RSD-MEMO-2-83

System Generation Using RSX-11M With Microbot Driver and System Maintenance

Laszlo M. Hideg

May 1983

Center for Robotics and Integrated Manufacturing

ROBOT SYSTEMS DIVISION
COLLEGE ON ENGINEERING
THE UNIVERSITY OF MICHIGAN
ANN ARBOR, MICHIGAN 48109

Preface

This document is intended as a guide and supplement to a typical SYSTEM GENERATION on a Digital Equipment Corporation PDP-11 computer. The operating system utilized is the RSX-11M Version 3.2 (copyright 1979 by DEC). This document assumes some knowledge of RSX-11M. Included in this document are the steps necessary to generate an I/O driver for the Minimover-5 table top robots (copyright by Microbot Inc.) The final system configuration is intended to accommodate eight CRT terminals, eight microbots and other peripherals to run a robotics instructional lab a the University of Michigan.

The major sections of this document are divided into Preparation (disk backups etc.), System Generation Execution (Sysgen), Post Generation Cleanup (Microbot Driver etc.) Sysgem Generation with the RSX-11M operating system is a tedious and lengthy process for the novice. Hopefully, this document will assist in the process.

Laszlo M. Hideg
University of Michigan
February 1983.

Symbols used in this document

<CR> Carriage Return, or Return, or Enter (Includes a linefeed)

( ) Parenthesis, usually in the system generation, contains information not critical to answer the question.

[ ] Square Brackets, usually in the system generation, contains allowed answers or ranges of answers.

CTRL Control key often in lower left hand corner of the keyboard. It is simultaneously pressed with the key that is listed after the CTRL.

* The wild card where it represents most any value or character string that is applicable wherever it is placed.

References

1) DEC Reference Manuals (Blue Binders)

MCR Operations Vol.2

Executive Reference Vol.3B

Utilities Vol.4

I/O Drivers Vol.5A

2) DEC Hardware Manuals:

Microcomputers and Memories 1982

Memories and Peripherals 1978-79

3) Omsi Pascal Manual-Oregon Software

Version 1.2 for RSX

4) RL01 Disk Subsystem User's Guide

by DEC

1. Preparation for System Generation

This document assumes that a working system already exists on the PDP-ll computer. Also, no AUTOPATCH orientation is included. The AUTOPATCH need be done only once for a computer and not done again. Refer to your cover letters or call your DEC representative for more information.

1.1. Baseline System

This document assumes a stand alone system. The distribution disk which contains the components of a system is the Baseline Disk. For a distribution kit on RL01 disks the following are the volume names of the disks.

Volume Name RL01	Description
RSXM26	Contains files that Constitute the baseline system.
MAPSRC	Contains files to generate a mapped system. Particularly, tasks for a mapped system are under the User Identification Code (UIC) [1,54].
UNMSRC	Contains files to generate on unmapped system. The unmapped tasks are under UIC [1,50]
RLUTIL	Contains various added utilities such as the Backup and Restore Utility (BRU), and Help text file information.

The Baseline system is a simple RSX-11M operating system with the ability to generate a more complex operating system as specified by the designer. First of all a backup of the Baseline disk is 'necessary. During a system generation, many files are created, deleted, and installed, thus the original form of the Baseline System is lost. Here are step by step instructions with explanations for the backup.

1) Assuming that the designer is always at the system console, hit the BREAK key to stop the system. Remove the old system disk and other disks by: First, be sure the WRITE PROTECT buttons are out (indicator lights are off). Spin down the disks by pressing the LOAD buttons. Immediately the DRIVE N indicators are off. N is for drive number. Wait for the yellow LOAD lights to come on and that a distinct click is heard. The click is the solenoid in the disk drive that will now allow opening the top of the drive and removal of the disks. Third, pull out one drive at a time. At the far right side is the top release mechanism. Squeeze it and raise the top like the hood of a car and latch it open. You will see a large round plastic disk. This is the bottom half of the disk pack. Remove it. You will now see the top half of the disk pack. There may be more than one type of top half thus more than one way to

remove it. Some packs require the sliding of a switch on the handle then raising the handle vertically. Others just require the raising of the handle. In either case there will be a click which disengages the disk from the drive. Fourth, carefully lift the disk vertically out and immediately place it in the bottom half of the disk pack. Compress the handle.

2) Reverse the operation to install the Baseline disk in DRIVE 0 and the extra disk for the copy in DRIVE 1. To boot in the new Baseline system type:

```
773000G
```

This is the starting address for the rebooting of the system. It will most likely be different for another computer. A set of instructions will be printed out along with a request for date, time and terminal width. We will be concerned with these later, just answer with carriage returns <CR> for now. At the end of the greeting the system is in the Monitor Console Routine (MCR) with the prompt > (greater than sign).

3) The next few instructions will install the Disk Save and Compress Utility for 800 Bits Per Inch (BPI) type storage devices. The baseline in Drive 0 will be copied to the disk in Drive 1. All original information on the DRIVE 1 disk will be lost. After the copy, hit the break key, remove the DRIVE 1 disk and reboot the system.

```
>   (MCR level-Monitor Console Routine)
>INS $BOO  <CR>
>BOO [1,51] DSCS8  <CR>
RSX11S V2.2 BL26 DISK SAVE AND COMPRESS UTILITY V3.2

DSCS8>DL1:/VE=DL0:/BL=12  <CR>
DSC -- 45 START VfY PASS

DSCS8>  (BREAK KEY)
001274
@
```

4) In the reboot of the system, enter the present date and time. The terminal width of 132 characters is good if you have a DEC WRITER III (LA120) as your console.

5) Since the system will use a large amount of memory, a mapped system will be used. On some Baseline disks, there may be material for unmapped systems under UIC[1,50]. For a successful system generation, delete these files.

```
>INS $PIP  <CR>  (Install File Handler)
>PIP [1,50]/TB  <CR>  (see how much room occupied)

STORAGE USED/ALLOCATED FOR DIRECTORY DL0:[1,50]
DATE TIME

TOTAL OF 2279./2279. BLOCKS IN64. FILES
```

```
>SET /UIC=[1,50] <CR> (go to unmapped directory)
>PIP *.*;*/DE <CR> (delete unmapped files)
>
```

6) The bootable system of the Baseline disk is small. To obtain the larger system with which to do the system generation type the following:

```
>BOO [1,54]RSX11M <CR>
```

Instructions will be printed and also two questions on date, time and the line width of the designers console. After the boot, the system is in MCR and under UIC [200,200] this contains the System Generation Command Files SYSGEN and SYSGEN 2.

1.2. Mapped System

The intended system generation will support up to four RL01 single density hard disk drives., two double density RX02 floppy disk drives, ten terminals (nine CRT, one teletype) and eight MICROBOTS for a robotics instructional lab. To accommodate such a load, the system will have to be quite large and therefore a mapped system. At this point, if desired, with Baseline disk in Drive 0, place MAPSRC into Drive 1 and spin up the disk. You must put in the disk at least before the actual System Generation is executed. The desired processor will be an 11/23 which has a maximum of 124.K blocks (. means base 10, K means thousand). The 11/23 will support the Pascal Compiler and the MICROBOT DRIVER.

1.3. Indirect File Processor Installation

At this point the MCR prompt should appear. Check to see which tasks are available by typing:

```
>TAS <CR>
```

The installed tasks are displayed. Notice that on all the system boots so far, there is a message that suggests installing the BIGINDirect file processor for a successful SYSGEN (SYStem GENeration). To facilitate and reduce the time to do a SYSGEN install the BIGMACro assembler, the BIGTKB task builder also.

```
>INS $ BIG MAC/INC=40000<CR>
INS--PARTITION PAR 14K NOT IN SYSTEM, DEFAULTING TO GEN
>INS $ BIGTKB/INC=40000<CR>
INS--PARTITION PAR 14K NOT IN SYSTEM, DEFAULTING TO GEN
>REM...AT. <CR>
>INS $ BIGIND/PRI=10. <CR>
INS--WARNING-PRIVILEDGED TASK OVERMAPS I/O PAGE
>
```

You are now ready to execute the SYSGEN and SYSGEN2 command files for your new system.

1.4. SYSGEN Command File

Presently the terminal (or console) is under UIC [200,200]. Of particular interest are two files listed below with the commands to see them.

```
>INS $PIP
>PIP/LI
.
.
.
.
SYSGEN.CMD;1    70.  ....
SYSGEN2.CMD;1   74.  ....
.
.
>
```

To begin SYSGEN then SYSGEN2 execution:

```
.
.
>@SYSGEN
.
.
: (System Generation)
.
.
.
.
> (New system in place)
```

The BIGINDirect file processor is necessary to handle the large amount of compilation, file handling, symbol table usage involved with a SYSGEN.

1.5. Possible Problems

I) SYSGENS take up a large part of the room available on a RL01 disk. If a mapped system is being generated, the deletion of unmapped files in UIC [1,50] is absolutely necessary. If not done the SYSGEN will crash, probably at some point in SYSGEN2 when the final system is being assembled together.

II) To be safe, a copy of the baseline system should be made, especially if the present SYSGEN fails or future SYSGENs will be done. The disk save and compress utilities must be intact. If not, find a copy from another Baseline disk for RSX-11M or contact your DEC representative.

III) The BIGINDirect file processor must be installed else, the SYSGEN will crash very early. Again, if the copy of BIGIND is faulty find another or contact your DEC representative. You can test BIGIND by creating a small command file and executing it. If the system crashes, BIGIND is faulty. For example after BIGIND is installed, try the following:

```
>INS $EDI <CR>
>EDI TEST.CMD <CR>
CREATING NEW FILE
TIME <CR>
PIP/FR <CR>
*EX <CR>
> @ TEST
(the time and free blacks on DLO are displayed)
>
```

If the system crashes, there will be a register dump and a prompt of the Executive Debugging Tool (@). It is better to find out before SYSGEN since toward the end of SYSGEN or SYSGEN2 the original bootable system is eliminated. There is recovery on the current Baseline disk if there is a SYSGEN Crash, but finding out early is preferred.

2. System Generation Execution

This document will provide some added commentary to what decisions were made and why on a particular System Generation. Your choices may be different and thus some of these questions may not appear, and others not listed may appear. Thus, it is strongly recommended to utilize the System Generation Document which is in vol.2 of the RSX-11M Ver 3.2 Documentation.

2.1. SYSGEN PHASE1 as a PREPGEN

A PREPGEN will be executed to safely anticipate the questions that will be asked based upon the designers intended system. If errors are made, then the designer may loop back to try again without the consequence of having a system partly generated. The answers will be saved in a file which in turn will do the answering on the actual SYSGEN. Listed below are the questions, answers and accompanying explanation to some of the answers.

Start with executing the SYSGEN file.

2.1.1. The SYSGEN as a PREPGEN

```
>@SYSGEN
....
(Identification)
....
>* 1. Inhibit MCR command lines ( )? [Y/N]:Y <CR>
>* 3. Use saved or canned answer file for input responses? [Y/N]:N <CR>
>*10. Save responses ( )? [Y/N]:Y <CR>
>*11. UIC for SYSSAVED files if not [200,200] [S]: <CR>
>*12. UIC for SYSGEN command files if not [200,200] [S]: <CR>
>*14. Device, when ready, for Executive sources (ddu:) [S]: DL1 <CR>
>;
>; Target Configuration
>;
>* 1. Processor ( ) [ ]: 11/23 <CR>
>* 2. Does processor have a switch register? [Y/N]:N <CR>
>* 3. Generating RSX-11S? [Y/N]:N <CR>
>* 4. Memory Size ( ) [ ]: 124. <CR>
>* 5. Mapped? [Y/N]: Y <CR>
>* 6. Any K-series laboratory devices? [Y/N]:N <CR>
>* 8. Floating point processor ( ) present? [Y/N]:Y <CR>
>*10. If using KW11-P as system clock: ... [ ]:0. <CR>
>*11. Line frequency: A-60Hz B-50Hz [ ] [S]:A <CR>
>*12. KW11-Y Watchdog timer support? [Y/N]:N <CR>
>*13. Memory parity support? [Y/N]:N <CR>
>*15. Highest Interrupt vector ( ) [ ]:0 <CR>
>;
>;                    Device Message
>;
>*16. Devices [S]:CO,DL,DY,NL,TT. <CR>
>;
>;                    Target Message
>;
>;                    Host Configuration
>;
>* 1. Are you running SYSGEN standalone ( )? [Y/N]:Y <CR>
>* 2. Is a line printer available? [Y/N]:N <CR>
>* 4. Does the listing/map device have at least 120 columns?
     [Y/N]:Y <CR>
>* 5. Assembly listings device ( ) [ ] [S]: <CR>
>* 6. Map device for Executive and device drivers ( ) [ ] [S]: <CR>
>;
>*EOS* Do you want to: <CR> -continue R-repeat section E-exit
      P-Pause [S]: <CR>
>;
>; Executive Options
>;
>; Answer Y(ES) if the following support is desired
>;
>* 1. Will DBMS-11 be used? [Y/N]:N <CR>
>* 2. FILES-11 ACP? [Y/N]:Y <CR>
>* 3. RMS record locking and placement control: [Y/N]:Y <CR>
>* 4. Non-resident tasks ( )? [Y/N]:Y <CR>
>* 5. Loadable task loader? [ ]:Y <CR>
>* 6. Memory management (PLAS) directives? [ ]:Y <CR>
>* 7. Send/Receive by reference directives? [ ]:Y <CR>
>* 8. Get mapping context directive? [ ]:Y <CR>
>;                    Additional
>;                    Comments
>*11. Multi-user protection? [ ]:Y <CR>
>*12. ANSI Magtape ACP [ ]:N <CR>
>*13. On-line formatting (FMT) or on-line user-mode diagnostics
     [ ]:Y <CR>
>*14. Loadable device drivers? [ ]:Y <CR>
>*15. Should the default for all applicable drivers be loadable?
     [ ]:Y <CR>
>;                    Additional
>;                    Messages
>*17. Task termination and device not ready messages ()?
     [ ]:Y <CR>
```

```
>*18. Power fail recovery? [ ]:Y <CR>
>*19. Install, request, and remove on exit? [ ]:Y <CR>
>*20. Large (20K) Executive? [ ]:N <CR>
>*21. Logical device assignment? [ ]:Y <CR>
>*22. Logging of device errors and timeouts? [ ]:Y <CR>
>*23. Logging of undefined interrupts? [ ]:Y <CR>
>*25. Do you intend to include a user written driver? [ ]:Y <CR>
>*26. Include routine $GTWRD? [ ]:Y <CR>
>*27. Include routine $PTWRD? [ ]:Y <CR>
>*28. Disk writecheck? [ ]:N <CR>
>*29. Executive Debugging Tool (XDT)? [ ]:Y <CR>
>*31. Industrial network or DECnet? [ ]:N <CR>
>*32. Queue manager or queued print spooler? [ ]:Y <CR>
>*33. Group global event flags? [ ]:Y <CR>
>; Directive Message
>*34. Enter code for directives [S]:ABCDEFHI. <CR>
>*35. Number of pre-allocated QIO packets [ ]:15. <CR>
>*36. Size of data transfer vector in words [ ]: <CR>
>*37. Round-robin scheduling interval in ticks [ ]: <CR>
>*38. Highest priority class for consideration [ ]: <CR>
>*39. Lowest priority class too consideration [ ]: <CR>
>*40. Executive level disk swapping interval [ ]: <CR>
>*41. Swapping priority [ ]: <CR>
>*42. Size of dynamic storage region ( ) in words [ ]: <CR>
>*45. System crash register and stack dump device CSR [ ]: <CR>
>*46. Checkpointing A-Yes B-With system ckpnt. file [ ]:B <CR>
>*47. System cntrld partitions A-yes B-With auto compaction (SHF)
      [ ]:B <CR>
>*48. Dump Analysis A-PANIC B-Crash ( ) [ ]:A <CR>
>*52. CSR address of PANIC dump device [ ]: <CR>
>;
>; Terminal Driver Options
>;
>* 1. Terminal driver desired ( ) [ ]:C <CR>
>* 2. Unsolicited input timeout valve in seconds [ ]:600
>;
>; Answer Y(ES) if the following support is desired
>;
>* 5. Task checkpointable during terminal input? [ ]:Y <CR>
>* 6. Unsolicited input character AST? [ ]:Y <CR>
>;                    Additional
>;                    Messages
>*10. Escape sequence handling? [ ]:Y <CR>
>*11. Get multiple characteristics? [ ]:Y <CR>
>*12. Set multiple characteristics? [ ]:Y <CR>
>*13. Get terminal driver options? [ ]:Y <CR>
>*17. Read after prompt: [ ]:Y <CR>
>*20. CRT rubout support? [ ]:Y <CR>
>*25. Hardware unrecoverable input error notification [ ]:Y <CR>
>*26. Device independent cursor positioning? [ ]:Y <CR>
>;
>*EOS* Do you want to <CR>-continue R-repeat section E-exH
     P-pause [ ]: <CR>
>;
>;                    Message and end of Executive
```

```
>;
>;                    Peripheral configuration
>;
>;                    Peripheral Message
>;
>* 3. DL controller 0 [ ] [ ]:,,4 <CR>
>*53. Contr 0 unit 0. is on RL01/RL02 [ ] [s]: <CR>
>*53. Contr 0 unit 1. is on RL01/Rl02 [ ] [s]: <CR>
>*53. Contr 0 unit 2. is on RL01/RL02 [ ] [s]: <CR>
>*53. Contr 0 unit 3. is on RL01/Rl02 [ ] [s]: <CR>
>;
>* 3. DY controller 0 [ ] [ ]: ,,2 <CR>
>*58. Contr 0 unit 0. is SINGLE/DOUBLE density [ ]:DOUBLE <CR>
>*58. Contr 0 unit 1. is SINGLE/DOUBLE density [ ]:DOUBLE <CR>.
>;
>;                    Message
>;
>*18. CO controller 0 [ ] [ ]: <CR>
>;                    Message
>*20. The terminal driver [ ] [ ]:10 <CR>
>;
>;  Enter vector, CSR for:
>;
>*21. ZL controller 0 [ ] [ ]: <CR>
>*21. ZL controller 1    [ ]:300,176500
>*21. ZL controller 2    [ ]:310,176510
>*21. ZL controller 3    [ ]:320,176520
>*21. ZL controller 4    [ ]:330,176530
>*21. ZL controller 5    [ ]:340,176540
>*21. ZL controller 6    [ ]:350,176550
>*21. ZL controller 7    [ ]:360,176560
>*21. ZL controller 8    [ ]:370,176570
>*21. ZL controller 9    [ ]:070,176600
>;
>*EOS* Do you want to: <CR>-continue R-repeat section
      E-exit P-pause [ ]: <CR>
>;
>;                    Messages
>;
>* 1. Do you wish to edit any of the Executive files? [ ]:N <CR>
>;
>;
>*EOS* Do you want to: <CR>-continue R-repeat section
      E-exit P-pause [ ]: <CR>
>;
>;                    Messages
>;
>*EOS* Do you want to: <CR>-continue R-repeat section
      E-exit P-pause [ ]: <CR>
>;
>;                    Exiting Message
>;
>@<EOF>
>  (MCR)
```

This procedure was a PREPGEN. That is, no system is in the process of being generated. There is presently a file created in UIC [200,000] called SYSSAVED.DAT. The PREPGEN answers are saved and properly formatted in that file. The files actual use will be discussed later. Below are reasons for some of the answers given in the PREPGEN.

2.1.2. PREPGEN Identification

1 Inhibit since this is a PREPGEN.

3 The saved answer file does not yet exist.

10 Save responses for the future SYSGEN.

11.12
Important files in UIC [200,200]

14 MAPSRC Disk in drive DL1.

2.1.3. PREPGEN Target

3 No since RSX-11S is NOT a standalone system.

4 124.K memory is largest allowed by 11/23.

5 Mapped since memory is large.

6 None available/none used by this system.

8 Floating point to process real numbers.

10.12
No K-series support.

15 SYSGEN to calculate Highest Interrupt Vector.

16 CO-Console, DL-RL01 Disk Drives, DY-RX02 Floppy Disk Drives, NL-Null Device, TT-Terminal Driver

2.1.4. PREPGEN Host

1 Generating RSX-11M on an isolated computer

5 No need for Assembly listings

6 Place Executive and drivers into system

2.1.5. EXECUTIVE Options

1 No Asynchronous System Traps

2 File format chosen

12 No Magtape drivers used

20 No need to make it larger adjust pool size in VMR file at end of SYSGEN

28 No, slows down disk throughput

31 No links to other systems available

34 No sense switch, since no switch register

35 QIO packets to 15 due to possible heavy usage of devices

37 Round Robin of five computer ticks, is amount spent on a task before continuing to the nest. Microbot rough motion can be changed by a different round robin number. Smaller yields smoother/slower motion. Larger yields rough/faster/more spread out motion.

46 Need file to support PASCAL Tasks

47 Add (SHF) shuffler to allow more tasks to be partly in system-controlled partitions

48 PANIC for I/O Driver analysis

2.1.6. Terminal Drivers

1 Full Duplex package desired

2 Time limit to enter characters on a line with <CR>, Characters without <CR> on a line are ignored after time limit

2.1.7. Peripheral Configuration

3 Four RL01 drives per DL driver. Let system handle the rest of data

53 all four are RL01 drives

3 Two RX02 drives per DY driver. Let system handle the rest of data

58 Double density for more data per disk

18 Console driver labeled TTO

20 Ten terminal drivers desired.

21 Console remains as specified by system. For further information check Module MXV11-A in DEC's Micro computers and memories. Also check for Asynchronous Serial Interface DLV11-J in DEC's Memories and Peripherals. For vectors and addresses check either book's appendices for a table on vectors. As of this printing, since the paper tape option is not in the system configuration an attempt to use it as a terminal port has not yet been made.

2.1.8. Executive Assembly

1. No editing at this point. Notice the long paragraph that follows in the SYSGEN message. During the actual SYSGEN the necessary editing will be illustrated.

2.2. SYSGEN PHASE 2 as PREPGEN

If SYSGEN PHASE 1 is a PREPGEN, PHASE2 will execute and acknowledge as a PREPGEN. The Microbot driver was chosen to be compiled and loaded after the new system is in place.

2.2.1. The SYSGEN 2 as a PREPGEN

```
>@SYSGEN 2
. . . .
(identification)
. . . .
>* 1. In what UIC is SGNPARM if not [200,200] [s]: <CR>
>* 2. Expanded comments desired for SYSGEN phase II? [ ]:Y <CR>
>* 4. Is SYSGEN still running standalone? [ ]:Y <CR>
>* 5. If not in DL1: enter object files device when ready ( ):[ ]: <CR>
>* 6. Have you already built the libraries, the Exec, and the
      loadable drivers? [ ]:N <CR>
>* 8. Create ANSLIB to support Big-buffering? [ ]:N <CR>
>;
>;  Loadable driver discussion
>;
>*10. List Executive Map? [ ]:N <CR>
>;
>;  Build Loadable drivers
>;
>* 6. Device for driver's maps [ ] [ ]: <CR>
>* 9. Build all selected loadable drivers into DRVPAR? [ ]:Y <CR>
>*10. Driver 2-character device minemonic [ ]: <CR>
>*11. FCP desired ( ) [ ] [ ]: <CR>
>*13. Build the unprivileged queue manager utilities? [ ]:Y <CR>
>*14. Build the task PMD? [ ]:Y <CR>
>*15. Build RMDEMO? [ ]:Y <CR>
```

```
>*17. Edit any privileged task build files? [ ]:N <CR>
>*18. Map device for privileged tasks ( ) [ ] [ ]: <CR>
>*19. Edit SYSVMR.CMD? [ ]:N <CR>
>;
>; Exiting Message
>@   <EOF>
>    (MCR)
```

This procedure continues the PHASE I PREPGEN. No system has been generated there have not been nor will there be further additions to the file [200,200] SYSSAVED.DAT. Below are some explanations to the answers chosen.

2.2.2. PREPGEN PHASE II Maintenance

1 The file remains in its place

2 Expand comments for maximum information

4 Still standalone

5 The MAPSRC disk will be on DL1:

6 Libraries, Executive built during the actual SYSGEN

8 No Big-buffer necessary

10 No reason to list

2.2.3. Build Loadable drivers

6 No reason to list

9 Places drivers into system partition DRVPAR

10 Microbot Diver built later

11 Middle for best file access

15 RMDEMO for system display

17 No partition changes necessary

18 Not necessary

19 Edit during actual SYSGEN

2.3. SYSGEN and Saved Answer File Execution

After SYSGEN PHASE I the file [200,200]SYSSAVED.DAT is complete. SYSGEN PHASE II (SYSGEN) does not add to that file. As a point of interest utilize the Peripheral Interchange Program (PIP) to view this answer file. There are many way to invoke PIP at this time.

```
>RUN $PIP
PIP>TI:=SYSSAVED.DAT

        . . . .
        LISTING OF FILE
        . . . .

PIP>



>INS $PIP
>PIP TI:=SYSSAVED.DAT

        ...
        LISTING OF FILE
        ...

> (MCR)
```

To utilize the answer file the DAT has to be changed to CMD. This can be either by copying or renaming while in PIP. It is strongly suggested to make a copy rather than renaming so as to have extra versions in case of file damage or deletion.

```
        TO RENAME
>PIP SYSSAVED.CMD=SYSSAVED.DAT/RE

        TO COPY
>PIP SYSSAVED.CMD=SYSSAVED.DAT
```

In listing the SYSAVED file, notice the many .SET* $* commands. The three character mnemonics next to * have to do with the chosen options, during SYSGEN PHASE I. Occasionally, the listing of SYSSAVED is distorted, but the file itself is not.

2.3.1. SYSGEN Using SYSSAVED.CMD

```
>@SYSGEN
. . . .
(Identification)
. . . .
>* 1. Inhibit MCR command lines ( )? [ ]:N <CR>
>* 2. Have you made a copy of the distribution kit? [ ]:Y <CR>
>* 3. Use saved or canned answer file for input responses? [ ]:Y <CR>
>* 4. Undefined answer wakeup? [ ]:Y <CR>
>* 5. Use saved host configuration? [ ]:Y <CR>
>* 6. Use saved peripheral configuration? [ ]:Y <CR>
>* 7. Skip end of section ( ) breakpoints? [ ]:Y <CR>
>* 8. Skip end of execution ( ) breakpoints? [ ]:Y <CR>
>*10. Save responses ( )? [ ]:Y <CR>
>*11. UIC forSYSAVED files if not [200,200] [ ]: <CR>
>*12. UIC for SYSGEN command files if not [200,200] [ ]: <CR>
>*13. Clean up files from previous GENS? [ ]:Y <CR>
>*14. Device, when ready for executive sources ( ) [ ]: DL1 <CR>
>;
>; Mount of DL1 and many deletions
>; Processing of target configuration,
>; Host configuration, Executive Options,
>; Terminal Driver Options, and
>; Peripheral Configuration. Many
>; Messages based upon Prepgen
>; Answers.  Executive Assembly
>; Coffee break approximately one half
>; hour.  Device Driver Assembly approx-
>; imately twenty minutes.
>;
>;   Long paragraph on resident and
>;   Loadable drivers printed then a
>;   pause to allow editing of the
>;   file RSXBLD.CMD.  To be safe
>;   look under [200,200] and [11,10]
>;   and edit both.
>;
>                    (MCR)
>INS $EDI <CR>
>EDI [11,10] RSXBLD.CMD <CR>
[00027 LINES READ IN]
[ PAGE 1 ]
*N14 <CR>
*P <CR>
CORSIZ=32
*I GBLDEF=$USRTB:0 <CR>
*EX <CR>
[ EXIT ]

>SET /UIC=[200,200] <CR>
>EDI RSXBLD.CMD <CR>
[ 00027 LINES READIN ]
[ PAGE 1 ]
*N14 <CR>
*P <CR>
CORSIZ=32
*I GBLDEF=$USRTB:0 <CR>
*EX <CR>
[ EXIT ]

>RES ...AT. <CR>

AT. --CONTINUING
```

```
>
>;
>; Task Build Preparation
>; and some file manipulation
>;
>@ <EOF>
> (MCR)
```

2.3.2. Actual SYSGEN

1. Want to do a real SYSGEN

2. Made a copy at kit long ago in this process

3. Using SYSAVED.CMD

4. In case of a missed answer, have the SYSGEN signal and wait.

5. Use Host configuration already determined

6. Use Peripheral configuration already determined

7. Skip breakpoints by choice

8. Skip breakpoints by choice

10. Save new file if there will be any changes

11. SYSSAVED stays under [200,200]

12. SYSGEN is under [200,200]

13. Make as much room on distribution kit as possible

14. Sources will be on DL1

Invoking Line Editor, EDI, to adjust RSXBLD.CMD under [200,200] and [11,10] to allow addition of Microbot driver later on; then continue with remainder of SYSGEN Phase I.

2.4. SYSGEN PHASE II Execution

At this point the system is not complete. The system is now ready for integration into executable and bootable files. PHASE II will acknowledge PHASE I as a real SYSGEN and continue accordingly.

2.4.1. SYSGEN 2

```
>@SYSGEN2 <CR>
>;
>; Introduction
>;
>* 1. In what UIC is SGNPARM.CMD if not [200,200] [ ]: <CR>
>* 2. Expanded comments desired for SYSGEN Phase II? [ ]:N <CR>
>* 4. Is SYSGEN still running stand alone? [ ]:Y <CR>
>* 5. If not in DL1: enter object files device when ready () [ ]: <CR>
>;
>; Mount attempt of MAPSRC on DL1
>;
>* 6. Have you already built the libraries, the EXEC, and
     the loadable drivers? [ ]:N <CR>
>* 8. Create ANSLIB to support Big-buffering? [ ]:N <CR>
>;
>; Building of drivers
>; Invoking Library task LBR
>;
>LBR
LBR>CTRL Z <CR>

>REM LBR
>;
>; Invoking Line Editor EDI to
>; enter designer drivers. Microbot
>; driver entered later.
>;
>EDI RSXBLD.CMD
[ 00028 LINES READ IN ]
[ PAGE 1 ]
*EX <CR>

[ EXIT ]

>;
>; Executive and Terminal Drivers Built.
>;
>* EOX * Did the Executive build successfully? [ ]:Y <CR>
>;
>* 10.List Executive Map? [ ] [ ]:N <CR>
>;
>; Build Loadable drivers
>;
>* 6. Device for drivers maps [ ] [ ]: <CR>
>* 9. Build all selected loadable drivers into DRVPAR? [ ]:Y <CR>
>;
>; File manipulation and Task building
>; and option to build own drivers.
>;
>*10. Driver 2-character device mnemonic [S]: <CR>
>*11. FCP desired ( ) [ ] [ ]: <CR>
>*13. Build the unprivileged queue manager utilities? [ ]:Y <CR>
>*14. Build the task PMD? [ ]:Y <CR>
```

```
>*15. Build RMDEMO? [ ]:Y <CR>
>;
>; List of some tasks to be part of the system.
>;
>*17. Edit any privileged task build files? [ ]:N <CR>
>*18. Map device for privileged tasks ( ) [ ] [ ]: <CR>
>;
>; Building of tasks listed
>; above. This building lasts
>; approximately forty minutes
>;
>*EOX* Did the privileged tasks build successfully [ ]:Y <CR>
>;
>;                  Device reassignment for next
>;                  section of SYSGEN PHASE II.
>;
>;                  Pause for designer to generate
>;                  RSX11M.SYS file as suggested.
>;

>                   (MCR, assume PIP is installed INS)
>PIP (suggested command above) <CR>
>RES ...AT. <CR>

>*19. Edit SYSVMR.CMD? [ ]:Y <CR>
>EDI SYSVMR.CMD
[ 00036 LINES READ IN ]
[ PAGE 0 ]
*N 15 <CR>
*P <CR>
SET /MAIN=FCPPAR:*:240:SYS
*C/240/400/ <CR>
SET /MAIN=FCPPAE:*:400:SYS
*REN <CR>
*N 11 <CR>
*P <CR>
INS UNL             !INSTALL UNLOAD
*I <CR>
INS PIP             !PIP <CR>
INS EDI             !LINE EDITOR <CR>
INS EDT             !TEXT EDITOR <CR>
INS TKB             !TASK BUILDER <CR>
INS MAC             !MACRO ASSEMBLER <CR>
<CR>
*EX
[ EXIT ]

>;
>;                  Additional installations and invoking
>;                  Virtual Monitor Routine, .VMR.
>;
>VMR
ENTER FILENAME:@SYSVMR <CR>
>;
>;                  Process system file so that is
>;                    will be bootable on power up.
>;
VMR>CTRL Z <CR>
>;
>;                    Exiting messages
>@ <EOF>
>BOO [1,54] <CR>
XDT:26

XDT>G <CR>

(SYSTEM BOOT)

>DEV <CR>
( Listing of devices in configuration )
>SAV /WB <CR>
( Saving system so that it is bootable )
>
```

2.4.2. SYSGEN 2 Explanations

1. SGNPARM.CMD remains in [200,200]

2. No need for expanded comments

4. Stand alone still running

5. Objects remain in DL1

6. Not yet built

8. No need of big-buffering

NOTE Some possible failures in creating new versions but it is not critical. Some module replacement also expected.
LBR. Microbot not resident driver

EDI. Microbot driver assembled later

EOX. If any errors occur, either retry or check executive section in PHASE I SYSGEN.

EOX. If any error, either retry or check terminal driver section in PHASE I SYSGEN

10. No need to list

6. No need to list

9. No reason to put drivers elsewhere

10. No drivers now, Microbot driver built later

11. Middle: anticipate much user editing thus best for file open/close environment

13. Though no line printer in the system configuration, various options to control file access printing may be used toward the console for partial file printing instead of entire file by PIP.

14. Post Mortem Dump if desire to see why system crashes

15. Can see task activity and system status especially during heavy usage.

17. No need to edit.

18. No need to list maps.

EOX. If any errors some files may be defective. Errors are not likely.

PIP. Create new version of contiguous space of RSX11M from a Task TSK to a SYSTEM SYS file

RES. Resume indirect file processing by SYSGEN PHASE II

19. SYSVMR to be edited so that there is no partition allocation failure on the system configuration.

EDI. SET /MAIN=FCPPAR:*:240:SYS

SET -Identify a system-wide characteristic

/Main -Establish a partition

=FCPPAR -Partition name

:* - moves beginning address to the lowest address possible

:240 - size of partition which increased to 400 which is really 400 X 100 base 8. This prevents a previous VMR allocation failure.

EDI. INS PIP to INS MAC. These tasks were installed here by choice, they could have just as easily been placed in DL0:[1,2]STARTUP.CMD as will be seen later.

VMR.

SYSVMR.CMD is now the bootable system for the current power up. If power down, the old bootable system takes over. But it may already have been deleted through the SYSGENs.

BOO. Attempt to boot the new system.

XDT. 26 means number of blocks allocated for system boot. G means to continue through Executive Debugging Tool XDT breakpoint.

DEV. Check for devices in system configuration (recognized by system)

SAV. At this point, all conditions allow a SAVE execution. /WB specifies hardware boot block for this new system. Automatically, system and library is redirected to DLO since distribution kit on RL01 disc. Automatically DLO is mounted and DLO:[1,2] STARTUP.CMD is executed.

2.4.3. Bootable System Address

If the console break key is hit, the computer stops and the executive prompt @ appears. To reboot the system type:

```
@773000G
```

or else type to proceed from the break point

```
@P
```

The boot address above may be different for other computers. The assignment of this above address may have come from an earlier SAV/WB command or else this address is assigned by the manufacturer.

3. System Generation Cleanup

The system is not yet fully operational for use in a student environment. User accounts must be made, the Microbot driver needs to be assembled and the PASCAL Compiler must be installed.

3.1. Device Check

As seen above the MCR command DEVice was used to check and see if all intended devices are available. For the system generated above, the system will respond as below:

```
>DEV  <CR>
DL0:  PUBLIC MOUNTED LOADED LABEL=RSXM26 TYPE=RL01
DL1:  LOADED TYPE=RL01
DL2:  OFFLINE LOADED TYPE=RL01
DL3:  OFFLINE LOADED TYPE=RL01
DY0:  OFFLINE LOADED TYPE=RX02
DY1:  OFFLINE LOADED TYPE=RX02
CO0:  TT0:
TT0:  [,] - LOGGED ON LOADED
TT1:  LOADED
TT2:  LOADED
TT3:  LOADED
```

```
TT4: LOADED
TT5: LOADED
TT6: LOADED
TT7: LOADED
TT10: LOADED
TT11: OFFLINE LOADED
NL0:
TI0:
CL0: TT0:
LB0: DL0:
SY0: DL0:
> (MCR)
```

PUBLIC - all users have access

MOUNTED - device actively in system

LOADED - device driver in system

LABEL - up to six character name of particular disk, only seen by privileged user

TYPE - device type

OFFLINE - in system configuration but the host (sometimes hardware) does not have it.

CO0: TT0: - terminal zero is console zero

[,] - LOGGED ON - with no accounts in existence this is a single user for now and he/she is on the system.

NL0: - the null device exists, so that if some material must be sent somewhere, this non existent medium is available.

TI0: - the issuing terminal driver, so that various operations may be sent to the users terminal screen

CL0: TT0: - line printing to console.

LB0: DL0: - system library on drive 0
SY0: DL0: - system files on drive 0

3.2. User Accounts

Multi-user systems require accounts for protection and partitions. At this point, no accounts exist, thus an account file must be created to be used by the ACNT task. The ACNT task is mostly self prompting. Some notes for convenient execution:

1. Accounts are identified by two numbers, first a group number, second an owner (user) number.

2. Group numbers from one to ten base eight will be privileged accounts. From eleven to three hundred seventy seven base eight are non-privileged, for the average user

3. Owner numbers range from one to three hundred seventy seven base eight

4. On the first run of ACNT you must create file, which is an option. If an account file already exists, a new one cannot be created.

5. On first creation of the file, the maximum number of accounts is requested

6. ACNT also creates User file Directories as need on the device specified

7. To execute the account task type the following

```
>(MCR)
>RUN $ACNT <CR>
(Into accounting task)
```

8. Add the accounts needed for student use. It is suggested to use one group number for all class accounts, this simplifies file manipulation later. Use a group number that is not present on the system disk just to be careful.

9. After returning to MCR, type;

```
>BYE <CR>
```

and signon to a privilege account if desired, to complete system cleanup.

3.3. MicroBot Driver for the New System

The MicroBot Driver is designed as a loadable driver so that development may be easier and more flexible. If the driver is made resident, any change in the driver will require another system generation. Since the MicroBot Driver is loadable, it can be unloaded, adjusted, reassembled then loaded again. This is possible since during the Executive portion of SYSGEN PHASE I, a command file was edited to recognize future user written drivers.

Assembly of the MicroBot Driver is a slightly involved process and must be done for any driver change or new system generation. Editing of the driver will not be discussed in this document. Only the necessary assembly steps for a compatible driver will be discussed. The processing of the driver involves compilation of the MACRO Assembly Code of the driver, coordination of the Object Code and Data Base of the Driver with the system Executive and finally loading the Driver for use. The MicroBot Driver source code, assembly command file, and a documentation file are on the disk volume labeled ROBOT 1 under the UFD [1,374]. The most critical files are:

ALL.CMD	1 Block
MDB.CMD	1 Block

MDT.CMD	1 Block
DRVBLD.CMD	1 Block
WARNING. DOC	3 Blocks

The ALL.CMD file handles most of the work. After the LBR task is invoked, the necessary line to use is listed in the file WARNING.DOC. The typical MicroBot Driver Assembly session may look like:

```
>        (MCR, On privileged account)
>SET /UIC=[1,374] <CR> (for proper execution)
>@ALL <CR>
>MAC @ MBD  (ALL.CMD file)
>MAC @MBT   (ALL.CMD file)
>RUN $LBR   (ALL.CMD file)
LBR>DL0: [1,24]RSX11M/RP=DL1:[1,374]MBDRV,MBTAB <CR>
LBR>CTRL Z <CR>
>TKB @DRVBLD (ALL.CMD file)
>RUN $SHUTUP (ALL.CMD file)
RSX11M SHUTDOWN PROGRAM

ENTER MINUTES TO WAIT BEFORE SHUTDOWN:0 <CR>
OK TO SHUTDOWN? [Y/N]:N <CR>
>@EOF
> (MCR)
```

At this point some new files will appear on parts of the disks. On UFD [1,374] the new files will be:

MDBRV.OBJ	2 Blocks
MBDRV.LST	66 Blocks
MBTAB.OBJ	2 Blocks
MBTAB.LST	67 Blocks

The above files are intermediate results, the actual driver files appear on the system disk on VFD [1,54]:

MBDRV.TSK	4 Blocks
MBDRV.MAP	4 Blocks
MBDRV.STB	1 Block

When the MicroBot Driver is loaded and invoked by the various driver commands the [1,54] files will be used. To load the MicroBot Driver, type the following while on a privileged account:

```
> LOAD MB: <CR>
```

The loading of the MicroBot Driver will be done during the system boot on power up or after system break. This will be discussed in the section labeled [1,2] STARTUP. If an old version of the driver which is not coordinated with the current operating system, an error during the load operation will occur and the driver is thus useless.

3.4. PASCAL Compiler and Library

When receiving a release kit from a software company, there are many files that will be present. For example from the Omsi-Pascal Release kit there are:

1) Documentation (DOC) files describing various components and capabilities of the release kit.

2) Object Library (OLB) files which will be parts of the functioning compiler.

3) Overlay Description (ODL) files that control usage of the Object Library files.

4) Command (CMD) files that execute the formation of other files into the Pascal Compiler.

5) Pascal (PAS) files that contain useful routines, games, and other capabilities

The release kit has been utilized to build the Pascal Compiler particularly suited for the RSX-11M Operating System. When the compiler has been built, some of the original files of the release kit are deleted since they are not needed. Some files are eliminated since the computer may have the support that the release kit has the capability to duplicate. For example, the SYSGEN has chosen a floating point processor/hardware. The release kit has various files with capabilities to work with an available floating point processor, or to simulate it on a system without the floating point processor. As of this printing, these pertinent files are in the following locations:

1) Disk Label: Robot 1

```
        [301,1]*.DOC;*
```

2) Summer 1981 System Disk Label: RSXM26

```
        [1,54]   PASCAL.TSK;1

        [1,1]    PASLIB.OLB;*
                 PAS.ODL;*
                 IMP.TSK;*
                 PROFIL.TSK;*
                 STRING.OBJ;*
```

3) Winter 1983 System Disk Label: RSXM26 Same files as Summer 1981

There are other files which include games and other capabilities most likely on Disks Labeled: ROBOT 1. The essential files are the following:

1) [1,54] PASCAL.TSK is the Pascal compiler suited to the RSX11M Operating System. Pascal Source code is checked for syntax and definitions and the compilation result is placed in a MACRO-11 Assembly Level Language file.

2) [1,1] PASLIB.OLB is the Pascal Library suited to the RSX11M Operating System. In Pascal Source Code, there are many functions and routines that are part of Pascal and they are contained in the Pascal Library

3) [1,1] PAS.ODL is the Overlay Design Language which controls patching into the Pascal Library for whatever support is needed for the particular program.

When developing Pascal Source Code into executable code, these files are used in various places. Consider a file with no syntax errors. The typical compilation looks like:

```
>PAS TEST=TEST <CR>
>MAC TEST=TEST <CR>
>TKB TEST/CP/FP=TEST,DL0:[1,1]PASLIB/CD <CR>
>      (MCR)
```

The PAS line invokes the Pascal Compiler under [1,54]. To use the compiler, it must be Installed, and will be discussed under the STARTUP file section. The TKB line utilizes the Pascal Library, and the overlay file controls patching into the Library as needed by the source code.

3.5. STARTUP Command File

When a system is booted, there are slight adjustments needed for a smooth operation. There may be the need to set the date and time, load loadable drivers, adjust terminal types and other operations. Observe the console on a system boot. By default, drive zero is considered the source of the system file and the system file is accessed by virtue of the SAVE command discussed earlier. Then the boot redirects drive zero as the system disk and the library disk. The disk in drive zero is mounted so the other files on the disk can be accessed. Finally, the file DL:[1,2]STARTUP.CMD is executed. It is in this command file that the adjustments are executed. The STARTUP file is very simple after a SYSGEN.

```
>PIP TI:=[1,2]STARTUP.CMD <CR>
.ENABLE SUBSTITUTION
.ASKS TIM Enter date and time (dd-mmm-yy hh:mm)
TIM 'TIM'
>  (MCR)
```

Since there is no BYE command, there is a user logged on to the console. The DEV command will show the user [,] on the console. This is a privileged user with no User File Directory. The ENABLE command transfers control of character strings to the STARTUP.CMD file, such as allowing the day on date entered by the user to be substituted into the string TIM above. The ASKS allows a string to be substituted from the keyboard for a variable and be used in MCR commands. Obviously the date and time are desired on each system boot. The STARTUP,CMD file to be used in a student environment will be more complex. The file is designed for one console, eight video terminals eight microbots, a second disk drive (DL1), the date and time, Pascal compiler and some multiuser system adjustments. For such needs, the STARTUP.CMD file may look like:

```
>PIP TI:=DL0:[1,2]STARTUP.CMD
.ENABLE SUBSTITUTION
acs dl0:/blks=512.
SET /TERM=TT0:LA120
SET /TERM=TT1:VT52
SET /TERM=TT2:VT52
SET /TERM=TT3:VT52
SET /TERM=TT4:VT52
SET /TERM=TT5:VT52
SET /TERM=TT6:VT52
SET /TERM=TT7:VT52
SET /TERM=TT10:VT52
SET /LOWER=TT0:
SET /LOWER=TT1:
SET /LOWER=TT2:
SET /LOWER=TT3:
SET /LOWER=TT4:
SET /LOWER=TT5:
SET /LOWER=TT6:
SET /LOWER=TT7:
SET /LOWER=TT10:
LOA MB:
.ASKS TYME ENTER DATE AND TIME (DD-MMM-YY HH:MM:SS)
INS $FTB
INS $PASCAL
REM MAC
INS $BIGMAC
SET /PUB=DL1:
.ASK CHECK ECE494 STARTUP
.IFT CHECK MOU DL1:ECE494
.IFF CHECK MOU DL1:ROBOT1
ACS DL1:/BLKS=512:
SET /NOPRIV=TT0:
BYE
> (MCR)
```

Notice the following

(1) ACS allows checkpointable area which is especially needed for Pascal originated files. Notice the /CP term on the TKB line in the Pascal Compiler section. This allows a path by which some executing tasks may be partially on the storage disk and the system calls up parts of it when needed.

(2) SET /TERM=TT0:LA120 describes Terminal Driver Zero as LA120 format (ie. a DECWRITER III). Also, the system console is TT0 by virtue of a SYSGEN specification.

(3) SET /TERM=TT1:VT52 describes Terminal Driver One as VT52 format (ie. a video terminal).

(4) SET /LOWER=TT2: enables recognition of lower case characters from the terminals

(5) LOA MB: Loads the MicroBot Driver that has been configured to the new system as described in the MicroBot Driver section.

(6) INS $FTB installs the fast task builder. FTB is not as elaborate as TKB but if TKB not needed, a smaller and faster executable task may result.

(7) INS $PASCAL installs the Pascal Compiler as described in the PASCAL Compiler and Library section.

(8) REM MAC removes the Macro assembler from the system.

(9) INS $BIGMAC installs a larger Macro assembler that is faster, has a larger variable list and also takes up more memory. Note, a warning message may occur due to difficulty in partition allocations in memory but it is not a fatal error.

(10) SET /PUB=DL1: allows access to the disk in drive one by all users.

(11) .ASK will generate a yes/no answer. If yes .IFT is executed, a no will have .IFF executed. This will only allow, at this point, the mounting of two particularly named disks. These three lines can easily be replaced by: MOU DL1:/OVR/VI The /OVR is a privileged switch which will mount a disk no matter what the name. The /VI allows the viewing of disk format. This will be discussed in the System Maintainance section under Virgin Disk Initialization.

(12) SET /NOPRIV=TT0: sets Terminal Driver Zero to be nonprivileged. Since TT0 is set as the console, any one logging on the privileged console has total access to the system. (Not desired in student environment and multiuser systems). The /NOPRIV cannot be countered with a /PRIV command by a privileged account. Only a new system boot without /NOPRIV can counter an earlier /NOPRIV command

(13) BYE closes off the command file execution and logsoff the privileged user of the booting [,].

The STARTUP.CMD file must be on the system disk under UFD [1,2] and the latest version is used on the system boot. It is strongly recommended that some type of [1,2] STARTUP.CMD file exist else the user [,] will remain signed on the console and unauthorized users will have free run of the system. Unfortunately, the same effect occurs if the STARTUP.CMD file is aborted with:

```
. . . .
. . . .CTRL C (Printing Suppressed)
MCR>ABO . . . AT. <CR>

(Abort Message)
> (MCR under [,])
```

When the system boots the console will print the following:

```
RSX-11M V3.2 BL26 124K MAPPED
>RED DL:=SY:
>RED DL:=LB:
>MOU DL:RSXM26
>@DL:[1,2]STARTUP
```

The last line with @ is the call to the indirect file processor IND. BIGIND is usually only used for SYSGEN. Only one or the other may be installed in the system and it is signified by . . . AT . in the system. Note, keep much of this privileged information away from non privileged users.

3.6. Copy New System on Extra Disc and First Cleanup

Check the status of the system disk by:

```
>PIP DL0:/FR <CR>

DL0:HAS 3148.BLOCKS FREE, 7092.BLOCKS USED OUT OF 10240.
LARGEST CONTIGUOUS SPACE=2798.BLOCKS

> (MCR)
```

Since the numbers are followed by a decimal, they are base ten. There are 3148. blocks available for files. There are 7092. blocks used either by files or are physically defective and thus not useful. There 10240. total blocks available on a RL01 disc. There are 2798. blocks all adjacent to one another in the largest single continuous region on the disk.

Through editing there may be extra obsolete copies of files like SYSSAVED.CMD, SYSVMR.CMD and others which are not needed. They may as well be deleted to make more room on the disk, for other uses.

Save all the work at the SYSGEN and system setup by making a backup copy. Obviously the disk MAPSRC is not to be used. RL01 discs have a density of 800 bits per inch (bpi). Thus the Disc Save and Compress Utility of 800 bpi must be used. A typical backup session may look like the following:

```
> (MCR on Priviledged Account)
>BOO DL0:[1,51]DSC58 <CR>
RSX11S V2.2 BL26 DISK SAVE AND COMPRESS UTILITY V3.2

DSC58>DL1:/VE=DL0:/BL=12 <CR>
DSC--45 START VFY PASS

DSC58> (BREAK KEY)
001274
@773000G  (Reboot System)
. . . .
. . . .
> (MCR)
```

Notice the following:

1) BOO boots the Disk Utility into memory and the regular system is eliminated from memory

2) DL1:/VE is the target disk which is either a never used virgin disk or any disk whose data is not needed. /VE is a switch to verify information between the source disk to the target.

3) DL0:/BL=12 is the source disk which has the new system. /BL=12 means 12 blocks base eight at a time are transferred from the source disk to the target.

4) CTRL C stops the use of DSCS 8

5) @773000G Reboots the system

Either disk can be used as the system disk. Notice however, that the ACNT system keeps track of number of LOGINs and the unused disk will not have these. Obviously a change on one disk can be duplicated by repeating the operations or by another Disk Save and Compress. The Disk Save takes about ten minutes and you will notice the Drive n lights flickering constantly. Occasionally an error message will display a file number and since it was not properly closed or maintained, it will not be copied. Check to see which file was not copied and if important, copy it.

4. System Maintainance Notes

This section will cover commands and tricks to keep the system running reliably in a student environment. Many of these procedures came out of improvisation and necessity to keep the system running under heavy use.

4.1. Virgin Disk Initialization

When either a new disk is delivered or the format of an old disk is to be changed, the commands in this section should be used. Notice however, if a disk with data is to be initialized, the data is lost since the format of the disk is changed. Many items will not be specified since default value was adequate. For example, the default of the number of Use File Directories and number of file entries on the disk was adequate. A typical disk initialization session may look like:

```
> (MCR on Priviledged Account)
>ACS DL1:/BLKS=0 <CR>
>DMO DL1:ECE494/DEV <CR>

        (Dismount Message)

>SET /NOPUB=DL1: <CR>
>ALL DL1:/TERM=TT0:
>INI DL1:ECE494/BAD=[AUTO]/FPRO=[RWED,RWED,,R]/LRU=11
      /WIN=14 <CR>
>DEA DL1: <CR>
>SET /PUB=DL1 <CR>
>MOU DL1:ECE494/VI <CR>
> (MCR)
```

Notice the following:

1) ACS removes the means of check pointing of files on the disk in drive one

2) DMO . . . . /DEV dismounts the disk from the system. At this point, a new disk may be put in or the present disk will be reinitialized.

3) SET /NOPUB sets the device to be no longer public to other users.

4) ALL allocates the device to only the specified terminal. Here the console is assumed.

5) INI initializes the disk as desired. The defaults can be changed. The disk is in drive one. The name of the disk volume will be ECE494. The BAD blocks will be AUTOmatically marked and not used by the system. The file Protection is specified. R-read, W-write, E-extend, D-delete. First set for system privilege (accounts with group number one to ten base eight). Second set for owner privilege so that the owner can do what is desired. Third set no privileges for others with the same group number. Fourth, set only Read by persons outside of the same group number. This the maximum file mask allowed for any file on the disk. The mask can be reduced by the user or privileged accounts. This file mask is used so that students may not view each others files or transfer files except by retyping. LRU allocates the number of directory file control blocks. Eleven base eight is adequate since only nine base ten terminals are liable to access the disk at a time by the lab enrollment limits. Due to high usage the LRU's may have to be increased. WIN allocates file window pointers used for all types of file manipulation.

6) DEA deallocates the disk, thus it reverses the allocation.

7) SET /PUB sets the disk to be accessible to all users.

8) MOU mount the disk to be used in a regular basis. Then the ACS command should be used. The /VI switch on the MOU command will list a description of the disk format for files, the device, the disk label and other information.

On the initial SYSGEN edit of [1,2] STARTUP.CMD both disks were allocated 200. blocks each. Due to heavy use of the system the blocks were both changed to 512. on the ACS command.

4.2. Pascal Compiler Failure

During heavy usage the Pascal Compiler occasionally fails. By experience, the usual failure is the inability to recognize the Pascal Reserve Word PROCEDURE. The recovery of a good compiler may look like:

```
> (MCR on Priviledged Account)
>ACS DL1:/BLKS=0 <CR>
>DMO DL1:/DEV <CR>
```

```
>MOU DL1:RSXM26.VI <CR>

                    (Disk Description Message)

>PIP DLO:[1,1]PASLIB.*;*/DE <CR>
>PIP DLO:[1,1]PAS.*;*/DE <CR>
>PIP DLO:[1,1]PASLIB.OLB=DL1:[1,1]PASLIB.OLB;1 <CR>
>PIP DLO:[1,1]PAS.ODL=DL1:[1,1]PAS.ODL;1 <CR>
> (BREAK KEY)
003640
@773000G  (REBOOT SYSTEM)
```

Notice the following:

1) ACS must close the checkpointable blocks and must DMO disc from system

2) MOU the system disk copy and for curiosity /VI views the disk information.

3) PIP . . . /DE removes the old defective library and overlay files.

4) PIP copies fresh versions fo the files for use.

5) Reboot has been found to be the most reliable manner to continue compiler usage. Occasional errors happen if the following is used instead:

```
>REM $PASCAL
. . . .
. . . .
>INS $PASCAL
```

4.3. Terminal Driver Failure

Another common failure of this system is software hangups. Occasionally the terminal driver becomes lost to the point where no keyboard input is recognized and no screen output occurs. Since terminal drivers are part of the system bootable file they cannot be removed (REM) and then installed (INS) again. The only recovery presently known is a system reboot.

4.4. Adding RL01 Drives

During the SYSGEN, four RL01 drivers are available. This matches the hardware available on one driver board. The disk drive numbers are distinguished by the Drive in white lights on the front of the disk drive. The next purchase of a disk drive should include the Drive 2 light. The physical connections of the new drive to the system should be done by qualified personnel such as the university technicians or a DEC field service representative. Notice that with a DEV command, two additional DL drivers are loaded and offline ready to be used.

4.5. Student Classroom Disk Maintainance

Due to heavy usage and student error some operations may be done almost daily. An often needed operation is a Disk Save and Compress for contiguous space for executable tasks. The disk copying will save many hours for a student who may have accidentally deleted source code files. A typical sequence to save particular files from a copy to the present disk may look like:

```
> (MCR Priviledged User)
>ACS DL1:/BLKS=0 <CR>
>DMO DL1:/DEV <CR>
>MOU DL1:/OVR/VI <CR>
>PIP DL1:[20,19]/LI <CR>

        (Directory Listing)

>PIP DL0:[1,4]TEST.PAS=DL1:[20,19]TEST.PAS;7 <CR>
>PIP DL0:[1,4]TEST.DOC=DL1:[20,19]TEST.DOC;2 <CR>
>DMO DL1/DEV <CR>
>MOU DL1: ECE494 <CR>
>ACS DL1:/BLKS=512. <CR>
>PIP DL1:[20,19]TEST.PAS;7=DL0:[1,4]TEST.PAS <CR>
>PIP DL1:[20,19]TEST.DOC;2=DL0:[1,4]TEST.DOC <CR>
>PIP DL0:[1,4]*.*;*/DE <CR>
> (wait for student to copy files)
>PIP DL1:[20,19]TEST.PAS;7,TEST.DOC;2/DE <CR>
> (MCR)
```

Notice the following:

1) This round about procedure must be done since only two disk drives presently exist. Further, all other users should be off the system since if the old disk is in, it is useless to them.

2) ACS,DMO present disk. Remove it.

3) Load the old disk into the drive and MOU it.

4) PIP lists the students directory. Ask students which old files are needed.

5) PIP copies the files from old class disk to system disk. Preferably to an empty UFD like DL0:[1,4].

6) DMO old copy of class disk. Remove it.

7) Load the present class disk and MOU and open the checkpoint allocation with ACS. Other student can now log on and continue.

8) Copy the student files to their directory and wait for them to copy it since they cannot delete files put into their directory by a privileged user.

9) DElete the student files on the system disk. And when the student is done copying, DElete the files put into the student directory by the privileged account.

10) Notice that if all accounts have the same group number ie. 20 in this case, the PIP commands are very easy to do.

Contiguous space becomes very critical after some time. Also students may not have deleted files which take up disk space and are useless to keep between sessions on a system terminal. Also, the more room that is made, the less often a Disk Save and Compress will have to be done. A typical session may look like:

```
> (MCR on Priviledged Account on the system console)
>PIP DL1:[20,*]*.MAC;*/DE/LD <CR>

        (List deleted files)

>PIP DL1:[20,*]*.OBJ;*/DE <CR>
>PIP DL1:[20,*]*.LST;*/DE <CR>
>PIP DL1:[20,*]*.*;*/PU <CR>
>BOO DL0:[1,51]DSCS 8 <CR>
RSX11S V2.2 BL26 DISK SAVE AND COMPRESS UTILITY V3.2
(IMPORTANT-Read the notes below first)
DSC58>DL1:/VE=DL0:/BL=12 <CR>
DSC-- 45 START VFY PASS

DSC58> BREAK KEY
001274
@ 773000G  (REBOOT SYSTEM)
. . . .
. . . .
> (MCR)
```

Notice the following:

1) MAC, OBJ, LST are intermediate results of a compilation that serve no purpose after an executable task is assembled.

2) The /LD on a /DE or/PU switch on PIP will List the Deletions if desired.

3) The /PU on PIP is only recommended if desperate for space. Students may have incomplete source codes with higher version numbers available, and if a Disk Save is done, that file is lost forever. Also, due to student editing errors, some files may not be closed properly.

4) TSK files are not usually deleted since the students will use up much disk space trying to generate another TSK.

5) VERY IMPORTANT, with students probably waiting to get on, a simple operation for speedy reboot is used. After the BOO, the OLD copy is placed in drive 1 and the PRESENT disk is in Drive 0. Thus the newly compressed disk will be ready and waiting in Drive 1. When the Disk Save is complete the

Drive 0 class disk is now the old disk and the system disk is now placed in Drive 0 and ready for a REBOOT of the system. That is, on the Disk Save and Compress command line, DL1: is the target and DL0: is the source.

6) When doing such a file deletion for a disk backup, the number of blocks freed up can in the thousands. Files are given an extension. That is, there are a number of blocks by which a file is extended beyond its allotted space. The disc save and compress automatically truncates all files after the end-of-file flag. If it appears that only a possible few hundred blocks are needed, type the following:

```
>PIP DL1:[20,*]*.*:*/TR <CR>
```

This removes all extensions from all files in group 20. This truncation will not work on files being currently created or edited by the Line or Text Editors.

7) Consider the PIP command lines. The DL1: are need only if the priviledged accounts UFD is not assigned to DL1:.

4.6. Other Commands

There are other commands available which may be useful to the curious or the expert.

```
>PAR <CR>
```

lists the partition locations and sizes of Terminal Drivers, Peripheral Drivers, the System, the file controller partition and others.

```
>TAS <CR>
```

list the partitions to various smaller tasks. These include MCR, MOU, DMO, EDI, BOO, LOA, HEL, BYE and others.

```
>TAL <CR>
```

list a snapshot of various routines in the system, the terminals issuing the routine and other information such as priority of the running task or routine.

```
>ACT <CR>
```

lists the names of active tasks issued at the terminal where the ACT command was given. There are other specifying switches, but this is most often adequate. If many tasks are being run simultaneously, then to keep track of them one would use this command to see what is active,

```
>RMDEMO <CR>
```

is a dynamic mapping of the system. This is a most exciting routine. As memory is allocated used, as users log on and off, this routine shows how the computer system routes data, partitions memory and uses devices. The routine is self prompting.

5. Closing Remarks

This document is a supplement to a SYSGEN Operation. It gives an account of some work that has to be done outside of the SYSGEN itself. Further, the assembly of the MicroBot Driver is included since a new driver is needed for each new system. It will be highly desirable in the future that the Pascal Procedure calling Sequence be known to simplify usage of the MicroBot Driver.

The System Maintainance Notes are derived primarily from actions taken to keep the system running for heavy student use. There are numerous other actions that are available but the Notes are a listing of major points.

Contributions that will increase system efficiency and capacity are encouraged and always welcome.