Models 31061, 30400, 30401, 30402, 30403, 30404
31010, 31011, 31012, 31013, 31014, 31015
31062, 31047, 31048, 31049, 31050, 31051
31080, 31074, 31075, 31076, 31077, 31078

1 MV/UX Release

This is the release notice for revision 3.00 of the MV/UX product. This release notice includes an overview of the MV/UX product structure, features and enhancements of this release, hardware and software prerequisites, documentation references, and installation procedures.

This notice covers several model numbers for the same basic product. The models differ in that they provide various limits on the maximum number of users supported by the system. For each user limit, there are two distributions, one for international release (without certain encryption functions; see "Notes and Warnings" later in this notice) and one for release in the United States. The contents of the release media are described in this release notice in "Product Summary" and "Product Organization" later.

Following is a list of the part numbers for each model number. The letter in parentheses after the model number indicates the media type. H refers to high density tape, G refers to 96 tpi mini diskettes, and C refers to cartridge tape.

MV/UX Model-Part Number Matrix

Model	Part Numbers	User Limit
Release Notice	085-000454-06	
U.S. Release, Commercial Models		
30404 (H,C)	071-1151-00, 071-0858-02	1-64+
30403 (H,C)	071-1151-00, 071-0859-02	1-64
30402 (H,C)	071-1151-00, 071-0860-02	1-32
30401 (H,C)	071-1151-00, 071-0861-02	1-16
(G)	081-600005-00 to 081-600023-00, 081-600027-00, 081-600024-00	
30400 (H,C)	071-1151-00, 071-0862-02	1-8
(G)	081-600005-00 to 081-600023-00, 081-600026-00, 081-600024-00	

31061	(H,C)	071-1151-00, 071-1152-00	1
	(G)	081-600005-00 to 081-600023-00, 081-600025-00, 081-600024-00	

U.S. Release, Educational Models

31015	(H)	071-1151-00, 071-0858-02	1-64+
31014	(H,C)	071-1151-00, 071-0859-02	1-64
31013	(H,C)	071-1151-00, 071-0860-02	1-32
31012	(H,C)	071-1151-00, 071-0861-02	1-16
31011	(H,C)	071-1151-00, 071-0862-02	1-8
	(G)	081-600005-00 to 081-600023-00, 081-600026-00, 081-600024-00	
31010	(H,C)	071-1151-00, 071-1152-00	1
	(G)	081-600005-00 to 081-600023-00, 081-600025-00, 081-600024-00	

International Release, Commercial Models

31051	(H,C)	071-1150-00, 071-0858-02	1-64+
31050	(H,C)	071-1150-00, 071-0859-02	1-64
31049	(H,C)	071-1150-00, 071-0860-02	1-32
31048	(H,C)	071-1150-00, 071-0861-02	1-16
	(G)	081-600005-00 to 081-600022-00, 081-600027-00, 081-600024-00	
31047	(H,C)	071-1150-00, 071-0862-02	1-8
	(G)	081-600005-00 to 081-600022-00, 081-600026-00, 081-600024-00	
31062	(H,C)	071-1150-00, 071-1152-00	1
	(G)	081-600005-00 to 081-600022-00, 081-600025-00, 081-600024-00	

International Release, Educational Models

31078	(H)	071-1150-00, 071-0858-02	1-64+
31077	(H,C)	071-1150-00, 071-0859-02	1-64
31076	(H,C)	071-1150-00, 071-0860-02	1-32
31075	(H,C)	071-1150-00, 071-0861-02	1-16
31074	(H,C)	071-1150-00, 071-0862-02	1-8
	(G)	081-600005-00 to 081-600022-00,	

		081-600026-00, 081-600024-00	
31080	(H,C)	071-1150-00, 071-1152-00	1
	(G)	081-600005-00 to 081-600022-00, 081-600025-00, 081-600024-00	

C release media--cartridge tape
H release media--magnetic tape reel
G release media--floppy disks (96 TPI)

The release notice contains the following sections:

2 Product Summary

MV/UX is the implementation of the System V UNIX (Trademark of AT&T Bell Laboratories) hosted on the Data General AOS/VS operating system. Data General Corporation has acquired the original source material by license from AT&T Technologies. In doing so Data General assumed the responsibility to prohibit its transmittal to unauthorized parties. Therefore this system is to be used and viewed only where specific authority to do so has been granted.

For the most part MV/UX is a collection of executable object files; source releases are available as separate models. The executables are the 'shell' and the commands. Included also are selected UNIX libraries for use when building C programs, on-line documentation, CLI macros to aid in the installation process, and include files for building C programs.

The MV/UX product is arranged in the following directory structure. The slash (/) replaces the AOS/VS colon (:) as the filename separator.

MV/UX Directory Structure

```
/bin   -  program files and command scripts.
/etc   -  essential data.
/lib   -  object libraries.
/tmp   -  temporary files (initially empty).
/usr   -  general purpose directory.
      /bin  - initially empty; recommended for user commands.
      /include   - standard include files.
      /lib  - utilities, miscellaneous
            /cron   - files cron and related commands
            /ctrace - contains source file for ctrace(1)
            /help   - help files for help(1).
            /lex    - lex(1) files
            /spell  - spelling program spell(1) files
            /terminfo - terminal database for curses(3x)
            /uucp   - uucp demons, logs, and files
      /mail - receptacle used by mail(1) (initially empty).
      /man  - MV/UX User's Manual sections.
            /docs   - Reference documents.
            /man0   - Preface, appendixes, etc.
            /man1   - Command documentation.
            /man2   - Emulator routines documentation.
            /man3   - Library routines documentation.
            /man4   - Special files.
            /man5   - File Formats.
            /man6   - Miscellaneous Facilities.
      /news - receptacle for news(1) items (initially empty).
      /preserve - storage used by vi and ex (initially empty).
      /tmp  - like /tmp (initially empty).
      /src - contains input files for the terminfo compiler, tic(1)
             (Source licensees will put emulator and command
              sources in this directory.)
      /<usernames> - username directories.
```

3 Product Organization

All directories and files included in the release are listed in the file tapelist, which is included with a machine-readable copy of the release notice (file mvux_rn) on the release medium (see below for location on the tape or diskette). To allow timely updates to be made to both MV/UX and the language utilities, the common language runtime libraries and the C compiler are packaged separately and are described in separate release notices; you will receive these components as part of your MV/UX order. Note that both should be loaded immediately after the installation described later in this document. Approximately 24000 disk blocks (12.3 megabytes) of disk space will be needed to load the basic product; the online manual requires about 4000 disk blocks (2 megabytes) more; the source code requires about 15000 disk blocks (7.7 megabytes) more. Depending on the model ordered, the product will be delivered on one of the following media:

H - standard magnetic tape (1200' reels)
C - cartridge tape (8" cartridges)
G - floppy disk (5 1/4", 96 track-per-inch diskettes)

Tape Organization

The organization of the basic product on tape is as follows:

TAPE 1
- File 0 basic file structure and commands
- File 1 Security administration package; this material is NOT INCLUDED in the international distribution.

TAPE 2
- File 0 MV/UX inner ring emulator program file (etc:system:ire.pr and etc:system:ire.st) and library (lib:libire.lb).
- File 1 MV/UX demon (bin:demon.[user_limit].[revision].pr) and demon link (bin:demon.pr).
- File 2 on-line documentation, release notice and tapelist

If you have ordered the source code distribution, you will receive a third tape containing the source code. Its organization is as follows:

TAPE 3
- File 0 MV/UX source directory structure (usr:src) and code for most libraries, the MV/UX commands, the inner ring emulator, the inner ring library, and the demon. (The source code for AOS/VS C is distributed separately.)
- File 1 Security administration sources; this material is NOT INCLUDED in the international distribution.

Labeled Diskette Organization

The labeled diskettes contain labels that conform to the proposed ANSI standard for labeled diskettes, #7665, at the Basic Interchange level. Each diskette has a volume id, which is noted on the external label.

The MV/UX diskette distribution is patterned after the tape distribution, which consists of 5 CLI DUMP files across 2 tapes. Each DUMP file from the tape distribution is placed on a series of one or more diskettes, labeled MVUX00 to MVUXnn. Each diskette has a filename that reflects which tape and which DUMP file it corresponds to.

See the "Installation" section for loading the MV/UX diskette release.

The organization of the product on diskettes is as follows:

Files corresponding to TAPE 1:
- TAPE1_FILE0 basic file structure and commands
- TAPE1_FILE1 Security administration package; this material is NOT INCLUDED in the international distribution.

Files corresponding to TAPE 2:
- TAPE2_FILE0 MV/UX inner ring emulator program file (etc:system:ire.pr and etc:system:ire.st) and library (lib:libire.lb).
- TAPE2_FILE1 MV/UX demon (bin:demon.[user_limit].[revision].pr) and demon link (bin:demon.pr).
- TAPE2_FILE2 on-line documentation, release notice and tapelist

4 Environment

MV/UX runs on any of the MV family of computers meeting the minimum configuration requirements for AOS/VS rev 6.00 and AOS/VS C rev 3.21.

MV/UX 3.00 requires AOS/VS rev 6.00, Update 4, or later because the 3.00 emulator relies on AOS/VS support for pipes introduced in AOS/VS 6.00. Update 4 contains fixes to some pipe problems (patch number 6.04.06).

MV/UX 3.00 requires AOS/VS C rev 3.21 or later, and the AOS/VS Common Language Library (LANG_RT.LB) rev required for the C rev being used (see the AOS/VS C release notice). Revision 3.21 of AOS/VS C requires revision 3.40 of AOS/VS LANG_RT.

Other implicit requirements are that all pieces of the AOS/VS release be consistent. Any language product used must be qualified for the revision of AOS/VS in use. Any supporting product required by the language product must be the minimum revision required by the language product.

5 Enhancements

The MV/UX 3.00 emulator provides several important product improvements:

* All AT&T based MV/UX commands and libraries have been upgraded to System V, Release 2, Version 2.

* The emulator code now runs in an inner ring. This means that code for system call support is no longer linked into each executable file; consequently, each command in MV/UX (and each user program built against the 3.00 inner-ring emulator) is smaller by about 35 KB. Since executables are smaller, there is less paging traffic, which improves system throughput.

* Piping is now done with the support of the AOS/VS AGENT. This means that MV/UX users can now put FORTRAN, PASCAL, etc., programs in a pipeline. Previously, only C programs could be easily used in pipes. The 2.00 and 3.00 piping implementations are NOT COMPATIBLE, however.
  This means that any programs built with the 2.00 emulator must be relinked before they can be run in pipelines with 3.00 commands.

* New MV/UX system calls have been added: fork and lockf calls are now supported. Fork is supported for the sake of compatibility so that users can quickly port programs that are not easily adapted to use vfork. For performance reasons, fork should be avoided in process-intensive contexts; if you have programs with this characteristic, you will probably want to replace the fork with vfork.

* A more UNIX-like environment is now supported.

    * An MV/UX process may now kill other processes spawned by the same user name, whether they are in the same process tree or not. E.g., if user "Joe" has a runaway process that has control of his terminal, he can log on as "Joe" at another terminal and kill the runaway. Previously, "Joe" would have needed super-process privileges to do this. Note that since this feature is only available to MV/UX processes, killing of ordinary AOS/VS processes (e.g. a CLI process) is not supported.

    * The new emulator provides support for more ioctl(2) (and thus stty(1)) options, so that users can use non-Data General terminals by changing control character assignments and carriage return/new line mapping.

The new ioctl(2) options are provided via a new line discipline, BELL_LD. The line discipline supported in older versions of MV/UX is still supported under the name AOS_LD, for those programs which require it. Requests for new ioctl(2) options will be silently ignored in AOS_LD. A description of the features of the line disciplines may be found in the file /usr/man/man4/termio.4.

Note that the new line discipline settings work only in the MV/UX context; if a user goes to the CLI, his terminal I/O is no longer mediated by the MV/UX emulator, but is handled by the AOS/VS peripherals manager (PMGR), which does not allow this capability.

* Note that some text formatting commands have been removed from the basic MV/UX product as a result of the upgrade to System V Release 2 source code. This tracks the AT&T unbundling of what is called the Documenter's Workbench (DWB). This means that nroff and tbl are now available only with a DWB license. The revision 2.00 nroff and tbl can be used, keeping in mind the restriction on using 2.00 programs in pipes. This restriction implies that "nroff | col" will not work. Use "nroff >tmpfile;col <tmpfile" instead.

* cron(1m) - and related batch commands (at(1), batch(1), crontab(1), cron(1m)) have been added.

* ctrace(1) - c program debugging command has been added.

* fork(2) - Implementation of the 3.00 inner ring emulator has permitted inclusion of the fork(2) system call in MV/UX. Fork implements all the features of UNIX fork, and is provided primarily for source compatibility. Vfork is much faster, and should be used whenever possible.

* lockf(2) - file/record locking system call has been added.

* lp(1) - new line printer commands (lpstat(1), lpadmin(1m), lp(1), cancel(1)) have been added. The lpr command, of previous revisions of MV/UX, is replaced by the functionally equivalent lp command as in System V.2.

* mknod(2) - was enhanced to support creation of named pipes.

* mvux_form.cli - this command is the same as unix_form.cli, which is being phased out. (See Notes and Warnings.)

* nohup(1) - command has been added.

* rsh(1) - the shell now supports invocation as a restricted shell. (See Notes and Warnings.)

* setuid(2) - A call similar to the UNIX setuid(2) has been provided. MV/UX setuid is limited to execution between fork (or vfork) and an exec, to specify the user ID to be given the exec'ed process. If the user does not have change username privilege in the AOS/VS user profile, the new process will have the original user ID.

* tar(1) - now automatically determines blocking factor on an extract. You may extract tar tapes with unknown blocking factor. Also, tar now supports archiving files of type PRG and PRV. (See Notes and Warnings.)

* tic(1m) - command is now supported. Tic compiles terminfo source files and places results under /usr/lib/terminfo.

* times(2) - tms structure has been changed to have four elements rather than two. Reference times on-line manual page.

* uucp(1) - related commands (uudecode, uuencode, uupick, uuto) have been added.

6 Notes and Warnings

This section contains information about MV/UX usage and behavior which all users should be made aware of.

6.1 Known Deficiencies

* cc(1) - When invoked such that its output file is a pipe, will abort from the link step. This is a problem with the AOS/VS linker, and is under investigation.

* csh(1) - The csh command still occasionally aborts with the message "assertion botched: p==alloct". This problem is under investigation.

* lpr(1) - This command is obsolete; replaced by lp(1)

* read(2) - Only one task within a process can read from any console at one time. Other tasks will pend in the read system call until the read completes. This is due to an AOS/VS bug, which is currently under investigation.

* stat(2) - Making this system call on a file of type FIFO does not return the proper file type. This is under investigation.

* tar(1) - The -a switch, an MV/UX extension, creates a tape that gives directory checksum errors when extracted. This is under investigation.

* nohup(1) - Note that the change username, superuser, and superprocess privileges are not passed to nohup processes. Use "qbatch su op <scriptfilename>" to run privileged processes in batch.

* A shell command line of the form "cc foo.c -o foo | tee outfile" will abort in the link step. This is a problem with the AOS/VS link utility used in pipes, and is under investigation.

6.2 International Distribution

In order to conform to Federal Export Regulations, the model numbers designated for international distribution do not include certain encryption capabilities. The international version contains a limited crypt(3) function, and versions of ed, ex, and vi that do not support a decryption option. The crypt(1) command is not included in the international version.

| Model numbers designated for US distribution include the same material as the international version, plus versions of ed, ex, and vi with decryption options, and a crypt(1) command.

6.3 MV/UX Implementation Characteristics

When using MV/UX you should pay particular attention to the semantics of facilities that are similar but not identical in both AOS/VS and other UNIX systems. MV/UX is a bridge between the two systems and sometimes implements a hybrid capability. Efforts have been made to identify these situations in the documentation and the pertinent manual pages should be reviewed. This section mentions some of the more obvious cases.

AOS/VS C

The AOS/VS C product includes a system call emulator library that offers some UNIX system call interfaces to C programs that do not require the more complete services of the MV/UX inner-ring emulator. The cc and ld commands executed in the MV/UX context by default build programs using the inner-ring emulator. If you build programs using the AOS/VS CC and CCL macros, you will not be using the inner-ring emulator and will notice these differences:

* your executable will be larger by about 30-35 KB.
* fork and lockf interfaces will not be available to your program, and ioctl functionality will be less complete.

Character Interpretation

Certain characters have semantics that are peculiar to the command line interface or operating system. File names are obvious examples. Other examples include meta-characters for grouping and quoting, filename generation with templates, and case sensitivity. It is important to know when and how special characters are interpreted. References: sh(1), cli(1), csh(1).

CPU Time Limit

The CPU time limit switch on AOS/VS batch streams must not be set since MV/UX relies heavily on creating processes without blocking the creating process.

Errors

MV/UX system call errors are available to C programs in the external variable 'errno'. The $? shell variable, or the $status csh variable, contains the most recently executed command's exit status. Typically, user program exit codes are in the range 0 to 127. Exit codes in the range 128-255 have special meaning. They represent the signal number of the signal taken. For instance, exit code 139 indicates that signal 139-128=11 was taken. The signal(2) manual entry reveals that signal 11 is a segmentation violation, which is equivalent to an "Inward Address Trap" on AOS/VS.

When executed from the CLI, MV/UX programs which return a non-zero status will cause an appropriate CLI warning message to be generated.

exec

exec(2) will not directly execute shell scripts. An interim solution is to prefix the execute string with the pathname of the shell. For example, instead of
execl ("/bin/scriptname","scriptname","scriptarg",0) use
execl ("/bin/sh","sh","/bin/scriptname","scriptarg",0)

Process IDs

Process IDs are recycled on AOS/VS. They are unique only for the life of a process, and thus are not ideal for creating unique filenames. Use of the $$ shell variable to create "unique" filenames may not work the same as on UNIX systems.

Files

Both systems support a hierarchical file system with compatible file naming, access permissions, and name resolution rules. Both offer a variety of file manipulation commands and system calls which perform basically equivalent operations.

File name suffixes are significant in AOS/VS while not so significant in a UNIX system. MV/UX again is a bridge and attempts to implement a compromise. Program files have the .pr suffix, shell scripts are suffixed with .sh (optional), object files are suffixed with .ob, and CLI macros have a .cli suffix.

Files created by MV/UX will be of type UNX (AOS/VS file type) when viewed by AOS/VS. Likewise, AOS/VS files are viewed as "ordinary files" by MV/UX. Files are "portable" and most file types may be read or written by both systems.

| Directories created under MV/UX are Control Point Directories when viewed by AOS/VS. Because UNIX does not support the notion of CPD's the maximum directory size is set to a large number, and for all practical purposes is unlimited. The use of CPD's provides for space accounting on a directory basis. The CLI SPACE command accesses this feature.
|
| Directories under UNIX have unlimited nesting depth. MV/UX has the AOS/VS imposed limit of 9 levels of nesting. This difference can cause problems when using tape to move files from UNIX systems to MV/UX systems.

MV/UX links are AOS/VS links in that they need not be "resolvable". That is, the link resolution name to which the link "points" need not exist, and the persistence of the link is independent of the link resolution name. However MV/UX links differ from AOS/VS links in that file status information refers to the link resolution name. AOS/VS does not resolve links in the FILESTATUS command, for example. Also note that MV/UX allows links to directories whereas UNIX does not.
| References: ls(1), sh(1), amove(2), glink(2), stat(2), comb(1).
|
| Due to these characteristics of links on AOS/VS, tar and cpio archive the actual file the link points to rather than the link. If that file does not exist, an error message is generated. When a tar or cpio tape created on an MV/UX system is loaded on MV/UX or DG/UX, where there were originally links, there will be complete copies of the original files. You could end up with several copies of the same file with different names. If the file is very large, this could present a space problem.

Mountable File Systems

Logical disk units (LDU's) may be mounted and dismounted using the AOS/VS facilities. Mountable file systems are not supported through the MV/UX interfaces. Reference: AOS/VS Operator's Guide.

Multitasking

Note that revision 3.00 and later of the AOS/VS C compiler supports multitasking capabilities. MV/UX users should be aware that any code using static data (eg., returning a pointer to a static array) cannot reliably use that data in more than one task at a time; the locking functions mfinit, mlock, and munlock must be used to prevent conflicting accesses. Consult the C documentation on mfinit, mlock, and munlock for more detail and a suggestion on how to use the standard I/O functions in a multitasking program.

Process Priorities

Setting the AOS/VS process type and process priority are supported using AOS/VS facilities. There are no MV/UX interfaces for modifying a process's priority or type. However, when a MV/UX program "procs" another process, that process is given the same process type of the parent process, rather than a constant default process type. Processes started with the nohup(1) command are priority 3, a lower priority than most other processes on the system.

Root Directories

UNIX requires that certain file system names reside in predefined places, namely the root directory (/). Reference installation instructions.

Search Rules

MV/UX supports a hybrid search algorithm supporting the 'shell' pathlist and name resolution, and also offers AOS/VS searchlists. Note that the CLI SEARCHLIST command both displays and sets the searchlist. The shell sea command sets the searchlist, and the psl command displays the searchlist. References: sh(1), sea(1)

Restricted Shell

The rsh(1) command is a means of setting up user profiles with limited access to a system. The system administrator should note that certain commands can be used to execute an unrestricted shell from a restricted shell. These include more, vi, and csh. The MV/UX extension commands x, and cli can also be used to execute an unrestricted CLI process. These commands should be left out of the restricted bin directory. The red(1) editor is safe to put in a restricted bin because it only executes a restricted shell.

Shared File Pointers

In a UNIX system the current file position is shared by all users who have a common file descriptor (i.e., they see each other's appends). Offspring processes share their parent's file position, for example. AOS/VS does not directly support this sharing, so MV/UX will not be able to demonstrate this sharing in all cases.

One technique now available in MV/UX 3.00 may help in some cases. When doing a fork/vfork to be followed by an exec, often the child will inherit the same standard output. Before the fork/vfork, the

| father should close(2) and open(2) standard output file, specifying
| O_APPEND in the open flag. Each subsequent write done by the father
| and child will first seek to the end of the file, and they will no
| longer overwrite each others data.

Magnetic Tape Transfer Sizes

On many UNIX systems a default block size of 10240 is used when writing to magnetic tape. In system generation for AOS/VS, the default maximum transfer size for tape drive type MTB is 8192. If this is left as the default, MV/UX tape operations may fail. To avoid this difficulty, systems on which MV/UX is installed should set up
| their tape systems to handle a maximum tape transfer of 10240 (10K).
| (MV/UX tar will work fine if smaller block sizes are specified, but
| will be unable to read and make tapes with greater than 8192 block
| size unless 10240 is specified in AOS/VS generation.)
|
|
| tar
|
| tar(1), the tape archiver, may now be used to dump and load files of
| type PRG or PRV. Since the notion of file type, in the AOS/VS sense,
| is not supported by standard tar format, the file's type is not
| preserved across a tar dump-load. All files loaded by tar will have
| type UNX. We recommend that you use AOS/VS DUMP/LOAD for "non-UNX"
| program files.
|
|
| uucp
|
| The demons should be started several times a day, depending upon the
| urgency and volume of UUCP requests. The UUCICO demon may need to be
| run at priority 1 to minimize the loss of data from the PMGR. Use
| cron(1) to run the uucp demons.
|
| UUCP will work only with one type of autodialer at a single baud rate
| at a time.
|
| The MV/UX extension of 31 character filenames and 255 character
| pathnames is permitted by UUCP. Users should limit filenames and
| pathnames to the maximum allowed by the target machine. DG/UX, and
| Berkeley file system based UNIX systems, allow longer filenames than
| the usual 14 characters of older UNIX versions. If there is doubt
| about the limitations of the target machine, limit filenames to 14
| characters in length and pathnames to 99 characters.
|
| Bell dialers have not yet been tested.

| A login entry must be set up for UUCP with the AOS/VS Preditor
| program. See the configuration instructions in this memo.

6.4 Future Changes

Though revision 3.00 supports the use of pipes among programs compiled and linked with revision 2.00 of MV/UX, this support will be dropped in a future revision. Therefore, any user programs which are used in pipes should be recompiled and linked with revision 3.00 so they will work with future revisions of MV/UX.

The name of the unix_form.cli command is changing to mvux_form.cli in the next revision of MV/UX. Both names are supported in revision 3.00 to facilitate conversion to the new name.

7 Fixes

7.1 Fixes in Revision 3.00

* awk(1) - will now handle 1-character patterns correctly.

* cc(1) - F77 link operations will no longer generate FMTMM undefined error messages. The -P switch now works correctly.

* col(1) - will now process nroff output correctly.

* csh(1) - will no longer abort with "event not found" error, and will no longer loop when interrupted with ^C^A in a while loop. Aborting user programs will now cause an appropriate error message to be printed. Also, the $status variable is set to the same value as the sh(1) $? variable gets for the same type of abort. (See Notes and Warnings.) The onintr command will no longer traceback from csh.

* dd(1) - command will now output to tape using the specified block size.

* delta(1) - no longer deletes the p-file on first checkin of multiply gotten files.

* diffmk(1) - now works when invoked from the csh.

* f77(1) - should now return errors from the link step.

* find(1) - now can distinguish between links and non-links.

* find(1) - will now handle "-exec rm" correctly, will now process keys properly, and will now work through links properly.

* fopen(3) - fopen will now correctly open /dev/tty.

* fork(2) - system call has been added.

* lint(1) - will now be able to handle more macro #defines so it will handle files like paru.h.

* lpr(1) - has been supplanted with lp(1), and lp fixes outstanding bugs against lpr(1).

* make(1) - no longer continues to the link phase upon syntax errors.

* pipe(2) - With the new AOS/VS based implementation, the occasional pipe hang problems are cleared up.

* qbatch(1) - now works properly when the initial login process is csh.

* sh(1) - now only attempts execution of files with "x" permission set, will no longer abort with some shell scripts, and repeated ^C^A interrupts no longer cause a System Ring Trap. The wait command can be interrupted with ^C^A. The shell no longer returns its own PID upon attempt to execute a non-existent command in background.

* su(1) - will now properly check passwords from /etc/passwd when present.

* tar(1) - now supports ownership for incoming files, and generates fewer spurious error messages.

* vi(1) - now correctly handles writing of a file after command errors are encountered.

* who(1) - now works correctly.

8 Documentation

Published documents for the MV/UX system are:

MV/UX User's Manual	093-701001
Support Tools Guide for the MV/UX and DG/UX Systems	093-701009
Learning To Use Your MV/UX System	093-701000

8.1 Changes

This section describes the cumulative documentation changes since the last manuals were printed.

8.1.1 Documentation Changes for Revision 3.00

Additional copies of this release notice may be obtained by loading the file mvux_rn from the release tape (see Installation).

The online user's manual reference pages, MAN1 through MAN6, have been updated; the introductory information in MAN0 has not been updated for revision 3.00.

The following manual pages are new (*) or updated for MV/UX 3.00:

* at.1	intro.2
cc.1	ioctl.2
* cron.1m	kill.2
* crontab.1	link.2
* ctrace.1	* lockf.2
* dc.1	lseek.2
find.1	mknod.2
* lp.1	nice.2
* lpadmin.1m	open.2
* lpstat.1	pause.2
* nohup.1	pipe.2
sh.1	* plock.2
stty.1	read.2
* tic.1m	* setpgrp.2
* uuencode.1c	* setuid.2
* uuto.1c	signal.2
access.2	stat.2
alarm.2	sync.2
amove.2	time.2
brk.2	times.2
chdir.2	umask.2
chmod.2	uname.2
chown.2	unlink.2

```
          close.2                  utime.2
          creat.2                  vfork.2
          dup.2                    wait.2
          exec.2                   write.2
          exit.2                   malloc.3x
          fcntl.2                * curses.3x
        * fork.2                 * terminfo.4
        * getpgrp.2              * termio.4
          getpid.2               * tty.4
          getuid.2               * fcntl.5
                                 * uuencode.5
```

8.2 Related Manuals

Other related documents are:

AOS/VS Programmer's Manual	093-000241
AOS/VS Operator's Guide	093-000244
AOS/VS SED Text Editor	093-000249
Learning to Use Your AOS/VS System	069-000031
Managing AOS/VS	093-000243
SWAT Debugger User's Manual	093-000258
AOS/VS Debugger and File Editor User's Manual	093-000246
AOS/VS Macroassembler (MASM) Reference	093-000242
AOS/VS LINK and Library File Editor	093-000245
AOS/VS Command Line Interpreter (CLI) User's Manual	093-000122
FORTRAN 77 Reference Manual	093-000162
C Language Reference and Runtime (AOS/VS)	093-000264
AOS/VS C Release Notice (included on C tape)	085-000294

9 Installation

9.1 Basic Product Installation

Please note that previous revisions of MV/UX were LOADed using the /NACL switch, causing all files to receive the default ACL of the installer. Beginning with revision 3.00, MV/UX is shipped with all ACLs set appropriately, so the /NACL switch is NO LONGER USED.

MV/UX requires that certain file system names reside in predefined places. DEV, BIN, ETC, LIB, TMP, and USR must exist in the root, ":", either as directories or links to directories.

If you have not previously installed the C or MV/UX products and use one of these names for another purpose, you must resolve the conflict and make the names available for MV/UX use. If MV/UX is to be loaded onto an LDU or directory other than the root, then the aforementioned names should be placed as links in the root.
For example [CRE/LINK BIN :UNIX:BIN]. This must be done for BIN, ETC, LIB, TMP, and USR.

This installation procedure is both for an initial installation (on a machine not currently running revision 2 of MV/UX), and for a re-installation (on a machine currently running revision 2 of MV/UX). If this is a re-installation, there are 2 possible approaches:

* Terminate all use of MV/UX and terminate the MV/UX demon, and load revision 3 in the directory containing the revision 2 software. This is re-installation over an existing MV/UX directory hierarchy. Once this is done, return to revision 2 would require reloading.

* If there is enough spare disk space (about 24K blocks without manual pages, 28K blocks with), you may wish to load and set up revision 3 into its own directory. This is re-installation in a new directory hierarchy. This can be done for the most part without affecting the currently running MV/UX system. You can make users' home directories appear under the new hierarchy by creating links to their current locations. You can move the :ETC:PASSWD file to the new hierarchy, along with any other needed files. When everything is set up, you can terminate users and the old demon, and create links in the ":" directory (for BIN, ETC, USR, and LIB), that point to those components of the revision 3 directory hierarchy. This affords some flexibility in phasing out the older revision. You could switch between revisions by changing links in ":" and terminating and restarting the appropriate MV/UX demons.

If this is a reinstallation, we strongly recommend installing MV/UX 3.00 in its own directory hierarchy. This is due to the fact that loading over an existing revision 2 hierarchy will not cause the ACLs of existing directories to be changed to the new settings. Also, obsolete revision 2 directories, such as usr/lib/uucpsys, will have to be deleted by hand.

The following procedures are used to install the initial MV/UX release or subsequent releases. The brackets denote CLI commands.

1) Read this notice in its entirety before proceeding.

2) If this is a re-installation over an existing MV/UX directory hierarchy, all users should be removed and the MV/UX demon program should be terminated. [SUPERPROCESS ON; TERM OP:UNIX; SUPERPROCESS OFF]

3) Insure that your searchlist allows you to find the various AOS/VS utilities such as an editor, MASM, LFE, LINK, etc. These products reside in :UTIL on most systems.

4) Become a superuser. [SUPERUSER ON]

5) Make your current working directory the LDU or directory into which you wish to load MV/UX. Thus if you wish to install MV/UX on your system's root pack you would enter [DIR :]. If you have a LDU initialized as UNIX (or just a directory named UNIX), the command would be [DIR :UNIX].

6) Before executing the next step take careful note of this warning: we use the /DEL switch in case this is a re-installation over a previous revision. Its purpose is to OVERWRITE whatever previously existed. THE USER SHOULD NOTE THAT IF ANY PREVIOUSLY INSTALLED FILES HAVE BEEN ALTERED, THEN THOSE ALTERATIONS WILL BE LOST UNLESS THE MODIFIED FILES ARE SET ASIDE UNTIL AFTER THE LOAD IS COMPLETED. Note that files like /etc/passwd and /etc/profile are not on the tape; they will not be overwritten.

   After saving any files you don't want overwritten,

7) If you are using 96 tpi mini diskettes, skip this step.

   a) Mount release tape 1 on a drive capable of reading the release media (cartridges or 1600 bpi magnetic tape), and position the tape at BOT.

   b) Load file 0 on the tape.
      [LOAD/BUFFERSIZE=8192/DEL @MTxn:0]
      Where x is the drive type, n the drive number. This

loads the MV/UX subordinate directory structure, and most commands and data files.

c) THIS STEP DOES NOT APPLY TO INTERNATIONAL DISTRIBUTIONS.

If you want the security administration package, with release tape 1 on the drive, load file 1.
[LOAD/BUFFERSIZE=8192/DEL @MTxn:1]

This step loads relevant parts of the BIN, LIB, and USR directories, overwriting the international versions of these materials.

d) Remove release tape 1 and mount release tape 2 on the tape drive.

e) Load file 0.
[LOAD/BUFFERSIZE=8192/DEL @MTxn:0]
This loads the new inner ring emulator.

f) Load file 1.
[LOAD/BUFFERSIZE=8192/DEL @MTxn:1]
This loads the MV/UX demon.

g) If you want to load on-line manual pages (and release notice and tapelist), with release tape 2 on the tape drive, load file 2.
[LOAD/BUFFERSIZE=8192/DEL @MTxn:2]

8) Do this step if you are using 96 tpi diskettes.

a) Insert the first diskette, labeled MVUX00:TAPE1_FILE0, in unit 0 of the diskette drive.

b) Enter the following CLI commands:

[OPERATOR ON]
[LOAD/V/DEL @LFD:MVUX00:TAPE1_FILE0]

c) The LOAD command responds:

PLEASE INSERT A DISKETTE IF NOT ALREADY INSERTED.
UNIT [@DPJ10] VOLUME ID [MVUX00] ? [Y]

Press newline.

d) The diskette will begin loading. A date and time line will appear, followed by the list of files loaded. For example:

```
21-AUG-85 10:01:28
  file1
  file2
   ~
   ~
   ~
  lastfile
```

e) A prompt will appear for the next diskette in the sequence:

```
PLEASE INSERT NEXT DISKETTE.
UNIT [@DPJ10] VOLUME ID [MVUX01] ? [Y]
```

Press newline. The volume id increments for each diskette. Each time this message appears, insert the appropriate diskette, and hit newline.

f) When all diskettes for the series are loaded, the CLI responds with:

```
PLEASE REMOVE THE DISKETTE.
```

g) THIS STEP DOES NOT APPLY TO INTERNATIONAL DISTRIBUTIONS.

If you want the security administration package, place the diskette labeled MVUX00:TAPE1_FILE1 in the drive and execute the following command line:

```
[LOAD/V/DEL @LFD:MVUX00:TAPE1_FILE1]
```

Proceed as in steps c-f above.

This step loads relevant parts of the BIN, LIB, and USR directories, overwriting the international versions of these materials.

h) Place the diskette labeled MVUX00:TAPE2_FILE0 in the drive.

```
[LOAD/V/DEL @LFD:MVUX00:TAPE2_FILE0]
```

Proceed as in steps c-f above.

This loads the new inner ring emulator.

i) Place the diskette labeled MVUX00:TAPE2_FILE1 in the drive.

[LOAD/V/DEL @LFD:MVUX00:TAPE2_FILE1]

Proceed as in steps c-f above.

This loads the MV/UX demon.

j) If you want to load on-line manual pages (and release notice and tapelist), place the diskette labeled MVUX00:TAPE2_FILE2 and execute the following command line:

[LOAD/V/DEL @LFD:MVUX00:TAPE2_FILE2]

Proceed as in steps c-f above.

k) Enter the following command to turn operator mode off: [OPERATOR OFF]

9) The file ETC:UTSNAME is required for the proper operation of the uname(2) system call and uucp. A prototype, ETC:UTSNAME.PROTO, is provided.

If this is an initial MV/UX installation, copy this file to ETC:UTSNAME, and edit it for your local installation.

If this is a re-installation in a new MV/UX directory hierarchy, you should copy the existing utsname file to the new etc directory [COPY ETC:UTSNAME :ETC:UTSNAME].

10) Certain MV/UX system routines and commands require information from the file ETC:PASSWD. All users who are to use MV/UX must be represented in this file. To use the ~user feature of csh, you must update the ETC:PASSWD entries to list the user's home directory: for example, the entry 'smith::12:2:::' becomes 'smith::12:2::/udd/smith:' Please note also that ETC:PASSWD.PROTO now contains certain required entries for users root, op, lp, uucp, and bin. If your system already has a copy of :ETC:PASSWD which you plan to use, you will have to add these entries.

If you have previously performed the AOS/VS C or MV/UX installation procedure then the :ETC:PASSWD file already exists.

If this is a re-installation in a new MV/UX directory hierarchy, you should copy the existing passwd file to the new ETC directory [COPY ETC:PASSWD :ETC:PASSWD], and edit in the entries for root, op, lp, uucp, and bin from ETC:PASSWD.PROTO.

If :ETC:PASSWD does not exist, then the file ETC:PASSWD.PROTO may be copied to ETC:PASSWD and edited. The format of the

file is one user entry per line. Each line is terminated by a new-line and each entry is delimited by a colon (:). Some entries will be null; see passwd(6) for a description of the file format. An explanatory entry is as follows:

```
name::id:::home:
```

Where name is the user name as known by AOS/VS and id is a simulated MV/UX user id which should be unique for each user.

```
example:  op::0:::/udd/op:
```

11) On AOS/VS systems user home directories reside in :UDD. MV/UX similarly implements home directories in :USR. Actually the MV/UX user home directories may be located in either place. However, the user home directory should be made to appear in both directories by using a link from one to the other. For example, if :USR is the "user directory" containing the user home directory XXX, :UDD should contain a link (named XXX) to :USR:XXX. [CRE/LINK :UDD:XXX :USR:XXX]

12) When the 'shell' is initialized when logging on and under certain other circumstances, it attempts to execute two initialization scripts. The first is a system-wide script named ETC:PROFILE. As with ETC:PASSWD the initial version of ETC:PROFILE is supplied as ETC:PROFILE.PROTO. If this is your first installation, ETC:PROFILE.PROTO should be renamed ETC:PROFILE. Its contents should be reviewed to see if your particular installation requirements differ from the supplied default. The second script created and maintained by the user in the user's home directory and is named :USR:<username>:.PROFILE. This file may be referenced in the 'shell' as $HOME/.profile.

13) MV/UX, like UNIX and most UNIX-derived systems, has a set of predetermined places where it locates program and data files. If there is a requirement to move some portion of the system to another location, then a link must be supplied in the vacated spot.

    Because of this policy MV/UX uses links in its standard places to locate those portions of other AOS/VS products that it requires. These links assume that these other AOS/VS products reside in a "standard" place. The installer at this time should enter the release directories BIN and LIB and do a filestatus command, [F/TYPE=LNK/AS/S] Each link should be verified to insure that the link resolution is as expected. Do this by applying the path command to each link file name; for example: [PATH linkname]. If any links are "unresolved" (i.e. target does not exist), these must be deleted and

replaced by correct links.

14) All users of MV/UX must be AOS/VS users also. Thus the AOS/VS utility PREDITOR must be invoked to enter each prospective MV/UX user into the system. If upon logging on the user wishes to come up directly in the 'shell', the user profile should specify the following:

```
initial ipc file [:etc:shell_logon.ipc]
program [:bin:sh.pr]
```

All MV/UX users must have the following privileges:

* CREATE WITHOUT BLOCK [YES]

* USE IPC [YES]

* UNLIMITED SONS [YES]

* CHANGE ADDRESS SPACE TYPE [YES]

15) If this is a re-installation in a new MV/UX directory hierarchy, and the old UNIX demon is still running, notify all MV/UX users that MV/UX will be down, then terminate it. [SUPERPROCESS ON; TERM OP:UNIX; SUPERPROCESS OFF]

16) Change directory to the root directory. [DIR :]

17) MV/UX requires that certain file system names reside in predefined places. DEV, BIN, ETC, LIB, TMP, or USR must exist in the root, ":", either as directories or links to directories.

If AOS/VS C or MV/UX has been previously installed, some or all of these will already exist. If you just loaded MV/UX onto an LDU or directory other than the root, then the aforementioned names should be placed as links in the root, pointing to the newly installed software.
For example [CRE/LINK BIN :MVUX:BIN]. This must be done for BIN, ETC, LIB, TMP, and USR.

Note that the empty directories TMP and USR:TMP are supplied on the release tape, so they are created on the same LDU as the rest of MV/UX. Since these directories are used in the pipe implementation, and for compiler and linker temporaries, and by many commands, there may be a performance advantage to placing TMP and USR:TMP on a separate LDU. This can be done using links in the root and USR directories respectively.

18) No matter where MV/UX is loaded, a link to :PER (named DEV) must exist in the root.
[CRE/LINK DEV :PER]

19) The MV/UX demon program must be active in order for UNIX software to run. To start the new MV/UX demon program, enter this command line at the system console: [:BIN:START_DEMON] To ensure that the demon is brought up each time the system comes up, you may add this same command line to the system initialization macro (usually a macro named UP.CLI).

20) The basic MV/UX system is now operational. If not already present, you must now load the C compiler and common language runtimes to make the cc and ld commands usable. Consult the release notices for these products for installation procedures.

9.2 Printer Configuration

A native UNIX system offers several commands to manage print queues and devices. MV/UX offers a subset of these commands (mostly those that the general user population would use), and relies on AOS/VS commands to provide the remaining printer management functions (typically performed by an operator).

MV/UX provides these commands:

lp-	send requests to an LP line printer
cancel-	cancel requests to an LP line printer
lpstat-	print LP status information
lpadmin-	modify LP configuration

MV/UX DOES NOT provide commands in the following list; AOS/VS EXEC commands can be used to perform some of these functions; these equivalencies are also listed.

accept-	allow LP requests
reject-	prevent LP requests
disable-	disable LP printers
enable-	enable LP printers
lpsched-	start the LP scheduler
lpshut-	stop the LP scheduler
lpmove-	move LP requests

For the commands MV/UX does not support, the following table lists the corresponding AOS/VS EXEC commands; see the How to Generate and Run AOS/VS(093-000243) Chapter 5 for detail.

Unix Command	AOS/VS EXEC Command
accept	CONTINUE
reject	PAUSE
disable	STOP
disable -c	FLUSH
enable	START
lpsched	START's on every active printer
lpshut	PAUSE's on every active printer
lpmove	No AOS/VS equivalent

Some Common Administrative Tasks and How to Perform them on MV/UX

* Adding a Printer

1) Use the AOS/VS EXEC commands CREATE, OPEN, START, and CONTINUE to set up the printer queue for the new printer.

   For the following operation, you must be logged in as user OP.

   Example: To establish queue LPT1, open it, associate it with device @LPB1, and to begin printing on device @LPB1, you would use the following commands:

```
) CONTROL @EXEC CREATE PRINT LPT1
) CONTROL @EXEC OPEN LPT1
) CONTROL @EXEC START LPT1 @LPB1
) CONTROL @EXEC CONTINUE @LPB1
```

2) Modify the lp configuration to allow use through MV/UX commands.

   Define a printer called LPT1, and associate it the device /per/lpb1. This affirms the association made by the EXEC START command above to the MV/UX lp system. For the following operation, you must have username LP.

```
$ lpadmin -pLPT1 -v/per/lpb1
```

* Changing the Default Destination for lp Printing

1) Use lpadmin -d[destination] to establish a default destination.

   For the following operation, you must be logged in as user LP.

   Example: To establish LPT1 as the default destination:

```
$ lpadmin -dLPT1
```

   Example: To establish no default destination:

```
$ lpadmin -d
```

For all the following operations, you must be logged in as user OP.

* Associate a Queue with a Printer
  Use the AOS/VS EXEC command START to associate a queue with a printer. This replaces enable(1).

* Disassociate a Queue from a Printer
  Use the AOS/VS EXEC command STOP to disassociate a queue from a printer. This replaces disable(1).
  Use the AOS/VS EXEC command FLUSH to replace disable -c.

* Allow a Device to Service Requests from a Queue
  Use the AOS/VS EXEC command CONTINUE to allow a device to service requests from a queue. This replaces accept(1M).

* Prevent a Device from Servicing Requests from a Queue
  Use the AOS/VS EXEC command PAUSE to prevent a device from servicing requests from a queue. This replaces reject(1M).

9.3 UUCP Configuration

Once the basic MV/UX system has been installed, UUCP can be configured and administered. The following documents should be read carefully:

* UUCP - UNIX to UNIX Copy" (contained in this release as /usr/man/docs/uucp_ref.ls). When using this document, note the following changes:
  /usr/lib/uucpsys is no longer needed or supplied on the release tape.
  The L.sys file now resides in /usr/lib/uucp.
  Ignore references to L.pubsys configuration file.
  Ignore references to uusys command.

* Support Tools Guide for the MV/UX and DG/UX Systems Chapter 10 "UNIX System to UNIX System Copy" (contained in this release as /usr/man/docs/uucp_tut.ls).

The specifications in uucp_ref.ls describe the modem/hardware and software requirements to use UUCP.

To configure UUCP the following steps must be taken:

1) If your machine is to have an autodialer you should set it up now. MV/UX supports 3 types of autodialers - Racal Vadic, Hayes Smartmodem 1200s and Bell. Racal Vadic and Hayes dialers have been tested. Bell dialers and direct connections have not yet been tested. Refer to uucp_ref.ls and your modem documentation to set up your autodialers. When specifying your modem lines in your AOS/VS sysgen, be sure to specify input buffer lengths of 254.

2) Create a profile for UUCP using the AOS/VS program PREDITOR. Its name must be UUCP. Its initial program file must be :USR:LIB:UUCP:UUCICO.PR and it must be allowed to use modems. Note that since the initial program is uucico.pr, the UUCP administrator cannot directly log in as UUCP. He must log in as some other username (such as OP or BIN) and then change his username to UUCP. A PREDITOR listing of the UUCP profile should look something like:

```
USERNAME: UUCP
INITIAL IPC FILE [:UDD:UUCP:INITIAL.IPC]
PROGRAM [:USR:LIB:UUCP:UUCICO.PR]
CREATE WITHOUT BLOCK [YES]
USE IPC [YES]
USE CONSOLE [YES]
USE BATCH [YES]
USE VIRTUAL CONSOLE [NO]
ACCESS LOCAL RESOURCES FROM REMOTE MACHINES [NO]
```

```
CHANGE PASSWORD [NO]
UNLIMITED SONS [YES]
CHANGE PRIORITY [YES]
CHANGE TYPE [YES]
CHANGE USERNAME [NO]
ACCESS DEVICES [NO]
SUPERUSER [NO]
SUPERPROCESS [NO]
MODEM [YES]
CHANGE ADDRESS SPACE TYPE [YES]
CHANGE WORKING SET LIMIT [NO]
PRIORITY [1]
MAX QPRIORITY [0]
DISK QUOTA [100000]
LOGICAL ADDRESS SPACE    - BATCH     [-1 SYSTEM DEFAULT]
LOGICAL ADDRESS SPACE    - NON-BATCH [-1 SYSTEM DEFAULT]
MINIMUM WORKING SET SIZE - BATCH     [-1 SYSTEM DEFAULT]
MAXIMUM WORKING SET SIZE - BATCH     [-1 SYSTEM DEFAULT]
MINIMUM WORKING SET SIZE - NON-BATCH [-1 SYSTEM DEFAULT]
MAXIMUM WORKING SET SIZE - NON-BATCH [-1 SYSTEM DEFAULT]
```

* Ensure that the correct entries for users initial working directories are made in /etc/passwd. UUCP uses these entries in template expansions.

* Ensure that the correct entries are made in /etc/utsname. UUCP gets the local hostname from this file. See uname(2) for more information.

* If configuration files do not already exist, create the UUCP configuration files. The files that must be configured are:

    * /usr/lib/uucp/L.cmds

    * /usr/lib/uucp/L.sys

    * /usr/lib/uucp/L?devices

    * /usr/lib/uucp/L?dialcodes

    * /usr/lib/uucp/USERFILE

  There are prototypes for these files in /usr/lib/uucp. Refer to uucp_ref.ls.

* For security reasons, ACLs are very important to UUCP. /usr/lib/uucp and /usr/spool/uucp and files in these directories should normally be permissioned so that only user uucp can modify them. /usr/spool/uucppublic, on the other hand, should be publicly readable and writable. The prototype files and

the directory structure have the correct permissions as released.

* Modify your system's UP.CLI macro. It should not enable the lines that are used by your autodialers. It may be simpler to change the UP.CLI macro to disable them after they are enabled (if, for example, all lines are enabled at one time). Regardless of the method used, the end result MUST be that the lines used by the autodialers are not enabled.

* This revision of UUCP is able to start necessary demon processes "on the fly", so that connections to remote machines, and local execution on behalf of remote entities is largely transparent. However, a cron job for uucp of the form

```
0 * * * * /usr/lib/uucp/uucico -r1
```

should be entered, so that aborted jobs are retried every hour. (You will probably want to change the zero in the example to some minute you choose "at random"... it is usually better if every uucico in the world doesn't try to get phone service at the same time...)

If you do not wish to run cron on your system, you can run uucico regularly using an AOS/VS EXEC style batch job. This can be submitted (by someone who logs on into the :CLI, like OP) like so:

```
*) defacl op oware uucp oware
*) create :usr:lib:uucp:uucp_janitor_output
*) create/i :usr:lib:uucp:uucp_janitor_input
*))directory :usr:lib:uucp
*))qsubmit/qoutput=uucp_janitor_output/after=+1 &
*))         uucp_janitor_input
*))process/default/block/ioc/user=uucp uucico -r1
*)))
*) qsubmit/qoutput=:usr:lib:uucp:uucp_janitor_output &
*)     :usr:lib:uucp:uucp_janitor_input
```

9.4 CRON Installation

Cron must be started from a process that will persist as long as cron is to run, and which has superuser, superprocess, and change-username privileges. A process with these properties is PID 2, process name OP:OP running from the operator's console. The following commands will start cron properly:

```
push
    super(user,process) on
    defacl op owre + wre
    process/def/input=@null/output=@null :etc:cron
pop
```

If desired, these commands can be added to the system UP.CLI macro.

9.5 Source Release Installation

Before installing the source release, please note the following:

* The AOS/VS C compiler must be set up for use with MV/UX in order to compile the MV/UX sources. See the Installation section of the C release notice for details.

* Source for the lorder command is not provided because it contains several PL/I library dependencies.

These instructions assume the MV/UX binary release has already been properly installed.

The following procedures are used to install the international source code release. The brackets denote CLI commands.

1) Become a superuser. [SUPERUSER ON]

2) Make your current working directory the LDU or directory that contains the USR directory from the MV/UX binary release. For example, if USR is in :UNIX:USR, [DIR :UNIX]. If it is in :USR, [DIR :].

3) Save any sources you don't want overwritten. The terminfo compiler sources (in USR:SRC:LIB:LIBCURSES:TERMINFO) which are part of the binary release will not be overwritten.

4) Mount the source code tape (release tape 3) on a drive capable of reading 1600 bpi magnetic tape and position the tape at BOT.

5) Load file 0 on the tape. [LOAD/BUFFERSIZE=8192/DEL @MTxn:0] where x is the drive type, n the drive number. This loads the MV/UX source subordinate directory structure, and most command sources.

6) THIS STEP DOES NOT APPLY TO INTERNATIONAL DISTRIBUTIONS.

   If you want the security administration sources, with the source code tape (release tape 3) on the drive, load file 1. [LOAD/BUFFERSIZE=8192/DEL @MTxn:1]

   This step loads relevant parts of USR:SRC:CMD, and USR:SRC:LIB, overwriting the international versions of these materials.

7) The file USR:SRC:TAPE3_LIST is a listing of all files on tape 3.

8) The source build requires LIB:INNER.LB. If your revision of LANG_RT has an INNER.LB, make LIB:INNER.LB be a link to it. Otherwise, make LIB:INNER.LB be a link to USR:SRC:LIB:INNER.LB.

9) The file USR:SRC:BUILD:_MK is a shell script which will create MV/UX from the sources. To use _MK, create a release area where the new commands will be installed. Edit _MK, assigning the release area directory name to the variable RELEASE. Unless you do this, _MK will OVERWRITE the commands in :BIN, :LIB, and :ETC. You must be SUPERUSER in order to run this script successfully. For example, to run the entire build in batch, from the shell (sh(1)), enter [qbatch su op _mk].

--End of MV/UX Release Notice--