procedures: disk recovery, tape recovery, and automatic recovery. In addition to these, there are two more risky recovery procedures, standalone recovery and disk rebuild, which should not be used without your manager's approval. This large number of choices means that you have more flexibility in responding to a crash, but it also means that you have more choices to make. Before you can bring up a system that is down, you must decide which recovery procedure to use. To make this decision, you must consider: 1) how the system will respond when it encounters an error while running; 2) the circumstances of the crash, what caused it, and what effect it had.

When a System XXV crashes, the first thing it does is check four internal switches, called "recovery switches". Recovery switches are four memory locations whose values tell the system how to respond to a crash. Recovery switches explained in section 12; here it is enough to know that they will tell the system to do one of two things: stop and wait for an operator, or immediately begin the automatic recovery procedure and try to come up. You should know how the recovery switches of each system have been set so you will know how it will respond to a crash. If you see a system crash and do not know what it will do, watch the system, until you know whether it is going to come up automatically or you need to begin a recovery procedure. If you see a system crash which you know is set for automatic recovery, it is wise to keep an eye on it and make sure it really does begin the automatic recovery procedure. Recovery switches are occasionally destroyed in system crashes. When this happens, a system originally set to recover automatically will simply sit there waiting.

If a system does manage to begin an automatic recovery, you have at first no decisions to make. If all goes well, the system will come back up and you will not need to do anything. However, automatic recovery does have two pitfalls. First, the recovery may not be successful and the system may hang or crash again. If you notice this happening, do not let another automatic recovery begin; the procedure is hardly likely to succeed on a second try. Instead, halt the system, if necessary, and bring it up yourself with the backup recovery procedure, tape recovery. The second problem that can keep the system from coming all the way up is errors in the file system. As the system comes up, it uses a program called "CHECKDISK" to examine the disk and make sure the files are OK. If CHECKDISK discovers problems, the system will stop to wait for someone to correct them. After correcting the errors, you will have to halt the system, and bring it up with disk recovery.

If, after a crash, a system does not try to come up automatically but instead just sits there waiting, then you must take over and begin some recovery procedure. Your choices at this point are disk recovery and tape recovery. Of these, disk recovery is more convenient and should be tried first, simply because it is so easy to use. However, disk recovery requires that the procedure itself survive the crash and that EDDT be available to begin it. Both of these are part of the old monitor. Crashes due to power failure destroy the old monitor (and microcode) completely and thus wipe them out. If you think the crash was the result of power failure, do not use disk recovery; instead, try tape recovery.

If you do decide to start with disk recovery and the system succeeds in running CHECKDISK, this means EDDT and the recovery procedure itself are OK. Even if CHECKDISK discovers file problems and the system does not come up, you may again use disk recovery, after correcting the problems and halting the system. File problems do not indicate that anything is wrong with the actual recovery procedure. However, if the first time you try disk recovery the system does not get as far as running CHECKDISK -- the recovery procedure never starts or the system crashes or hangs -- you will know that either EDDT or the disk recovery procedure or both did not survive the crash. In these circumstances, it is a waste of time to try the procedure a second time. Instead, halt the system, if necessary, and switch to tape recovery.

Tape recovery, the last of the three "normal" recovery procedures, is the backup procedure. In tape recovery the system copies the information it needs from tape; thus, recovery does not depend on any part of the system being able to function. Instead you enter all commands to the system through the control panel. However, although tape recovery is the most reliable of the recovery procedures, it is also the most time-consuming and inconvenient. Do not try tape recovery if you think disk recovery will work.

If you must use tape recovery, because the power failed or disk and automatic recovery do not work, feel free to try it several times. If you are using it after a power failure, or if nothing happens when you try to read the monitor tape, begin the procedure by reading in the microcode tape. If recovery does start, but the system never reaches the point of running CHECKDISK, halt the system, if necessary, and try the procedure over, again beginning by reading in the microcode tape. If, after trying the recovery three times from beginning, the system still does not run CHECKDISK, something may be seriously wrong. Notify a systems programmer or your manager; they may want to try standalone recovery. Once CHECKDISK does run, even if CHECKDISK discovers problems with the files system, the new monitor is in memory and this part of the recovery procedure has been a success. In addition, since EDDT and the disk recovery procedure are part of the monitor, they are again available. If CHECKDISK detects file problems and you must halt the system after correcting them, you may use disk recovery to bring the system back up.

Standalone recovery is the procedure used when some problem with the file system, CHECKDISK, the system jobs, and so forth, is causing the system to crash after the monitor is read into memory, but before it can come up all the way and return to normal use. In standalone recovery, after the monitor is copied from tape, the system simply stops where it is and waits. A systems programmer can then examine the monitor, the file system, and so forth. and try to determine what is wrong. A standalone recovery is not hard to do, but since the system comes up without checking how it is operating or making sure the file system is good, great harm may be done by mistake. Never undertake standalone recovery without expert supervision.

The final type of recovery procedure, disk rebuild, should be used only after a systems programmer has determined that a crash has damaged the file system beyond all hope of repair. Disk rebuild allows you to bring up the system in such a way that before anything is needed from the disk, the entire file system is replaced with backup files from tape. Because files can be replaced only with their most recent backups, the most current versions of many files will be permanently lost. The decision to do a disk rebuild can be made only by a manager or a systems programmer, and we hope the procedure will never have to be used.

5.5 Difficulty Bringing Up the System

If you cannot bring up the system or feel that something mysterious is going on, call Tymshare Maintenance or an OAD operating systems programmer.

6 DISK RECOVERY

6.1 Introduction

You should try to bring the System XXV up with the disk recovery procedure after any crash that is not due to power failure. The procedure is quick and easy; however, it relies on part of the monitor surviving the crash. This means it may not always work. Do not try to use disk recovery more than ONCE. If your first attempt at bringing up the system with disk recovery fails before CHECKDISK is run, you must halt the system, if necessary, and then switch to tape recovery. If CHECKDISK does run, then the system's new monitor is in place and this part of recovery has been successful. Even if CHECKDISK finds problems with the file system and you must halt the system after taking care of them, you may again use disk recovery to bring the system back up. To correct problems found by CHECKDISK and to halt the system, see section 13, Related Procedures.

6.2 Summary

1) Check the BUGHLT number on the operator's terminal and look it up in the list of BUGHLTs; in addition, note which error lights are lit. Record all this information.

2) On the operator's terminal, type "dskrld<ESC>g".

3) The reponse should be "reloading from disk". If you never get this message, begin a tape recovery.

4) When the system says, "BOOT FROM DISK PACK # [CR FOR ANY]", type <CR>. The system will begin to copy the monitor and will record its progress in messages.

5) When the operator's terminal says "EDDT", type "start<ESC>g". After a short time, the system should report the size of the memory and print several messages about BAT blocks.

7) CHECKDISK will run and check the file system. If it finds no major errors, it will report the number of disk pages used and the number available. If bad files are discovered, they will be listed and the system will announce "August not in operation".

8) If CHECKDISK runs successfully, the system will announce "August in operation" and ask for the date and time. Enter these in the form DD-MON-YY<SP>HH:MM; follow with <CR>. The system jobs will log in automatically.

9) When the system prompts you with "@", the prompt for EXEC, log in by typing "oper<SP>password<SP><CR>", where password stands for your password.

10) After the system prints various messages and prompts with you another "@", type "ena<CR>". The response will be a new prompt, "!".

11) Type "ref<SP>a<CR>".

6.3 Discussion

When the System XXV crashes, the first thing it does is check its recovery switches. (For information on recovery switches, see section 12, Recovery Switches.) If the recovery switches do not tell the system to come up automatically, the system simply stops and waits for someone to tell it what to do next. You now need to step in and bring the system up by providing a new copy of the monitor. In disk recovery, you do this by telling the system to get a new copy of the monitor from disk, where it is permanently stored. Disk recovery thus saves you the inconvenience of finding and loading the monitor tape and switching all the switches on the control panel. However, it will not always work. To start the copying procedure, you must use part of the old monitor called "EDDT". EDDT escapes most crashes without harm; however, crashes due to power failure always destroy EDDT and sometimes other crashes, for example, those due to power surges, will also damage it. If you suspect that the crash was due to power failure, do not try to bring up the system with disk recovery; use tape recovery instead.

You begin disk recovery by typing "dskrld<ESC>g". "Dskrld" stands for "disk reload"; it is the name of a location in the system's memory. This location is the beginning of the disk recovery procedure, a program that copies a new monitor from a file stored on the disk. When you type "dskrld<ESC>g", you tell EDDT to go to this procedure and begin running the program found there. When the procedure begins, it prints "reloading from disk". If this message never appears, it means EDDT was wiped out by the crash and you cannot reach the disk recovery procedure. In this case, begin a tape recovery.

If control is successfully transferred to the disk recovery procedure, the procedure first moves itself to a special spot in memory, beginning at location 3000, and makes room for the new monitor by clearing the rest of central memory. The system next needs to know where to should look for a new monitor. Each disk has a copy of the monitor stored in a file named <SYSTEM>MONITOR.PACK-x;1, where x stands for the disk number. The monitor file on disk pack 0, for example, is named MONITOR.PACK-0;1. To find out which disk it should check for the new monitor, the system will ask "BOOT FROM DISK PACK # [CR FOR ANY]". The standard answer here is <CR>. This tells the system to start by looking on disk pack 0 for the file; if it is not there, look on disk pack 1, and finally check disk pack 2. To tell the system to check only a particular disk pack, instead of answering the question with <CR>, give the number of the pack. If the system cannot find a good monitor file, it will print out "FAILED TO READ RESIDENT MONITOR". Since disk recovery cannot work without reading the resident monitor from disk, you will have to halt the system with Method B in section 13.5 and then use tape recovery to bring the system up.

If the system does find a usable copy of the monitor file, the disk reload procedure copies a new resident monitor from disk into the cleared memory. The system will inform you of its progress with various messages. Since you still don't really know if this procedure escaped the crash without harm, it is wise to keep an eye on these messages. If anything goes wrong before the system runs CHECKDISK and reports on the status of the file system, it means the procedure is unusable. If the system hangs as it comes up, halt it with Method B documented in section 13.5 of "Related Procedures" and bring it up with tape recovery. If the system crashes again, begin a tape recovery.

After the resident monitor is in memory, the system will go into EDDT and print "EDDT" on the operator's terminal. When you type "start<ESC>g", you transfer control to the Start procedure. This procedure starts up the rest of the recovery procedure and copies the swappable monitor from the second part of the monitor file.

As the system copies the new monitor, the old settings of the recovery switches are replaced by the default switch settings that are part of the new monitor. These default setting are: DBUGSW = 1, DCHKSW = 0, RELDSW = 1, and COMPSW = 1. This tells the system that after crashing it should stop and wait for instructions on what to do next. Once the system is up, you may change these default switch settings with the procedure documented in section 12, Recovery Switches. That section also explains recovery switches in general.

Once the system's new monitor is in place, the remainder of disk recovery is exactly the same as tape recovery. Thus, the following explanation is identical to the last part of the Discussion in the section on tape recovery. This explanation is included here for your convenience; if you are already familiar with tape recovery, you do not need to read further.

Now that it has its new monitor, the system turns its attention to the memory and file system. It first reports on the size of the memory and tells you about the BAT blocks. "BAT" stands for "Bad Address Table". BAT blocks contain tables that are used to keep track of what parts of the disk are bad and thus should not be used. Once it is determined what parts of the disk are bad and should not be used for storage, the system runs a program called "CHECKDISK". CHECKDISK, as the name indicates, checks the disks and the integrity of the file system It makes sure that no section of the disk is allocated to more than one file and that all file addresses are valid. If CHECKDISK discovers errors in the file system, it lists the bad files, and the system stops and waits for you to correct them. In this case, the system will not be able to come up; to let you know what is happening it will announce, "August not in operation". For more information on CHECKDISK and instructions for correcting the errors it detects, see section 13.2, Correcting Problems Found by CHECKDISK.

If CHECKDISK finds no serious file problems, it reports on disk use, and then, once it is finished, the system will announce, "August in operation". At this point, the system is completely ready to come up and open itself to users. It needs only two more things from you, the date and time. When the system directs you to enter the current date and tims, type two numbers for the day, a dash, the first three letters of the month, a dash, and then two numbers for the year. Follow these with a space and then give the time, on a 24 hour basis, as two numbers for the hour, a colon, and then two numbers for the minutes; be sure to give the correct time. Follow all this with a carriage return. For example, you would enter the date March 9, 1981, and the time 5:04 pm, as "09-mar-81<SP>17:04<CR>". If you enter the wrong date and time, finish the recovery procedure and then correct your mistake as documented in section 13.6, Changing the Date and Time.

After you have entered the date and time the system is officially up. The system jobs will now log in automatically and you will be prompted with "@", the prompt for EXEC. This is an invitation to log in. Log in as an operator by typing "oper<SP>password<SP><CR>", that is: "oper" (for operator), a space, your password, a space, and then a carriage return. In the interests of secrecy, your password will not print. After you have logged in, the system will print various messages and another "@". Type "ena<CR>". This stands for "enable" and tells the system to allow you to perform operations denied the normal user. Once you have "enabled", or identified yourself to the system as a person with special powers, the system will change its prompt to "!". Now refuse automatic logout by typing "ref<SP>a<CR>". AUGUST normally logs out users who leave their terminals idle.

6.4 Errors and Recoveries

Nothing happens when you type "dskrld<ESC>g"

If nothing happens when you type "dskrld<ESC>g", this means the disk recovery procedure cannot be used. Instead, use the tape recovery procedure.

The system cannot find a monitor file

If the system cannot find the monitor file, it will tell you "FAILED TO READ RESIDENT MONITOR". If you have told the system to look on a specific disk for the monitor, halt the system (with Method B of section 13.5), try another disk recovery and tell the system to look on a different disk for the monitor. If, after checking them all (either by typing a <CR> or individually giving the number of each disk), you discover that none of the disks have a good copy of the monitor file, you cannot use disk recovery. Halt the system, if necessary, and bring it up with a tape recovery.

Errors before CHECKDISK reports on the file system

If the system hangs or crashes before CHECKDISK reports on the status of the file system and you never get the message "August in operation" or "August not in operation", recovery will not be successful. Bring up the system with the tape recovery procedure.

CHECKDISK discovers problems with the file system

If CHECKDISK finds anything wrong with the file system, the System XXV will stop and wait for you to correct the problems. It cannot come up while something is wrong with the file system; the risk of destroying files is too great. For directions on how to correct any problems CHECKDISK finds, see section 13.2, Correcting Problems Found by CHECKDISK.

7 AUTOMATIC RECOVERY

7.1 Introduction

How a System XXV responds to a crash is determined by its recovery switches. Recovery switches are explained in section 12, Recovery Switches. If they are set for automatic recovery BEFORE a crash, then, after one occurs, the system should immediately try to bring itself up with the procedure documented here. This is convenient, since you do not have to start a recovery procedure every time the system crashes. However, do not assume that systems set for automatic recovery will never need your help. If the crash was due to power failure or it damaged memory, automatic recovery will never begin; the system will simply sit there and you will have to begin a tape recovery. If the recovery procedure was somehow damaged in the crash, the system may start to bring itself up and then hang or crash again. In this case too, you must step in, halt the system, if necessary, and use tape recovery. Even if automatic recovery begins and gets as far as running CHECKDISK, success is not guaranteed. If CHECKDISK detects problems in the file system, automatic recovery can proceed no further. After correcting the file problems and halting the system (both documented in section 13, Related Procedures), you will have to use disk recovery to bring the system up.

7.2 Summary

If a System XXV set for automatic recovery comes up successfully, you do not need to do anything until you log in as an operator. If recovery never starts, if it fails before CHECKDISK reports on the file system, or if CHECKDISK discovers bad files, see "Errors and Recoveries" in this section for instructions.

1) The system will begin to bring itself up; various messages will record its progress.

2) CHECKDISK will run and check the file system. If it finds no major errors, it will report the number of disk pages used and the number available. If bad files are discovered, they will be listed and the system will announce "August not in operation".

3) If CHECKDISK runs successfully, the system will announce "August in operation". The system is now up. The system jobs will log in automatically.

4) When the system prompts you with "@", the prompt for EXEC, log in by typing "oper<SP>password<SP><CR>", where password stands for your password.

5) After system prints various messages and prompts with you another "@", type "ena<CR>". The response will be a new prompt, "!".

6) Type "ref<SP>a<CR>".

7.3 Discussion

When a System XXV crashes, the first thing it does is check its recovery switches, the switches which tell it what it should do next. Recovery switches and their various settings are explained in section 12, Recovery Switches. One setting of the recovery switches will tell the system to come up automatically. If, after a crash, the system discovers that the switches are set in this way, it will immediately start to bring itself back up without waiting for an operator.

There are, however, several problems that can stop systems from coming up automatically. First of all, the system may never find out that it was supposed to do this. In crashes due to power failure and those that damage memory, the recovery switch settings may be lost or never checked. Consequently, a system you think is set to come up automatically will not. Instead, it will wait for you to begin a recovery procedure, just as it normally does after a crash. Keep an eye on all systems set for automatic recovery; if you see a system that appears to be down and not trying to come up, you will have to use the tape recovery procedure to bring it up. Do not try to use the disk recovery procedure; it too will be lost along with the recovery switches.

If the system does remember its recovery switch settings and try to come up automatically, it usually begins by copying the current contents of central memory into two files. The first 512 pages of memory are stored in a file called <SYSTEM>CORDMP.LOW and the second 512 pages are stored in a file called <SYSTEM>CORDMP.HGH. These files are used by systems programmers to find out what was in central memory right after the crash. If you do not want the system to bother with this copying, you may set the recovery switches so that it will not be done. See section 12, Recovery Switches.

Once the contents of core have been safely stored in the CORDMP files, the system next needs a new copy of the monitor. The system copies the monitor from disk with a procedure very much like the disk recovery procedure. The procedure prints messages to help you follow its progress. It is a very good idea to read these messages and make sure recovery is progressing successfully. Even after the recovery procedure starts, things can still go wrong. If the procedure was damaged by the crash, the system may hang as it comes up or may try to come up, fail, and crash again. After crashing, the system would once more check the recovery switches, discover it should come up automatically. take another core dump, and try to come up. As it tried to come up, the system would encounter the same problem and crash again. The system could thus get caught in a loop of crashing, trying to come up, and crashing again. If you notice a system set for automatic recovery that appears to be hung or having some kind of trouble, watch it for a while. If it never gets to the point of running CHECKDISK, halt the system with Method B of section 13.5, Halting the System. Then switch to tape recovery.

If all goes well with the automatic recovery procedure, it will announce "reloading from disk", move itself to a special place in memory, starting at location 3000 and clear the rest of core. A new resident monitor is then copied from a file where it is is permanently stored on the disk. After the resident monitor is read, the system starts up and transfers control to the Start procedure. This very much like the procedure you get when you give the Start command in the disk recovery procedure. The Start procedure starts up the rest of the reocvery procedure and copies the swappable monitor from the second part of the MONITOR file.

Now that it has its new monitor, the system turns its attention to the memory and file system. It first reports on the size of the memory and tells you about the BAT blocks. "BAT" stands for "Bad Address Table". BAT blocks contain tables that are used to keep track of what parts of the disk are bad and thus should not be used. Once it is determined what parts of the disk are bad and should not be used for storage, the system runs a program called "CHECKDISK". CHECKDISK, as the name indicates, checks the disks and the integrity of the file system It makes sure that no section of the disk is allocated to more than one file and that all file addresses are valid. If CHECKDISK discovers errors in the file system, it lists the bad files, and the system stops and waits for you to correct them. Thus, if CHECKDISK discovers problems, the system cannot come all the way up automatically. Instead, the system will announce, "August not in operation" you must take over and fix the file problems CHECKDISK has found. For instructions on how to do so, see section 13.2, Correcting Errors Found by CHECKDISK.

If CHECKDISK finds nothing wrong with the file system, it reports on disk use, and then, once it is finished, the system will announce, "August in operation". At this point, the system is up. Notice that you are not required to enter the date and time as you must do to end the disk and tape recovery procedures. During automatic recovery, unlike the other recovery procedures, the system's internal clock continues to run. To learn the correct time, the system simply uses it instead of asking you. In addition to using the system's clock to find the time, the end of the automatic recovery procedure differs in another way from disk and tape recovery. During automatic recovery, the system saves the original recovery switches settings. When recovery is over, these settings are restored and replace the default switches settings that are read in as part of the new monitor. This means that after an automatic recovery the recovery switches continue to be set for automatic: next time the system crashes it will again try to bring itself up automatically.

Once the system has found out the time and come all the way up, the system jobs can log in automatically and you will be prompted with "@", the herald for EXEC. This is an invitation to log in. Log in as an operator by typing "oper<SP>password<SP><CR>", that is: "oper" (for operator), a space, your password, a space, and then a carriage return. In the interests of secrecy, your password will not print. After you have logged in, the system will print various messages and another "@". Type "ena<CR>". This stands for "enable" and tells the system to allow you to perform operations denied the normal user. Once you have "enabled", or identified yourself to the system as a person with special powers, the system will change its prompt to "!". Now refuse automatic logout by typing "ref<SP>a<CR>". AUGUST normally logs out users who leave their terminals idle.

7.4 Errors and Recoveries

The System never begins to bring itself up

If a system that is supposed to be set for automatic recovery never announces "reloading from disk" to show it has begun, bring the system up with tape recovery.

Errors before CHECKDISK reports on the file system

If the system hangs or crashes before CHECKDISK reports on the status of the file system and you never get the message "August in operation" or "August not in operation", recovery will not be successful. Bring up the system with the tape recovery procedure.

CHECKDISK discovers problems with the file system

If CHECKDISK finds anything wrong with the file system, the System XXV will come up automatically; the risk of destroying files is too great. Instead, it will stop and wait for you to correct the problems. For directions on how to correct any problems CHECKDISK finds, see section 13.2, Correcting Problems Found by CHECKDISK.

8 TAPE RECOVERY

8.1 Introduction

Tape recovery is basically a backup recovery procedure. It should be used after automatic or disk recovery has failed or after a crash due to power failure. Tape recovery is normally a straightforward and not particularly difficult procedure, but it can grow rather complicated, particularly if the crash has somehow damaged the file system or wreaked other havoc. If you encounter problems while bringing the system up, check section 8.4, Errors and Recoveries -- you may find the solution to your problem there. If your problem is not covered in Errors and Recoveries, try the whole procedure over, starting from step 3. If this does not work, something may be seriously wrong. Notify your manager: he or she may want to try Standalone Recovery or, as a last resort, Disk Rebuild.

8.2 Summary

1) Check the BUGHLT number on the operator's terminal and look it up in the list of BUGHLTs; in addition, note which error lights are lit. Record all this information.

2) If the power has gone off, you must reload the microcode as documented in steps 3 through 12. If the power has not gone off, skip to step 13.

3) Mount the microcode tape on the tape drive.

4) Put all switches on the control panel off (down).

5) Put address switch 32 on (up).

6) Put MICRO PROCESSOR STOP on.

7) Put MICRO PROCESSOR MIPC on.

8) Put MICRO PROCESSOR CLR momentarily on.

9) Put MICRO PROCESSOR CONT momentarily on.

10) Put MICRO PROCESSOR MIPC off.

11) Put MICRO PROCESSOR STOP off.

12) Put MICRO PROCESSOR CONT momentarily on. The tape should spin and then stop. Remove the microcode tape from the tape drive.

13) You are now ready to read in the new monitor. Mount the monitor tape on the tape drive. [Start here if you do not want to load the microcode.]

14) Put all switches on the control panel off.

15) Put address switches 24 and 26 on.

16) Put MICRO PROCESSOR STOP on.

17) Put MICRO PROCESSOR MIPC on.

18) Put MICRO PROCESSOR CLR momentarily on.

19) Put MICRO PROCESSOR CONT momentarily on.

20) Put MICRO PROCESSOR MIPC off.

21) Put MICRO PROCESSOR STOP off.

22) Momentarily put MICRO PROCESSOR CONT on. The tape should spin and then stop.

23) Put address switches 24 and 26 off.

24) Put address switches 29 and 30 on.

25) Momentarily put CONSOLE START on twice.

26) When the operator's terminal says "EDDT", type "start<ESC>g". The monitor tape should spin.

27) Remove the monitor tape from the tape drive.

28) After the system reports the size of the memory, put MI PAR ERR STOP and MEM PAR ERR STOP on. The system will print several messages about BAT blocks.

29) CHECKDISK will run and check the file system. If it finds no major errors, it will report the number of disk pages used and the number available. If bad files are discovered, they will be listed and the system will announce "August not in operation".

30) If CHECKDISK runs successfully, the system will announce "August in operation" and ask for the date and time. Enter these in the form DD-MON-YY<SP>HH:MM; follow with <CR>. The system jobs will log in automatically.

31) When the system prompts you with "@", the prompt for EXEC, log in by typing "oper<SP>password<SP><CR>", where password stands for your password.

32) After system prints various messages and prompts with you another "@", type "ena<CR>". The response will be a new prompt, "!".

33) Type "ref<SP>a<CR>".

8.3 Discussion

Although you can bring System XXV up after most crashes simply by providing it with a new monitor, this is not always true. Some crashes, especially those due to power failure, wipe out not only the monitor, but also the microcode. Since the microcode is the information the microprocessor uses to translate the monitor instruction into something it can understand, when this happens you must begin recovery by reading in a tape containing the microcode. Only after the mocrocode is available can the system correctly read the monitor tape.

Since a system that does not have access to microcode or any part of the monitor cannot understand any instructions given on the operator's terminal, to reload the microcode you must give the system instructions from the control panel. Mount the microcode tape on the tape drive and put address switch 32 on by pushing the switch up. Also, find the row of switches labeled "MICRO" and put the MICRO PROCESSOR STOP and MICRO PROCESSOR MIPC on. You then briefly put on MICRO PROCESSOR CLR followed by then MICRO PROCESSOR CONT. This process tells the microprocessor that the address specified through the address switches is where it should look for instructions on what to do next. The address you specify by putting on address switch 32 is address 10 octal. This is the beginning of a tape-reading routine which is permanently stored in the memory of the microprocessor. You now want to tell the microprocessor to execute this routine and read the tape containing the microcode. You do this by putting off MICRO PROCESSOR MIPC and MICRO PROCESSOR STOP and putting on MICRO PROCESSOR CONT.

Once the system has read the tape containing the microcode, it has all the information necessary to read the first part of the monitor tape, which contains the resident monitor. The procedure for reading this tape is identical to that for reading the microcode tape, EXCEPT that you specify a different address with the address switches. First put off all the switches on the control panel, then put on address switches 24 and 26, put on MICRO PROCESSOR STOP and MICRO PROCESSOR MIPC, and finally, again momentarily put on MICRO PROCESSOR CLR and MICRO PROCESSOR CONT. This tells the system it should begin executing the instructions at address 5000 octal, the address specified with address switches 24 and 26. Address 5000 is the beginning of instructions for reading the monitor. To execute these instructions, put MICRO PROCESSOR MIPC and MICRO PROCESSOR STOP off and again momentarily put on MICRO PROCESSOR CONT. The microprocessor will read into memory the first part of the monitor tape: this contains the resident monitor.

Once the resident monitor is in memory, you want to start it running. When you put on address switches 29 and 30 and then hit CONSOLE START twice, you tell the system to go to location 140 and start running the procedure it finds there. The procedure beginning at this location brings the system alive and starts up the resident monitor. Because EDDT is part of the resident monitor, the system can now go into EDDT and will print "EDDT" on the operator's terminal to notify you. Since EDDT can understand typed commands, you can start up the rest of the recovery procedure by typing "start<esc>g" on the operator's terminal. The system will begin running and read in the swappable monitor, the second part of the monitor tape.

When the system copies the new monitor, the old settings of the recovery switches are changed to the default switch settings that are part of the new monitor. These default settings are: DBUGSW = 1, DCHKSW = 0, RELDSW = 1, and CDMPSW = 1. This tells the system that after crashing it should stop and wait for instructions on what to do next. Once the system is up, you may change these default switch settings with the procedure documented in section 12, Recovery Switches. That section also explains recovery switches in general.

Now that it has its new monitor, the system turns its attention to the memory and file system. It first checks how much memory is physically available and reports on the size of the memory. Putting on MI PAR ERR STOP and MEM PAR ERR STOP tells the system to stop if a parity error is encountered in central memory or in the microcode. The system will next tell you about the BAT blocks. "BAT" stands for "Bad Address Table". BAT blocks contain tables that are used to keep track of what parts of the disk are bad and thus should not be used. Once it is determined what parts of the disk are bad and should not be used for storage, the system runs a program called "CHECKDISK". CHECKDISK, as the name indicates, checks the disks and the integrity of the file system It makes sure that no section of the disk is allocated to more than one file and that all file addresses are valid. If CHECKDISK discovers errors in the file system, it lists the bad files, and the system stops and waits for you to correct them. In this case, the system will not be able to come up. Instead, after CHECKDISK runs, the system will announce, "August not in operation". For more information on CHECKDISK and instructions for correcting the errors it detects, see section 13.2, Correcting Problems Found by CHECKDISK.

If CHECKDISK finds no serious file problems, it reports on disk use, and then, once it is finished, the system will announce, "August in operation". At this point, the system is completely ready to come up and open itself to users. It needs only two more things from you, the date and time. When the system directs you to enter the current date and tims, type two numbers for the day, a dash, the first three letters of the month, a dash, and then two numbers for the year. Follow these with a space and then give the time, on a 24 hour basis, as two numbers for the hour, a colon, and then two numbers for the minutes: be sure to give the correct time. Follow all this with a carriage return. For example, you would enter the date March 9, 1981, and the time 5:04 pm. as "09-mar-81<SP>17:04<CR>". If you enter the wrong date and time, finish the recovery procedure and then correct your mistake as documented in section 13.6, Changing the Date and Time.

After you have entered the date and time the system is officially up. The system jobs will now log in automatically and you will be prompted with "@", the prompt for EXEC. This is an invitation to log in. Log in as an operator by typing "oper<SP>password<SP><CR>", that is: "oper" (for operator), a space, your password, a space, and then a carriage return. In the interests of secrecy, your password will not print. After you have logged in, the system will print various messages and another "@". Type "ena<CR>". This stands for "enable" and tells the system to allow you to perform operations denied the normal user. Once you have "enabled", or identified yourself to the system as a person with special powers, the system will change its prompt to "!". Now refuse automatic logout by typing "ref<SP>a<CR>". AUGUST normally logs out users who leave their terminals idle.

8.4 Errors and Recoveries

Using an old monitor tape

You may sometimes have to perform a tape recovery with an old monitor tape, for example, when you do not have a copy of the current monitor or the current tape is bad. When this happens, you can use the resident monitor from an old tape to start the system runnin. Once the system is up, you can switch to disk recovery to replace the old resident monitor with a good copy of the monitor taken from disk. The procedure is as follows:

1) Follow the tape recovery procedure from step 13 through 25. If there has been a power, problem, do step 3 through 25.

2) When the system types "EDDT" on the operator's terminal, type "dskrld<ESC>g" to start a disk recovery.

3) Follow the disk recovery procedure beginning from step 3.

Problems reading the microcode tape

If you cannot read the microcode tape, there is a hardware problem. Call Tymshare Maintenance.

Problems reading the monitor tape

If you cannot read the monitor tape, the microcode may have been destroyed. Start the recovery procedure over and begin at step 3 by loading the microcode. If you do not succeed, call Tymshare Maintenance.

The interrupt message

If you get a message, "Interrupt at nnn.", where nnn is some number, try reading both tapes again. If you are unsuccessful, call Tymshare Maintenance.

CHECKDISK is never run

If the system hangs or crashes before CHECKDISK reports on the status of the file system and you never get the message "August in operation" or "August not in operation", recovery will not be successful. Halt the system, if necessary, and try the recovery procedure over from step 3. If the complete procedure does not work on the third try, call Tymshare Maintenance.

CHECKDISK discovers problems with the file system

If CHECKDISK finds anything wrong with the file system, the System XXV will stop and wait for you to correct the problems. It cannot come up while something is wrong with the file system; the risk of destroying files is too great. For directions on how to correct any problems CHECKDISK finds, see section 13.2, Correcting Problems Found by CHECKDISK.

9 STANDALONE RECOVERY

9.1 Introduction

Bringing the system up with the standalone recovery procedure is useful when some error in the disk, CHECKDISK, or the system jobs is causing the system to crash before it can come all the way up. During a standalone recovery, the system does not run CHECKDISK and the other checking programs that are part of the three "normal" recovery procedures. Instead, the system comes up without checking itself and, after it is up, is shut to normal users; only the person at the operator's terminal is allowed in. Standalone recovery is risky. Do not bring the system up with this procedure unless specifically instructed to do so.

9.2 Summary

1) Check the BUGHLT number on the operator's terminal and look it up in the list of BUGHLTs; in addition, note which error lights are lit. Record all this information.

2) Follow the procedure for tape recovery (section 8) from step 13 through step 25. If you suspect there has been a power failure, do step 3 through 25 of the tape recovery procedure.

3) When the operator's terminal says "EDDT", type "dbugsw/". The system will print either 0 (zero) or 1.

4) Type "2<CR>".

5) Type "start<ESC>g". The monitor tape should spin.

6) The system will request the date and time. Enter these as DD-MON-YY<SP>HH:MM and follow with <CR>.

7) You will automatically be logged in as "system", but not enabled.

9.3 Discussion

To use standalone recovery procedure, you begin by following the tape recovery procedure. But after the system reads the first part of the monitor tape and tells you it is in EDDT, you do NOT type "start<esc>g", to start the system running and read in the rest of the tape. Instead, you work in EDDT an interactive language for debugging. EDDT is part of the resident monitor and is used to patch and otherwise manipulate it. Because it can change the monitor, EDDT is a very powerful tool; use it with care.

Once the system tells you that you are in EDDT, do not wait to be prompted. EDDT has no prompt; as you use it, it simply waits for you to type something and then reacts. When you see that are in EDDT, immediately type "dbugsw/". This command has two parts. The first part, "dbugsw", is the name of an address. The "/" means "print". Thus, "dbugsw/" instructs the system to print what it finds at the location DBUGSW. This location contains one of the system's recovery switches, the debugging switch. The number at this address tells the system what it should do when it encounters a fatal error. A zero (0) at DBUGSW means the system should respond to errors by crashing. A 1 means the system should take breakpoints; that is, when a fatal error occurs the system should not crash but should stop where it is, preserve the context of the error, and print out a BUGHLT address. This address is what you record after a crash when you are instructed to record the BUGHLT number. Knowing the address of the error that caused the system to crash helps systems programmers find out what happened. Recovery switches are further explained in section 12, Recovery Switches.

After you print the current contents of DBUGSW, type "2<CR>. This tells the system to enter 2 at this location. When DBUGSW is 2, it instructs the system to skip running CHECKDISK and the system jobs, and to come up "standalone". When a system comes up standalone, it accepts input only from the operator's terminal: it does not allow any ordinary users to log in.

Once you have made sure the system will come up isolated from the outside world, you start it by typing "start<ESC>g". The system will read the second part of the monitor tape, the part containing swappable monitor, and ask you for the date and time. After you have entered these (as DD-MON-YY<SP>HH:MM<CR>), the system will come up and automatically log you in as "system". This automatic login keeps the system from going through the complicated login procedure. When you are logged in as "system", you have the same powers as if you had logged in as "operator"; remember to enable if you want to do anything requiring special powers.

10 DISK REBUILD STRATEGY OR TOTAL CATASTROPHE

10.1 Introduction

Occasionally, a particularly deadly system crash destroys the file system. If you suspect this has happened, immediately notify your manager and, if possible, an OAD operating systems programmer; do not attempt to do more.

When the file system is destroyed, you must use the various dumps made each week to rebuild the disk and restore the files as completely as possible to their pre-crash state. This must be finished before the system needs anything from the disk. Rebuilding the disk is a fairly simple procedure, but the loss of users' files and the possibility that they may be damaged or incompletely restored is so serious that you should NEVER undertake a disk rebuild without specific instructions and assistance of a manager or an operating systems programmer.

10.2 Summary

WARNING: Never attempt this without specific instructions from a manager or an operating systems programmer.

1) Check with Tymshare Maintenance to make sure the hardware is good.

2) Follow the tape recovery procedure from steps 13 through 25. If there has been a power failure, do steps 1 through 25.

3) When the operator's terminal says "EDDT", type "dbugsw/". The system will print either zero (0) or one (1).

4) Type "2<CR>".

5) Type "syslod<ESC>g".

6) The system will ask, "Do you really want to clobber the disk by reinitializing?".

7) Type "y<CR>". This stands for "yes". Do not type more than y.

8) The system will say, "OK, You asked for it..."

9) The system will reinitialize all the files and then report, "No EXEC".

10) Load the DLUSER tape on the tape drive.

11) Type "l", for "load". When the system asks, "Load from magtape MTAN:", type "mta0:<CR>" ("0" here is zero) and confirm with another <CR>.

12) When the system asks, "File Number?", type "0<CR>".

13) The system will now read the DLUSER file, the first part of DLUSER tape.

14) When it has finished, the system will prompt you with a period (.), the prompt for MINI-EXEC. At the period, type "s.".

15) The system will print, "Interrupt at nnn", where nnn is some number, followed by a period.

16) To read the second file in the tape, the DUMPER file, type "l", for "load". When the system asks, "Load from magtape MTAN:", type "mta0:<CR>", and confirm with another <CR>.

17) When the system asks, "File Number?", type "1<CR>".

18) The system will now read the DUMPER file.

19) When the system prompts you with a period, type "s.".

20) The program DUMPER is now loaded and ready to start restoring the files. Mount the first Full Dump Tape. Make sure you load the Full Dump Tapes in numerical order.

21) DUMPER will now ask a series of questions, preceding each of them with instructions.

22) To answer the first question, "DUMP, LOAD, CHECK, OR SINGLE?", type "l", for "load".

23) For the second question, "DO YOU WISH TO SUPERSEDE OLDER VERSIONS ALWAYS?", type "n", for "no".

24) When DUMPER asks, "SPECIFIC USERS?", type "n".

25) When it asks, "INTO SAME DIRECTORIES?", type "y".

26) Finally, when requested, "TYPE MAG TAPE UNIT NUMBER", type "0" (zero).

27) DUMPER will now read the tape; when it is finished, it will print, "MOUNT NEXT TAPE, IF ANY. TYPE C, WHEN READY, N, IF NO MORE". Mount the next tape and type "c".

28) DUMPER will again ask for the mag tape unit number; type "0".

29) Continue mounting and loading Full Dump Tapes, typing "c" to continue and then giving 0 (zero) for mag tape unit number, until all the tapes have been read.

30) When you have finished loading the Full Dump Tapes, begin loading the Incremental Dump Tapes. Be sure that you load the Incremental Dump Tapes in chronological order, beginning with the one made right after the Full Dump and ending with the most recent.

31) After the system has read the last Incremental Dump Tape, when DUMPER says, "MOUNT NEXT TAPE, IF ANY. TYPE C, WHEN READY, N, IF NO MORE", type "n".

32) DUMPER will stop and the system will print an interrupt message followed by a period, the prompt for MINI-EXEC.

33) The files have now been restored as completely as possible. Halt the system, and begin a disk recovery. (To halt the system use Method A of section 13.5, Halting the System.)

10.3 Discussion

A disk rebuild is necessary when a crash destroys the files. Since a system that has lost its files cannot be expected to run CHECKDISK or the system jobs successfully -- these are stored on the disk and everything on the disk has been lost -- the system must be brought up in such a way that it does not need anything from its files; in fact, it does not even realize they are lost. This means the system must be brought up standalone, since in a standalone recovery the system skips running the system jobs, does not check the file system with CHECKDISK , and does not open itself for normal use.

But bringing up the system for a disk rebuild is not a completely "normal" example of bringing the system up standalone. After you have set DBUGSW to 2 (to make the system come up without checking itself and closed to users), you do not then type "start<ESC>g" to start the system running. Instead you type "syslod<ESC>g". This stands for "system load". It tells the system to begin running a program that wipes out all existing files and then allows you rebuild the entire file system with files copied from tape.

When you give the Syslod command, the system will ask, "Do you really want to clobber the file system?". When you respond "y", for "yes", it will print, "OK, you asked for it..." and reinitialize all the files. When the files have been reinitialized, the system will state: "No EXEC". EXEC disappears because it was stored on the disk. After informing you of EXEC's disappearance, the system will prompt you with a period (.), the prompt for MINI-EXEC. MINI-EXEC is a group of basic commands that are loaded with the monitor. MINI-EXEC has two important features: First, MINI-EXEC recognizes commands by their first letter, so you type "l" for "load", "s" for "start", and so on; and second, in MINI-EXEC you must end all simple commands that do not ask you for further information with a period for confirmation.

NOTE: Because MINI-EXEC recognizes commands by their first letter, if you make a mistake in giving a simple command, type a few more random characters before you confirm with a period. The additional letters you type will make the command unrecognizable. When you are again prompted with a period, repeat the command you want. If you are giving a command that asks you for further information before it is executed, type some random characters in answer to the additional question. This will cause the command to be aborted and the period will reappear.

Once you are in MINI-EXEC, you can begin the procedure for rebuilding the disk from backup tapes. Mount the DLUSER tape on tape drive zero, and type "l", for "load". When the system asks you, "Load from magtape MTAN:", identify your drive as "mta0:" and confirm by typing "<CR><CR>". The system will now ask which file on the tape it should read by printing: "File Number?". The number of the DLUSER file, which should be printed on the tape casing, is zero (0). Enter this and follow it with a carriage return.

DLUSER stands for "dump and load users". The DLUSER file contains data about all the directories on the system, both the user and system directories, and a program that can use this information to rebuild them. When you type "s.", you instruct the system to run this program.

When the system has rebuilt the directories, it will print an interrupt message. This means it is ready to read another file. You now want to load the file containing the DUMPER program. To do this again, type "l", and then again, when asked, "Load from magtape MTAN:", identify your drive as "mta0:" and confirm by typing "<CR><CR>". Next, you will be asked for the file number. The file number for the DUMPER file is one (1); this also should be printed on the tape casing. Enter 1 and follow it with a confirming <CR>. The DUMPER file contains a program able to read files from tape and restore them to the correct directories. Once you have loaded DUMPER and typed "s." to start it, the DUMPER program will start running. You can now use this program to rebuild the disk by mounting and reading back into the system the dump tapes that contain the back-up versions of all the files on the disk.

To ascertain what it should do, DUMPER will ask you a series of questions. Each question is preceded by an explanation of how you should answer it. These explanations are designed for people using DUMPER for routine maintenance of the file system. Do not be alarmed if the answers you are instructed to give here do not agree with what the system tells you to do. Disk rebuild is not a normal situation.

The first thing DUMPER will want to know is what you plan to do. To find out, DUMPER will ask: "DUMP, LOAD, CHECK, OR SINGLE?". Since you want to load files from tape back into the disk type "l", for "load". Then mount the first Full Dump Tape on the tape drive. Full Dump Tapes are tapes made at regular intervals, usually weekly, that contain a record of the entire contents of the disk. The first Full Dump Tape contains all the information the system needs to run. As you mount this and the following tapes on the tape drive, make sure they do not have write rings.

DUMPER now will try to find out what to do with the information on the tape. It will first ask "DO YOU WANT TO SUPERSEDE OLDER VERSIONS ALWAYS?". Answer with a "n", for "no". This makes sure that DUMPER will put the files from tape and the files already on the disk in the correct order and pay attention to version numbers. DUMPER will then ask "SPECIFIC USERS?". DUMPER asks this because it normally restores the files of single users whose directories are somehow lost or mutilated. Since you want to restore the all the files of every user, type "n", for "no"; and, when DUMPER wants to know: "INTO SAME DIRECTORIES?", type "y". DUMPER's final request will be: "TYPE MAG TAPE UNIT NUMBER". After you type "0", DUMPER will copy the files from the currently mounted tape into their directories. When it has finished, it will print: "MOUNT NEXT TAPE, IF ANY: TYPE C WHEN READY, N, IF NO MORE". Load the next full dump tape, making sure it does not have a write ring, and type "c", for "continue". When the mag tape unit number is requested, answer with "0". This new tape will then be read, and DUMPER will again ask if you want to go on. Continue loading the Full Dump Tapes until all have been read.

When you have finished loading the Full Dump Tapes, it is time to load the Incremental Dump Tapes. Incremental Dump Tapes are tapes made every night that contain only files altered during the preceding day. As you enter the Incremental Dump Tapes, you progressively update the files entered from the Full Dump Tapes. Enter the Incremental Dump Tapes in chronological order, beginning with the tape made right after the full dump, and ending with the tape made most recently. Use the same procedure you used to load the Full Dump Tapes: mount the tape, type "c", and enter the unit number. When you have loaded the final, i.e. the most recent, Incremental Dump Tape, you will have restored the files as well as they can be restored. At this point, answer "n", for no, to DUMPER's question about any further tapes. DUMPER then will halt, and the system will print an interrupt message followed by a period, the prompt for

MINI-EXEC. Now halt the system (with Method A of section 13.5, Halting the System) and bring it up with a disk recovery.

10.4 Errors and Recoveries

Inability to read the DLUSER tape

If you are unable to read the DLUSER tape, halt the system and start the entire procedure over again.

Inability to read the first Full Dump Tape (the tape after the DLUSER tape)

Since the DLUSER tape contains a copy of DUMPER, once you have read this tape, DUMPER is stored on the disk. If you then cannot read the second tape, that is, the first Full Dump Tape, type <CTRL-P>. (You may have to do this several times.) You will get a period, the prompt for MINI-EXEC. After you have the period, halt the system and bring it up as documented in the section "Standalone Recovery". When the system is up, you can run DUMPER from disk by typing "dumper<CR>" at the EXEC "@". Once DUMPER is running, start from step 20 in the procedure documented above. If you still cannot read the first Full Dump Tape, halt the system and start the whole process again from step 1.

11 RECOVERY FROM MEMORY PARITY ERRORS

11.1 Introduction

System XXVs run on odd parity. This means that for every word of memory, the sum of the bits turned on plus the parity bit must be odd. The system checks the parity whenever it uses stored information. If it finds a word with even parity, a parity error occurs. If the system discovers a parity error, it first tries to correct the error itself. If the error cannot be corrected, then the system automatically scans core, prints an error message listing the locations and contents of the offending addresses, and stops with a BUGHLT. Tymshare Maintenance must be called for all System XXV parity errors, as they indicate that the memory hardware may be bad.

11.2 Summary

Call Tymshare Maintenance for all parity errors.

12 RECOVERY SWITCHES

12.1 Introduction

The System XXV has four "recovery switches" that tell it how to respond to system errors and what to do when it crashes. These switches are actually four locations in the system's central memory, each controlling a particular aspect of the system's response. Every location or switch can have at least two different values. Changing the values changes what the system will do in the particular situation that the switch controls. For example, the switch controlling what the system does when it encounters a BUGCHK, a less serious error than a BUGHLT, can be set to 0 (zero) or 1 (one). When the switch has the value 0, the system ignores BUGCHKs, when it is set to 1, the system crashes when it encounters a BUGCHK. Thus, to make the system run as you wish, you simply set each switch to the appropriate value. The rest of this section will help you discover what this value may be and teach you how to set it. The first part, "Switches and Their Settings", discusses each of the four switches and what they control, and describes the effects of their different settings. The second part, "How to Change Switch Settings", explains how to set a switch to have the value you want.

12.2 Switches and Their Settings

The System XXV's four recovery switches are:

DBUGSW, which controls reponse to a BUGHLT

DCHKSW, which controls the response to BUGCHK

CDMPSW, which tells the system whether or not to take a core dump

RELDSW, which tells the system whether or not to actually begin automatic recovery

The system checks DCHKSW when it encounters a BUGCHK, a relatively minor type of error. The system then immediately does as this switch instructs it; no other switches are looked at. When the system encounters a BUGHLT, a fatal error, it checks DBUGSW. DBUGSW may then tell it to check the two remaining switches, CDMPSW and RELDSW. If DBUGSW does not instruct the system to look at CDMPSW and RELDSW, they are never checked.

The table below outlines the values each recovery switch can have and the effect of setting the switch to this value. The first column gives the name of the switch, the second column lists the possible values for this switch, and the third column describes how the system will act when the switch has this value. The "normal" value for each switch is marked with a stars (*). When all switches have their normal values, the system prints messages at BUGCHKs, and crashes at BUGHLTs. After the system crashes, it will wait for an operator to bring it up. If you want the system to recover automatically, rather than wait for operator's instructions, simply change the setting of DBUGSW from 1 to 0. See the next section for instructions.

Switch	Value (*=normal)	Effects
DBUGSW	0	Stop at BUGHLTs, check CDMPSW and RELDSW and do what they say. (For Automatic Recovery)
	1*	Stop at BUGHLTs, don't check CDMPSW and RELDSW, go into EDDT, and wait for an operator to begin recovery. (For Disk or Tape Recovery)
	2	Stop at BUGHLTs, don't check CDMPSW and RELDSW, go into EDDT. Don't run system-checking programs and come up shut. (For Standalone Recovery)
DCHKSW	0*	Don't stop at BUGCHKs, print error message and continue.
	1	Stop at BUGCHKs, print error message, go into EDDT, and wait for an operator to begin recovery.
CDMPSW	0	Don't take core dump before beginning recovery.
	1*	Take core dump before beginning recovery.
RELDSW	0	Don't begin automatic recovery after crashing.
	1*	Begin automatic recovery after crashing.

12.3 Switch Descriptions

DBUGSW: DBUGSW, located at memory location 76, is the switch the system checks when it encounters a BUGHLT while running. A BUGHLT is a serious system error. The system must crash when a BUGHLT occurs; this switch determines what the system does after the crash. DBUGSW can be set to 0, 1, or 2. A 0 at DBUGSW is the setting for automatic recovery. It instructs the system to check the switches CDMPSW and RELDSW and do as they say. CDMPSW will tell it whether a core dump should be taken; RELDSW will tell the system whether to actually start the recovery. (See below and the section on automatic recovery for details.) A 1 at DBUGSW is the standard setting. With this setting, upon encountering a BUGHLT, the system stops where it is, prints out a BUGHLT message, goes into EDDT, and waits for instructions from the operator's terminal. You can then begin whatever recovery procedure is appropriate. A 2 at DBUGSW has the same effect as a 1, and, in addition, after a recovery procedure is started, causes the system to come up standalone. The system will come up without running CHECKDISK and the system jobs and, after it is up, only the person at the operator's terminal is allowed access. (To do a standalone recovery, you put a 2 in DBUGSW before the system comes all the way up.)

DCHKSW: DCHKSW, located at memory address 77, is the only switch checked when the system encounters a BUGCHK; no other switches are consulted. DCHKSW can be set to 0 or 1. When DCHKSW is 0, the system will print out a BUGCHK message and then continue running. When DCHKSW is 1, the system will print out a BUGCHK message and then stop, go into EDDT, and wait for further instructions about how to come up. Since BUGCHKs are not serious errors, the usual setting for DCHKSW is 0.

CDMPSW: CDMPSW is located at memory address 100. It is checked only when a BUGHLT occurs and the system finds that DBUGSW is set to zero, the setting for automatic recovery. CDMPSW tells the system whether or not to make a copy of the contents of central memory before beginning to come up. The process of copying the contents of memory is called "taking a core dump". If CDMPSW switch is set to 0, then the system will not take a core dump. It will simply check RELDSW to see if it really should come up automatically. If CDMPSW is set to 1, before checking RELDSW, the system will copy system first 512 pages of memory into a file called <SYSTEM>CORDMP.LOW and the second 512 pages into a file called <SYSTEM>CORDMP.HGH. Since systems programmers may need to look at the contents of the memory to investigate the crash, CDMPSW is generally set to 1.

RELDSW: RELDSW is located at memory address 101. It, like CDMPSW, is checked only after a BUGHLT occurs and DBUGSW set to zero, the setting for automatic recovery. RELDSW tells the system whether or not it should actually begin this automatic recovery. A 0 in RELDSW tells the system not to recover automatically; the system will then wait for instructions from the operator's terminal just as if DBUGSW were set to 1. A 1 tells the system "yes, do begin to come up automatically". Because when DBUGSW is 0 you usually do want the system to recover automatically, the normal setting of RELDSW is 1.

12.4 How to Change Switch Settings

Introduction

This section tells you how to change the values of the System XXV's four recovery switches. To do this the system must be running correctly and you must be able to enable. To learn the names of the recovery switches, their values, and what they mean, see the previous section.

Summary

1) Check your prompt. If it is an exclamation mark (!), you are enabled. If it is not, type "ena<CR>" at the EXEC "@".

2) At the "!" prompt, type "mddt<CR>".

3) Type "switchname/", where switchname stands for the name of the switch you want to change.

4) The system will give you the current value of the switch.

5) Type "switchvalue<CR>", where switchvalue stands for the new value the switch should have: 0, 1, or 2.

6) To check the new value, type "switchname/" again. The system should show you the new value.

7) Type "<CTRL-C>"; you should return to EXEC and get the "!" prompt.

Discussion

To change the values of the recovery switches, you need to work in MDDT an interactive language for debugging. It is part of the resident monitor and is used to change and manipulate it. To enter MDDT, you first need to make sure you are enabled. Check your prompt: if it is an exclamation mark (!), you are enabled. If it is anything else, type "ena<CR>" at EXEC "@" prompt. After you are sure you are enabled, enter MDDT by typing "mddt<CR>". The system will print "mddt", to show you have entered, and then do nothing more. Like EDDT, MDDT has no herald; as you use it, it simply waits for you to tell it something and then reacts.

When you are in MDDT, to go to the switch you want to change and look at its current value, type the switch name followed by a slash (/), for example, "dchksw/". This command has two parts. The first part, "dchksw", is the name of the address that contains the recovery switch value. The "/" means "print". Thus, "dchksw/" instructs the system to show you the contents of the location "dchksw".

Once MDDT has shown you the value of the switch, it waits at this location to see if you want to do anything else. If you decide you do not want to change this switch, simply type a carriage return. To enter a different value in this address, type the value you want followed by a carriage return. The number you type will immediately become the new value of the switch. To make sure that you entered the value you wanted, again type the switch name followed by a slash. If the value is correct, simply type a carriage return. This means you are finished working with this address. If it is not correct, type the correct value and then a carriage return.

After you have changed as many of the four recovery switches as you wish, you are ready to leave MDDT. To do this, type <CTRL-C> and you will returned to EXEC.

13 RELATED PROCEDURES

13.1 Introduction

This section describes the following procedures.

Correcting Problems Found by CHECKDISK, section 13.2

Running CHECKDISK Yourself from EXEC, section 13.3

Deleting and Expunging Files, section 13.4

Halting the System, section 13.5

Changing the Date and Time, section 13.6

Connecting to and Disconnecting from the Micronode TYMBASE, section 13.7

When a recovery process requires that one of these procedures be used, you will be referred here. If you find that you never have to use any of them, do not be alarmed. This is a sign of success. These procedures are used only when something goes wrong -- when, for example, CHECKDISK finds file problems that must be corrected, you need to halt the system, or the system some how comes up with the wrong date and time.

13.2 Correcting Problems Found by CHECKDISK

Introduction

CHECKDISK is a program the system uses to check the file system before it comes all the way up and opens itself to users. If it finds any problems, the system states, "August not in operation" and stops to waits for them to be corrected with the procedure documented below. Once this is done, halt the system as documented later in "Related Procedures", and then bring it up again with disk recovery. This section deals only with recovery from file errors detected by CHECKDISK. It assumes that CHECKDISK has been run automatically. To learn how to run CHECKDISK manually, see 13.3, Running CHECKDISK Yourself from EXEC.

Summary

CHECKDISK checks the files for Illegal Disk Addresses (IDAs), Multiple Disk Addresses (MDAs) and Bit Table Errors (BTEs). If it finds any of these, it lists the files involved and their errors. To correct the problems found by CHECKDISK do the following:

1) If only one file has errors, delete and expunge that file. Be sure to type the entire file name, including all extensions; do not use <ESC> to fill out names. The

process of deleting and expunging files is described in section 13.4, Deleting and Expunging Files.

2) If more than one file is involved, delete and expunge all files with IDAs. Do not delete the files with MDAs at this point.

3) Halt the system with Method A of section 13.5, Halting the System. Then bring it up again with disk recovery. If CHECKDISK again finds files with IDAs, repeat this procedure.

4) Once no files have IDAs, if one or more files have MDAs, delete and expunge the file with the largest number of MDAs, halt the system, and bring it up with disk recovery. Do this three times. If you then still have more than 20 files with MDAs, call an operating systems programmer.

NOTE: Keep a list of the files you delete and expunge, and restore them after the system comes up. Always send messages to all users whose files have been deleted and restored.

Discussion

CHECKDISK can detect three types of errors: Bit Table Errors (BTEs), Illegal Disk Addresses (IDAs), and Multiple Disk Addresses (MDAs). CHECKDISK can correct BTEs without assistance. It cannot, however, correct IDAs or MDAs. These two errors are what are known as Page Table Errors. They occur when the system's file map, stored in what is called a "page table", is incorrect. AUGUST memory is divided into units called pages, each consisting of 512 words. File storage is allotted by pages, and one page is the smallest unit of storage that can be transferred from disk to core. A page table is like a table of contents for the disk storage. For each file, it records the addresses of all the pages allocated to that file.) An IDA means there is a disk address that is garbage. An MDA means the system has assigned the same part of the disk to two or more files. If these errors are allowed to go uncorrected, they can destroy the file system.

The remedy for problems detected by CHECKDISK is to delete the files that really do have bad storage addresses. If there is only one file with bad addresses, there is not a serious problem; simply delete that file. If more than one file is afflicted, begin by deleting all files with IDAs. IDAs are a frequent cause of MDAs. Often, when the system follows an IDA, it will find other things that it can interpret as more addresses, but which are not. These phony addresses may duplicate the real addresses of pages belonging to other files, thus causing MDAs. After IDAs are taken care of, files with MDAs may remain. Deleting the single file with the most MDAs may take care of the problem.

Once you have deleted the appropriate files, you should halt the system and bring it back up with disk recovery. If CHECKDISK again finds errors, you must again correct them, bring the system down, and then back up. If the fourth time CHECKDISK is run it still finds errors, notify an OAD systems programmer. Remember that once the system does come up successfully, the owners of the files must be notified about all files deleted.

13.3 Running CHECKDISK Yourself from EXEC

Introduction

CHECKDISK is a program that checks the address system and page allocation of the disk. CHECKDISK usually runs automatically as the system comes up. However, there may be occasions, for example after a standalone recovery, when you need to run CHECKDISK yourself. This section documents that procedure. What CHECKDISK does is explained in section 13.2, Correcting Errors Found by CHECKDISK.

Summary

1) At the EXEC "@", type "<system>checkdisk<ESC><CR>".

2) When CHECKDISK asks, "Do you want to run in multiple fork mode?", type "Y", for yes". All answers to CHECKDISK's question must be capitalized. Do not type more than a single letter, since CHECKDISK will take any excess letters as answers to following questions.

3) When CHECKDISK asks, "Do you want to run backwards?", type "N", for no.

4) To the question: "Rebuild the bit table?", type "N".

5) To the question: "Scan for disk addresses?", type "N".

6) CHECKDISK will now check the disk for bad files. For instructions on how to deal with bad files, see section 13.2.

Discussion

You invoke CHECKDISK by typing "<system>checkdisk<ESC><CR>". Once CHECKDISK is loaded, it will ask you a series of questions to determine how the disk should be checked and how much information about its status you want to get and store. When CHECKDISK runs automatically, these options are already specified; however, when you run CHECKDISK manually, you must specify them yourself.

The first question CHECKDISK will ask is, "Do you want to run in multiple fork mode?". This means, "Do you want to fire up a different fork of EXEC to run CHECKDISK separately for each disk?". The standard answer here is "Y", for "yes", since running CHECKDISK simultaneously on all the disks is faster than going through the disks one at a time. Note that you should type only the first letter of your answers to CHECKDISK's questions and that this letter must be capitalized. This is important. CHECKDISK cannot recognize lowercase letters. Moreover, if you type more than one letter, CHECKDISK will read the second and following letters as answers to later questions. This can cause a lot of problems.

Once CHECKDISK knows how many forks you want, it will ask if you want it to run backwards and check the disk from the last file to the first. The standard answer here is "N", for "no". CHECKDISK will then ask, "Rebuild the bit table?". Again, answer "N". The bit table is used to keep track of which pages on the disk have been used and which are free. However, the bit table is not updated after every process that frees pages in the disk. When you delete a bad file, for instance, the bit table will still mark as taken the pages that you have freed. Thus, it is a good idea to rebuild the bit table occasionally; otherwise, the whole disk could eventually be marked as taken, when parts of it were actually free. But rebuilding the bit table is too time consuming a process to do when you are bringing the system up from a presumably unscheduled crash.

The CHECKDISK will then ask if it should scan for disk addresses. CHECKDISK wants to know if you want the names of the files that are actually associated with all the bad disk addresses. Since this information is useful only to systems programmers, answer "N".

CHECKDISK will now check the disk and print a list of bad files and their errors. For instructions on how to deal with bad files, see section 13.2, Correcting Problems Found by CHECKDISK.

13.4 Deleting and Expunging Files

Summary

1) If your prompt is not an "!", type "ena<CR>" at the EXEC "@".

2) Connect to the directory that contains the file by typing "cd<SP>directoryname<CR>", where directoryname stands for the name of the directory you need.

3) Type "del<SP>filename<CR>". Make sure you type the entire file name including extensions. Do not use <ESC> to fill out the name -- the file may not be recognized correctly. Procede all unusual characters in the file name, for example @, with <CTRL-V>.

4) If the system tells you the file is perpetual, type "not<SP>perp<SP>filename" and then delete the file.

5) Type "exp<ESC><CR>".

6) Remember to connect back to directory "oper" when you finish deleting files by typing "cd<SP>oper<CR>".

NOTE: Always send a message to any user whose files you have deleted.

13.5 Halting the System

Introduction

There are two ways of halting the System XXV both halt the system immediately and for no designated length of time. They are used when you have encountered some problem during a crash recovery and want to bring the system down so that you can start again in the normal way. Method A is designed to halt a system that is running and will respond to commands given from the operator's terminal. This is probably the procedure you will most often use. You would use Method A, for example, to halt the system after fixing file problems found by CHECKDISK. Whenever Method A does not work because the system is hung or for some reason does not respond to the operator's terminal, you should resort to Method B. After halting the system with either of these methods, you may use whatever recovery procedure seems appropriate bring it back up.

Method A. Halting a Running System from the Operator's Terminal

This procedure has two steps. First, you need to get into MINI-EXEC, and then you need to halt the system. If you are already in MINI-EXEC when you decide to halt the system, start this procedure on step 4; if you are not, start at step 1. The way to tell if you are in MINI-EXEC is to look at the prompt. If it is a period (.), you are in MINI-EXEC; if it is anything else, you are not.

1) If your prompt is not an "!", type "ena<CR>" at the EXEC "@".

2) Type "quit<CR>"

3) When the system asks, "Do you really want to go into AUGUST monitor? (Confirm)", type "<CR>".

5) At the period (.) prompt, type "h".

6) The system will echo, "HALT TENEX".

7) Type ".".

8) The system will halt.

Method B. Halting the System from the Control Panel

1) Put address switch 31 on (up).

2) Put data switch 2 on.

3) Put CONSOLE DEPOSIT THIS on.

4) Put data switch 2 off.

5) Put data switch 0 on.

6) Put CONSOLE DEPOSIT THIS on.

7) When activity (the flickering of the lights, etc.) stops; the system has halted.

13.6 Changing the Date and Time

Summary

1) If your prompt is not an "!", type "ena<CR>" at the EXEC "@".

2) When you see the prompt "!", type "<CTRL-E>set<SP>DD-MON-YY<SP>HH:MM<CR>"; that is, two numbers for the day, a dash, the first three letters of the month, a dash, and then two numbers for the year. Follow this with a space, then give the time on 24 hour basis, and end with <CR> You must type the entire date and time to reset any part of it.

3) Type a confirming <CR>.

4) At the EXEC "@", type "day<CR>" to check the new date and time.

13.7 Connecting to and Disconnecting from the Micronode TYMBASE

Introduction

The two sets of procedures documented below allow you control whether or not the system will communicate the micronode TYMBASE. The ability to control the system's interaction with the micronode is useful when some micronode error is causing system problems or when the micronode is down and the system should not try to connect to it. Each set of procedures allows you to do the same things: Turn the micronode connection off, which causes the system to ignore the micronode; and turn the micronode connection on, which tells the system to synchronize with the micronode. Method A and Method B differ in where you give the controling commands. Method A uses commands given in EXEC. In Method B, in the other hand, you use EDDT. Method B should be used only during crash recovery, when you must control how the system interacts with the micronode as it comes up. In all other cases, control interaction from the EXEC with Method A.

Method A: Controling interaction from EXEC

To turn off the micronode connection

1) At the EXEC "@", type "<CTRL-E>tymnet<SP>off<CR>".

To turn on the micronode connection

1) At the EXEC "@", type "<CTRL-E>tymnet<SP>on<CR>".

Method B: Controling interaction from EDDT

To turn off the micronode connection

1) In EDDT, type "tymflg/".

2) After the system prints a value, type "0<CR>".

To turn on the micronode connection

1) In EDDT, type "tymflg/".

2) After the system prints a value, type "-1<CR>".

APPENDIX

This section is designed for quick reference: use it when you need to look up a certain step in a procedure, cannot remember exactly what order to do things, and so forth. No explanations of when to use these procedures, discussions of what they do, or suggestions about what to do if things go wrong are included here. For this type of information go to the first part of this manual where the procedures outlined in this section are discussed in greater length. All sections references in this appendix are also to earlier sections in this document.

WHAT TO DO IF THE SYSTEM IS HUNG

1) Put address switch 31 on (up).

2) Put data switch 2 on.

3) Put CONSOLE DEPOSIT THIS momentarily on.

4) Put data switch 0 on.

5) Put CONSOLE DEPOSIT THIS momentarily on.

6) Wait until activity (the flickering of the lights, etc.) stops.

7) Bring the system up with the disk recovery procedure.

DISK RECOVERY

1) Record the BUGHLT number and error lights.

2) Type "dskrld<ESC>g". The reponse should be "reloading from disk". If you never get this message, begin a tape recovery.

3) When the system says, "BOOT FROM DISK PACK # [CR FOR ANY]", type <CR>.

4) When the operator's terminal says "EDDT", type "start<ESC>g".

5) If CHECKDISK runs successfully, the system will announce "August in operation" and ask for the date and time. Enter these in the form DD-MON-YY<SP>HH:MM and follow with <CR>.

6) At the @ prompt, log in by typing "oper<SP>password<SP><CR>", where password stands for your password.

7) Type "ena<CR>".

8) Type "ref<SP>a<CR>".

AUTOMATIC RECOVERY

If a System XXV set for automatic recovery comes up successfully, you do not need to do anything until you log in as an operator.

1) At the @ prompt, log in by typing "oper<SP>password<SP><CR>", where password stands for your password.

2) Type "ena<CR>".

3) Type "ref<SP>a<CR>".

TAPE RECOVERY

1) Record the BUGHLT number and error lights.

2) If the power has gone off, you must reload the microcode as documented in steps 3 through 12. If the power has not gone off, skip to step 13.

3) Mount the microcode tape on the tape drive.

4) Put all switches on the control panel off (down).

5) Put address switch 32 on (up).

6) Put MICRO PROCESSOR STOP on.

7) Put MICRO PROCESSOR MIPC on.

8) Put MICRO PROCESSOR CLR momentarily on.

9) Put MICRO PROCESSOR CONT momentarily on.

10) Put MICRO PROCESSOR MIPC off.

11) Put MICRO PROCESSOR STOP off.

12) Put MICRO PROCESSOR CONT momentarily on. The tape should spin and then stop. Remove the microcode tape from the tape drive.

13) You are now ready to read in the new monitor. Mount the monitor tape on the tape drive. [Start here if you do not want to load the microcode.]

14) Put all switches on the control panel off.

15) Put address switches 24 and 26 on.

16) Put MICRO PROCESSOR STOP on.

17) Put MICRO PROCESSOR MIPC on.

18) Put MICRO PROCESSOR CLR momentarily on.

19) Put MICRO PROCESSOR CONT momentarily on.

20) Put MICRO PROCESSOR MIPC off.

21) Put MICRO PROCESSOR STOP off.

22) Momentarily put MICRO PROCESSOR CONT on. The tape should spin and then stop.

23) Put address switches 24 and 26 off.

24) Put address switches 29 and 30 on.

25) Momentarily put CONSOLE START on twice.

26) When the operator's terminal says "EDDT", type "start<ESC>g".

27) Remove the monitor tape from the tape drive.

28) After the system reports the size of the memory, put MI PAR ERR STOP and MEM PAR ERR STOP on.

29) If CHECKDISK runs successfully, the system will announce "August in operation" and ask for the date and time. Enter these in the form DD-MON-YY<SP>HH:MM and follow with <CR>.

30) At the @ prompt, log in by typing "oper<SP>password<SP><CR>", where password stands for your password.

31) Type "ena<CR>".

32) Type "ref<SP>a<CR>".

STANDALONE RECOVERY

1) Record the BUGHLT number and error lights.

2) Follow the procedure for tape recovery (section 8) from step 13 through step 25. If you suspect there has been a power failure, do step 3 through 25 of the tape recovery procedure.

3) When the operator's terminal says "EDDT", type "dbugsw/".

4) Type "2<CR>".

5) Type "start<ESC>g". The monitor tape should spin.

6) The system will request the date and time. Enter these in the form DD-MON-YY<SP>HH:MM and follow with <CR>.

7) You will automatically be logged in as "system", but not enabled.

DISK REBUILD STRATEGY OR TOTAL CATASTROPHE

WARNING: Never attempt this without specific instructions from a manager or an operating systems programmer.

1) Check with Tymshare Maintenance to make sure the hardware is good.

2) Follow the tape recovery procedure from steps 13 through 25. If there has been a power failure, do steps 3 through 25.

3) When the operator's terminal says "EDDT", type "dbugsw/".

4) Type "2<CR>".

5) Type "syslod<ESC>g".

6) When the system asks, "Do you really want to clobber the disk by reinitializing?", type "y<CR>".

7) Load the DLUSER tape on the tape drive.

8) Type "l", for "load". When the system asks, "Load from magtape MTAN:", type "mta0:<CR>" ("0" here is zero). Confirm this with another <CR>.

9) When the system asks, "File Number?", type "0<CR>".

10) When the system has read the DLUSER file and prompts you with a period (.), type "s.".

11) When you see "Interrupt at nnn", where nnn is some number, type "l", for "load", then specify "mta0:<CR>", and confirm with another <CR>.

12) When the system asks, "File Number?", type "1<CR>".

13) At the period prompt, type "s.".

14) Mount the first Full Dump Tape. Make sure you load the Full Dump Tapes in numerical order.

15) To DUMPER's question, "DUMP, LOAD, CHECK, OR SINGLE?", type "l", for "load".

16) For the second question, "DO YOU WISH TO SUPERSEDE OLDER VERSIONS ALWAYS?", type "n".

17) For the question, "SPECIFIC USERS?", type "n".

18) To answer, "INTO SAME DIRECTORIES?", type "y".

19) When requested, "TYPE MAG TAPE UNIT NUMBER", type "0" (zero).

20) DUMPER will now read the tape; when it is finished, it will print, "MOUNT NEXT TAPE, IF ANY. TYPE C, WHEN READY, N, IF NO MORE". Mount the next tape and type "c".

21) When DUMPER asks for the mag tape unit number; type "0".

22) Continue mounting and loading Full Dump Tapes, typing "c" to continue and then giving 0 (zero) for mag tape unit number, until all the tapes have been read.

23) When you have finished loading the Full Dump Tapes, begin loading the Incremental Dump Tapes. Be sure that you load the Incremental Dump Tapes in chronological order, beginning with the one made right after the Full Dump and ending with the most recent.

24) After the system has read the last Incremental Dump Tape, when DUMPER says, "MOUNT NEXT TAPE, IF ANY. TYPE C, WHEN READY, N, IF NO MORE", type "n".

25) When you see an interrupt message followed by a period, halt the system, and begin a disk recovery. (To halt the system use Method A of section 13.5, Halting the System.)

CHANGING RECOVERY SWITCH SETTINGS

1) If your prompt is not an "!", type "ena<CR>" at the EXEC "@".

2) At the "!" prompt, type "mddt<CR>".

3) Type "switchname/", where switchname stands for the name of the switch you want to change.

4) Type "switchvalue<CR>", where switchvalue stands for the new value the switch should have: 0, 1, or 2.

5) Type "<CTRL-C>" to return to EXEC.

CORRECTING PROBLEMS FOUND BY CHECKDISK

1) If only one file has errors, delete and expunge that file. Be sure to type the entire file name.

2) If more than one file is involved, delete and expunge all files with IDAs. Do not delete the files with MDAs at this point.

3) Halt the system with Method A of section 13.5 and bring it up with disk recovery. If CHECKDISK again finds files with IDAs, repeat this procedure.

4) Once no files have IDAs, if one or more files have MDAs, delete and expunge the file with the largest number of MDAs, halt the system, and bring it up with disk recovery. Do this three times. If you then still have more than 20 files with MDAs, call an operating systems programmer.

NOTE: Keep a list of the files you delete and expunge, and restore them after the system comes up. Always send messages to all users whose files have been deleted and restored.

RUNNING CHECKDISK YOURSELF FROM EXEC

1) At the EXEC "@", type "<system>checkdisk<ESC><CR>".

2) When CHECKDISK asks, "Do you want to run in multiple fork mode?", type "Y", for yes". All answers to CHECKDISK's question must be capitalized and one letter.

3) To the question, "Do you want to run backwards?", type "N", for no.

4) To the question: "Rebuild the bit table?", type "N".

5) To the question: "Scan for disk addresses?", type "N".

6) CHECKDISK will check the disk for bad files. For instructions on how to deal with bad files, see section 13.2.

DELETING AND EXPUNGING FILES

1) If your prompt is not an "!", type "ena<CR>" at the EXEC "@".

2) Connect to the directory that contains the file by typing "cd<SP>directoryname<CR>", where directoryname stands for the name of the directory you need.

3) Type "del<SP>filename<CR>". Make sure you type the entire file name including extensions. Do not use <ESC> to fill out the name -- the file may not be recognized correctly. Procede all unusual characters in the file name, for example @, with <CTRL-V>.

4) If the system tells you the file is perpetual, type "not<SP>perp<SP>filename" and then delete the file.

5) Type "exp<ESC><CR>".

6) When you finish deleting files, type "cd<SP>oper<CR>".

NOTE: Always send a message to any user whose files you have deleted.

HALTING THE SYSTEM

Method A. Halting a Running System from the Operator's Terminal

This procedure has two steps. First, you need to get into MINI-EXEC, and then you need to halt the system. If you are already in MINI-EXEC when you decide to halt the system, start this procedure on step 4; if you are not, start at step 1. The way to tell if you are in MINI-EXEC is to look at the prompt. If it is a period (.), you are in MINI-EXEC; if it is anything else, you are not.

1) If your prompt is not an "!", type "ena<CR>" at the EXEC "@".

2) Type "quit<CR>"

3) When the system asks, "Do you really want to go into AUGUST monitor? (Confirm)", type "<CR>".

5) At the period (.) prompt, type "h".

6) The system will echo, "HALT TENEX".

7) Type ".".

8) The system will halt.

Method B. Halting the System from the Control Panel

1) Put address switch 31 on (up).

2) Put data switch 2 on.

3) Put CONSOLE DEPOSIT THIS on.

4) Put data switch 0 on.

5) Put CONSOLE DEPOSIT THIS on.

6) When activity (the flickering of the lights, etc.) stops; the system has halted.

CHANGING THE DATE AND TIME

1) If your prompt is not an "!", type "ena<CR>" at the EXEC "@".

2) When you see the prompt "!", type "<CTRL-E>set<SP>DD-MON-YY<SP>HH:MM<CR>"; that is, two numbers for the day, a dash, the first three letters of the month, a dash, and then two numbers for the year. Follow this with a space and then give the time on 24 hour basis. You must type the entire date and time to reset any part of it.

3) Type a confirming <CR>.

4) At the EXEC "@", type "day<CR>" to check the new date and time.

CONNECTING TO AND DISCONNECTING FROM MICRONODE TYMBASE

Method A: Controling interaction from EXEC

To turn off the micronode connection

1) At the EXEC "@", type "<CTRL-E>tymnet<SP>off<CR>".

To turn on the micronode connection

1) At the EXEC "@", type "<CTRL-E>tymnet<SP>on<CR>".

Method B: Controling interaction from EDDT

To turn off the micronode connection

1) In EDDT, type "tymflg/".

2) After the system prints a value, type "0<CR>".

To turn on the micronode connection

1) In EDDT, type "tymflg/".

2) After the system prints a value, type "-1<CR>".

```
+-----------------------------+
|          TYMSHARE           |
| Distributed Systems Section |
+-----------------------------+
```

Hardware Configuration Specification PDP-11 Based TYMBASE

Author...J.M.Stammers.
Date.......Sep 18 1980
Revision...........243
Update....1/13/81;DRE