Documentation for NULU.COM
as of 11/01/85
Version 1.5

ACKNOWLEDGEMENTS

This program would not have been possible were it not for the elegant work of Gary P. Novosielski and those others working with the LU library structure.

Additionally, I would like to thank the people who have supported the development of NULU in a variety of ways, from suggestions to financial contributions. All comments (good, bad and disastrous) are welcome. Special thanks to Keith W. Antcliff of Mead, Washington for his development of NULUTERM.ASM, a patcher overlay, and to Ron Fowler of Nightowl Software for his gracious permission to include MLOAD23.COM in the NULU release library.

Honors for Courage go to Dennis Recla, Norm Gregory, and Jim Switz for having the guts to "risk everything" by testing NULU and offering suggestions and pointing out bugs.

INTRODUCTION

NULU is a complete replacement for LU and LSWEEP. Weighing in at 15k, it includes all features of both programs and then some. It features:

Two operation modes with a full menu for each LU style CRC support and verification
Viewing and printing of member files and disk files, unsqueezing when necessary
Extraction of member files with an option to unsqueeze where appropriate
Bigger libraries, up to 800 members or more depending on available memory
Automatic member sorting, all members kept sorted at all times
Faster reorganization
Enhanced error handling
Swapped disk handling
Automatic self-verification
Access to deleted members
Access to read/only libraries
Wildcard library processing
Optional read/only patch
CP/M Plus support
General compatibility with LU and NSWEEP syntax
Numerous user patches

DIFFERENCES BETWEEN NULU 1.5 AND NULU 1.1

NULU 1.5 incorporates several significant changes, not least of which are the bug fixes. Version 1.1 had several bugs, including occasional improper extraction of large files, and clobbering of bytes in directory entry member names. All known bugs have been addressed as of the new release. (Actually, the extraction "bug" was due to one of the bugs in CP/M that effect unusual combinations of direct and sequential file access. If you watch carefully, you will see BDOS re-allocate used blocks of the disk, in descending order, no less!)

If you attempt to make changes to a library file that is logically read/only, NULU will display the message:

```
Library is READ-ONLY.  Modify anyway? (y/n)
```

and will wait for you to say yes or no. If you say no, then the operation you were attempting will be cancelled. If you say yes, you will be able to make changes and when you close the library, the file will be read/only once again.

There have been other, more visible changes. The disk interface has been expanded so that several commands that used to apply only to library member files, can now also work on disk files. These commands include, -D, -P, -V and a new command, -S, which performs a limited string search. For example, to delete a library member the command is still:

```
-d filespec
```

but you can also delete disk files with:

```
-d du:filespec
```

where the drive/user combination informs NULU that you are talking about disk files instead of library members. (Note that the -Z command has been re-assigned.) This syntax applies to the other commands listed above, as well.

Additionally, I have made an effort to bring NULU 1.5 further in line with LU syntax and operation, especially LU310.COM. Therefore, in addition to other changes, the user will find that the -M command for the NULU Command Menu has been changed to –H for Help.

NULU

To the best of my knowledge, NULU will run on any system running CP/M 2.x or higher, however it has only been tested by me on the Osborne 1 and the Osborne Executive computers. It is written in 8080 compatible, Z80 assembly language.

A complete discussion of the nature and advantages of library files is beyond the scope of my endurance, however a short recap is in order.

A library file is a single CP/M file which contains 1 or more smaller files as its members. Each file is written into the library and recorded in the library directory. After that, it may be accessed for viewing, printing, or execution, or may be extracted from the library to assume the status of a stand-alone file once again.

The reasons for doing this are many. For one, under CP/M each file occupies a minimum amount of disk space. This amount can range from 1k to 16k, depending on the system. Even though a file may only be 1 byte in length, it will still occupy the minimum required amount; the rest of the space is wasted. Putting these files in a library minimizes the possible waste to 127 bytes.

Each file also occupies a directory entry. On most floppy disk systems, the maximum number of file entries per disk is 64. By including a file in a library the directory space it used to occupy is freed for another file.

Additionally, using library files can simplify the process of categorization by subsuming several files of similar types or subjects into one file.

However, there are also dangers associated with using libraries. The main one is that if a library file is lost or damaged, all of the member files are lost as well. The solution to this problem can be summed up in one word: BACKUP. NULU probably isn't perfect, and we all know that disk systems aren't, so backup your important libraries, please. I don't want to hear any sad stories.

Almost as important as backing up libraries is the use of discretion when deciding what to put into them. Let's face it, if a file is in a library it is not as easy to get to. If you need to get to a file every 15 minutes, it probably shouldn't be in a library except as a backup measure.

AN ASIDE CONCERNING FILESPECS

When specifying a filespec, wildcards may be used freely, however, unlike most programs, NULU doesn't insist that a "." separate the filename from the filetype. For example, the filespec "**" means the same thing as "*.*" to NULU. On the other hand, the filespec "FRED.TXT" should be typed in just like that. "FREDTXT" would make NULU think that no filetype had been specified. The rule is simple: use a dot any time you like, but it is only strictly necessary when it is needed for clarity.

When specifying a drive and/or user area the syntax is very liberal. "A15:**" means the same as "15/A:*.*", as does "1A5:**". The colon is absolutely necessary. If a different user area is desired for the default drive, the drive specifier may be omitted, as in "15:**", just as the user area may be omitted as in "A:**".

If a drive/user specification is made, but no filespec is indicated, a filespec of ????????.??? is generated. Therefore to indicate the default drive, current user area, and a filespec of ????????.???, just type a colon by itself (e.g., ":")!

NULU OPERATION

When NULU loads, it automatically verifies its own CRC. This process causes a 2.5 second delay on my 4mhz machine. However, because of the safety that this feature affords, I hope that you will find it to be a reassuring pause. Note that the CRC is not affected by normal user patches.

NULU may be invoked with or without a command tail. A command tail can consist of any combination of valid NULU operators up to 128 characters (the limit imposed by the CCP. Once NULU is running, command lines can be as long as 254 characters.)

As in LU, all NULU commands are preceded by a dash. This is a copy of the menu for the command mode of operation:

```
-A Add members              -B Brief toggle
-C Close the library        -D Delete members/files
-E Extract members          -F Filesweep mode
-G Get filespec             -H Help
-K Krunch library           -L List members
-N Rename members           -O Open a library
-P Print members/files      -Q Unsqueeze members
-R Replace members          -S String search
-T Replace/Add members      -U Drive/User change
-V View members/files       -W Wildcard open
-X Exit NULU                -Y Disk directory
-< Redirect input           -> Redirect output
```

Commands may be strung together, each terminated by a space. They are processed left to right. All characters are converted to upper case.

The default drive and user area are displayed along with the name of the current command mode each time the console is prompted for input.

The commands will be explained in order of their appearance above. All items in [square brackets] indicate optional parameters. The word "filename" refers to an unambiguous file specification, while the term "filespec" refers to a potentially ambiguous file specification. Filespec parameters enclosed in (parenthesis) indicate references to deleted member files. Three dots ("...") after a parameter indicate infinite repetition.

During the execution of most commands which receive an ambiguous filespec as a parameter, entering a ^C at the console will interrupt the current process.

COMMANDS

-A ADD MEMBERS

Syntax: - a filespec[filespec...]

Use this command to add files from disk to the current library. NULU will make a series of passes through the directory, adding files as it goes, until the list of matching files is exhausted. If a matching filename already exists as an active member in the library, its name will be displayed and the file will not be added.

In all file addition and replacement operations, if a deleted entry of identical size can be located, that disk space and directory entry will be used rather than allocating new space for the file. This means that it will be necessary to reclaim wasted disk space less often.

An LU style CRC will be generated for each file added.

-B BRIEF TOGGLE

Syntax: -b

This toggles the prompting mode. The release version of NULU is setup to print the full name of the current mode for a prompt, as in "-Add members A0:>". If BRIEF is turned on the user will simply see "-A A0:>". NULU may be permanently patched to default to BRIEF ON or OFF. See NULU MODIFICATION.

-C CLOSE THE LIBRARY
Syntax: -c

This command closes the current library, writing its directory to disk if any changes have been made to the directory. The library directory is NEVER written except when the library is closed, so be sure to do it. If you forget to do so and remove the disk, NULU will prompt you for the disk again and will attempt to recover.

Some other operations that cause the current library to be closed are:

```
      -k, -o, -w, -x
```

-D DELETE MEMBERS/FILES
Syntax: -d filespec[filespec (filespec)...]
Syntax: -d du:filespec[du:filespec...]

If the drive/user specification is included, the DELETE function will apply to files on disk, otherwise it applies to library member files.

Member files matching the given filespec will be given deleted status in the library directory, except when the filespec is enclosed in parenthesis. In that case matching deleted members will be given active status. That is, they will be undeleted. If the filename of a matching deleted member file already exists as an active member, the filename will be displayed and the file will not be undeleted.

If the member file being deleted has a size of zero sectors, its entry will be entirely REMOVED from then library directory, thereby creating a new, free entry.

-E EXTRACT MEMBERS
Syntax: -e filespec[=newfilespec filespec du:filespec...]

Use this command to extract active member files. If extraction to the current drive/user area is desired, no further syntax is necessary. To indicate another drive, however, a destination drive/user area may be included in the source filespec. Example:

```
     -e a5:**
```

would extract all active members to drive A, user area 5. Files may be renamed as well as redirected by indicating a filespec along with, or in place of, a drive user specification following an equals sign. Examples:

```
      1.  -e *asm=*bak
      2.  -e *asm=5:*txt
      3.  -e fred.txt=sam.txt jane.inf=c8:girls.dbf
```

The examples would produce the following results:

1. Extract all files with a type of .ASM to the default drive/user renaming them with the filetype of .BAK.
2. Extract all files of type .ASM to user area 5 of the default drive, renaming each with the .TXT filetype.
3. Extract the member file FRED.TXT to the default drive/user under the name of SAM.TXT, and extract the member file

JANE.INF to user area 8 of drive C: with the filename GIRLS.DBF.

The two redirection rules to remember are:

1) if a destination filespec is entered it cannot be any less ambiguous than the source filespec, i.e., "*asm=*bak" is valid while "**=*bak" is not valid; and
2) any filespec following an equals sign takes precedence over any drive/user specification in the source filespec, i.e., "-e 6:*asm=*bak" would extract all member files to the current drive/user, renaming them, and would ignore the "6:" specification.

During extraction, a CRC is generated for each file. After the destination file has been closed, the CRC so generated is compared to the CRC for that library member that was stored in the library directory, if that value was non-zero. If the results are not identical, an error message will be displayed before the next member file or filespec will be processed. The erroneous destination file will not be deleted from disk by NULU as a result of this error.

-F

FILESWEEP MODE
Syntax: -f

This command places NULU in its second operational mode: the filesweep mode. This allows the user to move through the directory of active member files as if they were individual files being examined by a program like NSWEEP. The filesweep mode's command list is as follows:

```
A Next member             B Previous member
C Close library           D Delete member
E Extract member          F Find member
L Log new drive/user      M Mass operations
O Open new library        P Print member
Q Unsqueeze member        R Rename member
T Tag member              U Untag member
V View member             W Wildcard rename
X Exit NULU               Y Disk directory
Z NULU command mode       ? Help!
```

Because of the extreme similarity between these commands and the commands of NULU's command mode, only a short description of each command will be given.

```
A   --   Advance to next member
         (spacebar, cr, or lf will produce the same result)
B   --   Back up to previous member
C   --   Close the current library
D   --   Delete current member file
E   --   Extract current member file (prompt allows redirection)
F   --   Find first member matching input filespec
L   --   Change drive/user defaults (returns file pointer to the
         top of the file list and untags all member files)
M   --   Mass operations on all tagged files
         (allows  Deletion, Extraction, Printing, Unsqueezing, or
         Viewing)
O   --   Open new library (closes the current library)
```

P -- Dumps the current member file to LST: (unsqueezes if needed)

Q -- Extract current member file, unsqueezing if necessary (prompt allows redirection)

R -- Rename current member file

T -- Tag current member

U -- Untag current member

V -- View current member file (unsqueezes if needed)

W -- Wildcard rename (prompts for oldname & newname)

X -- Exit NULU (closes all files)

Y -- Get disk directory for default drive/user (returns file pointer to the top of the file list and untags all member files)

Z -- Return to NULU command mode (current library remains open)

? -- Print the filesweep mode menu

If, when the filesweep mode is entered, or after a library has been closed, there is not a library currently open, the filesweep mode will prompt with:

```
No library open.
```

and will accept only the following commands:

```
L,O,X,Y,Z,?
```

Likewise, if a library is open but only has a directory with no other active members, the message

```
No member files.
```

will be printed and only the commands listed above will be accepted.

During filesweep operation, each file will be listed in the order in which it is found in the directory, along with the size in K that the file would occupy if it were extracted to the default drive.

If the filesweep mode is terminated by a return to the NULU command mode, any commands that followed the -F command on the previous NULU command line will be executed.

-G GET FILESPEC
Syntax: -g filespec

NULU will search for the filespec indicated. If it is found, processing continues. If not, the user is prompted to insert the disk containing that filespec. The drive is then reset and search again. The program will prompt forever until it receives the proper filespec or until a ^C is entered, forcing NULU to continue without the filespec being found.

This command can be useful when attempting to control NULU through a submit utility like DRI's SUBMIT.COM. For example, one might type:

```
nulu -o a:asm -g b5:-work.005 -e b10: -g a0:-5.005 -x
```

After loading, NULU would open a library called ASM.LBR on drive A: in the current user area. Next it would search user area 5 of drive B: for filespec -WORK.005 until it was found. Then all active member files would be extracted to user area 10 of drive B:. Finally, NULU would search user area 0 of drive A: for a filespec called -5.005 until found. Then NULU would terminate. Notice here that the ASM.LBR didn't have to be closed before the search for the final filespec because no change had been made to the library directory. If a change had been made, after the new filespec had been loaded, NULU would have demanded the disk with ASM.LBR back so it could update the directory. Therefore the GET operation would be effectively negated.

-H HELP
Syntax: -h

Print the command mode menu.

-K KRUNCH LIBRARY
Syntax: -k[<number of entries to allow]

Even though all members are kept in alphabetical order at all times, when a member is deleted the disk space it occupies is not released to the operating system until the KRUNCH command is executed. KRUNCH consists of the following steps:

1. If a file of the same name as the library already exists in the default drive/user area AND THE LIBRARY FILE ORIGINATES FROM A DIFFERENT DRIVE/USER, a message indicating that the file exists will be displayed and the KRUNCH command will be skipped. You must delete the offending file or change drive/user area before the KRUNCH can be effected.
2. The library is closed.
3. A new library called WORK-LBR.$$$ is opened in the default drive/user area and the user is prompted to input the number of files that the new library should be able to contain. If the user wants just enough entries to contain the currently active members, the number 1 or any number less than or equal to the current number of active entries may be entered. If the optional parameter listed above is entered, it will be used as the number of entries. (The KRUNCH process may be aborted here by typing RETURN or 0. Please note that the library directory counts as an entry and will automatically be accounted for by NULU.)
4. Once WORK-LBR.$$$ is opened, the active member files from the original library will be copied one by one into the new library. The copy routine uses a recursive process to copy as many files at one time as possible. A CRC is calculated for each member as it is copied. If this number does not match the CRC recorded in the original library directory, the message:

   ```
   CRC ERROR-Continue? (y/n)
   ```

   will appear after the filename in question. A reply of "N" will result in the work library being deleted and the original library being re-opened. Otherwise the newly generated CRC for the file will be stored in the directory of the new library.
5. The old library is deleted and WORK-LBR.$$$ is renamed with the old library name.

Examples:

1. -k
2. -k <1
3. -u b5: -k <63
4. -u b5: -k <1 -u a0: -k <63

The above examples would produce the following results:

1. The library would be KRUNCHed to the default drive/user area. The user would be prompted to input the number of entries to allow.
2. The library would be KRUNCHed to the default drive/user area with just enough entries to contain all currently active member files.
3. The default drive/user area would be changed to B5: before the KRUNCH begins. Then the library would be KRUNCHed, allowing 63 entries.
4. The default drive/user area would be changed to B5: before the KRUNCH begins. The KRUNCH would commence, allowing only enough entries to contain all active members. The drive/user area would be changed to A0:. Then the library would be KRUNCHed again, allowing 63 entries.

-L LIST MEMBERS

Syntax: -l[filespec]
Syntax: -l[(filespec)]

Use this command to list the contents of the library directory. Each member filename will printed followed by the starting relative sector number in the library file, the size of the member in sectors, the size in K that the file would occupy if it were to be extracted to the default drive/user area, and the CRC for the file. Finally, a recap of the size of all member files listed is printed, along with the number of sectors occupied by deleted files.

The parameters listed above can be used to control the selection of files for display. Examples:

1. -l
2. -l **
3. -l (**)
4. -l *asm
5. -l (fred.txt)

The above examples would produce the following results:

1. All active member files listed.
2. All active member files listed.
3. All deleted member files listed.
4. All active member files matching the filespec of ????????.ASM listed.

5. All deleted member files called FRED.TXT listed. (Note that it IS possible to have more than one deleted file with the same name.)

-N RENAME MEMBERS

Syntax: -n oldfilespec=newfilespec[...]
Syntax: -n (oldfilespec)=newfilespec[...]

This command is used to rename active or deleted member files. Wildcards are fully supported, but the newfilespec can be no less ambiguous than the oldfilespec. Deleted members may be renamed by enclosing the oldfilespec ONLY in parenthesis. Examples:

1. -n *asm=*bak
2. -n (fred.txt)=sam.txt
3. -n **=**

The above examples would produce the following results:

1. All active members matching the filespec ????????.ASM would be renamed with a filetype of .BAK.
2. The first deleted member file called FRED.TXT would be renamed to SAM.TXT. All other members matching that filespec would be listed along with a message indicating that they could not be renamed.
3. All active member files would be listed, each with a message indicating that they could not be renamed because the name already exists.

-O OPEN A LIBRARY

Syntax: -o filename[<number of entries to allow]

With the exception of the filesweep Open Library command, this is the only method to open or create a library. NULU will search for the filename indicated. If it is found, it will be opened and a message will be displayed indicating the size of the Data Transfer Buffer. This is the number of sectors in memory that NULU will have to use in extracting, adding or copying member files.

If the file is not found, the user will be prompted for the number of entries to allow in the new library. If the optional parameter above is passed, the number will be used to determine the directory size of the library. All directories can contain some multiple of 4 entries. All numbers input will be rounded up to the nearest multiple of 4. Remember that the directory itself counts as one entry and that NULU will automatically make space for it.

The filename passed must be unambiguous.

NOTE ON DIRECTORY SIZE:

Depending on available memory, libraries with as many as 800 entries (theoretically) can be opened. But watch the Data Buffer Size! If it is less than 9 sectors you will not be able to unsqueeze any files. With a user TPA of 58k, I have been able to open libraries with no more than 584 total members and still be left with a buffer of 9 sectors. (See -< command and how it affects library memory.)

-P PRINT MEMBERS/FILES
Syntax: -p filespec[filespec (filespec)...]
Syntax: -p du:filespec[du:filespec...]

If the drive/user specification is included, the PRINT function will apply to files on disk, otherwise it applies to library member files.

With this command, the ascii contents of files matching the filespecs given will be dumped to the list device. If the file is squeezed, NULU will unsqueeze it. Deleted library members may be listed by enclosing the appropriate filespec in parenthesis.

-Q UNSQUEEZE MEMBERS
Syntax: -q filespec[=newfilespec filespec...]

This command is identical in operation and syntax to the EXTRACT command, except that if the matching members are squeezed, they will be unsqueezed.

-R REPLACE MEMBERS
Syntax: -r filespec[filespec...]

This command is identical in operation and syntax to the ADD command, except that matching files will be added to the library ONLY if they already exist in the library. They will be deleted, then the new files will be added.

-S STRING SEARCH
Syntax: -s filespec "string to search for"
Syntax: -s du:filespec "string to search for"

If the drive/user specification is included, the SEARCH function will apply to files on disk, otherwise it applies to library member files.

This provides a way to do a SIMPLE search for a given string among a group of files. If the "string to search for" is found you will be notified. All files searched are listed. The individual line on which the item was matched is not displayed because the match is not made on a line basis. That is, you may search .COM files or squeezed .OVR files just as easily as searching text files. However, the implementation is admittedly rather weak at this point as things like control characters and tabs are ripped out of the command line before the search function even sees the command. Enjoy it and I'll improve it later.

-T REPLACE/ADD MEMBERS
Syntax: -t

This command is a combination of the ADD and the REPLACE commands. If the files do not exist in the library, they will be added. If they do exist, they will be replaced.

-U DRIVE/USER CHANGE
Syntax: -u new drive/user

Use this command to change the default drive/user area. A colon (":") must follow the drive/user spec. Example:

```
    -u a5:
```

would switch the default drive/user area to A5:.

-V VIEW MEMBERS/FILES
Syntax: -v filespec[filespec (filespec)...]
Syntax: -v du:filespec[du:filespec...]

If the drive/user specification is included, the VIEW function will apply to files on disk, otherwise it applies to library member files.

This command will list the ascii contents of all files matching the given filespec to the console device. Deleted library members may be indicated by enclosing the filespec in parenthesis. If the files are squeezed, they will be unsqueezed. At each page, NULU will accept one of the following commands:

1. ^X to skip to the next matching file
2. ^C to abort further listing
3. L to list one line
4. CR, space, or LF to list another full page

At the end of each file the message

```
    "Press RETURN"
```

will be displayed and NULU will wait for a keystroke before moving to the next matching file.

-W WILDCARD OPEN
Syntax: -w filespec commands to process

<u>For advanced users only.</u>

This can be one of the most powerful, and possibly the most dangerous, of all NULU commands. It allows the user to specify a group of libraries on which to let a set of commands operate. A simple example:

```
    -w a5:*lbr -l
```

would open all files found on drive A in user area 5 as libraries and list the contents of each. This would happen in sequence for all matching files in the order in which they were found in the directory. Note that this command has the same effect as the -O command except that it processes wildcard library filespecs and that the user's control over each library opened is limited to the single line that invokes the -W command. That is, all commands applying to each library opened must be contained on the one command line.

The advent of this command is the reason that certain commands that used to be restricted to NCF redirection files are now available at the command level. Take this example:

```
    -w *lbr -k <5 -p ** -l -' "hit enter for next file " -:
```

That command would result in the following for each matching file:

1) The library would be KRUNCHed to 5 members.
2) All member files would be sent to LST:.
3) The library directory would be listed.
4) The prompt, "PRESS ENTER FOR NEXT FILE " would be displayed to CON:
5) NULU would wait for the user to press RETURN before passing control back to -W which would then do the same thing for the next library.

Use a lot of caution when using this command. It is new even to me, but I have found it to be EXTREMELY useful. I really like this one:

```
-w *lbr -f
```

which allows me to look at a lot of libraries through filesweep without having to know their names.

I have tried something like this:

```
-w *lbr -< command.ncf
```

but with very disappointing results. It doesn't work. I will work on that for next time, if there is a next time.

-X

EXIT NULU
Syntax: -x

This command will set a flag indicating that when the current NULU command line is exhausted, NULU should terminate. But the command is actually a toggle. If it is entered twice on a line the effect of the second issuance will be to negate the first. Note that the support for command chaining has been removed. Examples:

1. -x
2. -x -x

The above examples would produce the following results:

1. The termination flag will be set and NULU will terminate when the current command line is exhausted.
2. The termination flag is set and then reset. NULU does not terminate.

-Y

DISK DIRECTORY
Syntax: -y[filespec]

This command will print an unsorted directory of the disk indicated by the parameter passed, or a complete directory if none is passed. The scope of the directory may be limited by indicating the appropriate ambiguous filespec. If the dim or reverse video strings have been defined, NULU will use that characteristic to indicate that certain file attribute bits are set in each filename.

-Z ZAP DISK FILES
Syntax: -z filespec

Use this command to delete disk files while still running NULU. Each file matching the ambiguous filespec given will be listed and deleted. Libraries can be deleted just as easily as any other type of file, so be careful. The only protection afforded is that the currently open library, if any, will never be deleted.

-< REDIRECT INPUT
Syntax: -< filename

This command will cause NULU to open the filename indicated and begin accepting commands from it instead of receiving them from the console. This type of file is, in effect, a NULU Command File. The filetype will default to ".NCF" if no filetype is specified. Syntax of the commands in this file is compatible in every way to the syntax used in normal NULU operation. Additional functions have been added to facilitate the development of NCF files. Each command line in the file must be terminated by a CR,LF, and will be converted to upper case and echoed to the console before processing. The last line in the file should end with at least two CR,LF combinations. After all NCF commands have been processed, control will be returned to the console. Any commands appearing after a "-<" command will be ignored.

If another "-<" command is encountered in the file, the current command file will be suspended and the new file opened. When the commands in that new file have all been processed, control will "return" to the file that "called" the new file. This nesting can continue to the level allowed by the number of input files patch described in NULU MODIFICATION. (It can actually continue to a much deeper level if no library is open, but what's the point? Hmmm..., maybe a tutorial...)

The filename passed must be unambiguous.

Of the following commands, these are valid only when received from an NCF file: -j, -z, -". The others may be used at any time but are included here and are omitted from the menu because of their programmatic nature.

-------------------------- **Programmatic Commands** --------------------------

-J JUMP TO LINE
Syntax: -j line number

Use this command to have NULU start reading command lines from the current NCF file at the line number indicated. The example:

```
    -j 1
```

would cause NULU to start reading lines from the current NCF file beginning at line number 1. Forward and backward references are allowed. A reference to line number 0 will be ignored. Any references to non-existent line numbers will cause the current NCF file to terminate, thereby returning to the next higher NCF level, or to the command mode if this is the highest NCF level.

Note that if you add or delete any lines in an NCF file, that will completely foul up any jump commands in that file. That's a bug, not a feature.

-Z

TOGGLE CONSOLE
Syntax: -z

This command toggles all output to the console device. That is, if this command is issued once, all console output is shut down, with the exception of the special commands listed below. Issue it again, and console output will be re-enabled.

-'

PRINT TEXT
Syntax: -' "text to print"

Use this command to print text to the console, no matter what the condition of the console output toggle. Text to be printed must be enclosed in quotes. The opening quote MUST be present, but the closing quote is only necessary to delimit any following commands.

-"

PROMPT AND CHAIN TO PREVIOUS MODE
Syntax: -mode to chain to -" "prompt" -failure mode

My method of syntax diagramming falls utterly short of describing this command, I know. Let's take it one step at a time. This command allows the NULU programmer to specify the mode to which he wishes to pass a parameter, prompt the user for that parameter, and to also specify actions to be taken if no response is received from the user. Now for an example...

```
-y *lbr -o -" "open which library? " -j 65535
```

The following result would be produced:

1. A directory listing of all files with a type of .LBR would be listed to the console device if the console were toggled on.
2. The OPEN LIBRARY mode would be transparently entered just long enough for NULU to take note of it.
3. The console would be prompted with the message:

   ```
   (CR, LF)
   OPEN WHICH LIBRARY?
   ```

   and NULU would wait for the user to enter a line of text NOT EXCEEDING the length of the prompt(!). If the user entered a RETURN by itself or a null line, the prompt would reappear on the next line. If the user entered a valid line, the first word on that line (delimited by a space, tab, etc.) would be passed to the OPEN LIBRARY mode and the balance of the line would be ignored. If, on the other hand, the user entered an ESCAPE followed by a RETURN (1bh, 0dh) the commands after the prompt would be executed and control would be passed to line number 65535, almost surely a non-existent line, thereby terminating the current NCF file.

   Some commands which normally accept parameters cannot receive parameters from this mode because they are decoded differently. These include:

   ```
   -:, -;, -l, -n, -x, -y, ->
   ```

-: WAIT FOR RETURN
Syntax: -: -failure mode

This command will force NULU to wait for the RETURN (0dh) character to be input to the console device. If a RETURN is received, the balance of the command line will be ignored. If a ^C is entered, the remaining commands on the line will be executed.

-; COMMENT
Syntax: -; comment about this kludge of a JCL

This command will cause NULU to ignore all text appearing after the command on the same physical command line.

------------------------ **End Programmatic Commands** ------------------------

-> REDIRECT OUTPUT
Syntax: -> filename
Syntax: ->

With the first form of this command, NULU output will be sent to the filename indicated. The default filetype of ".NOF" will be used if no filetype is specified. If the file already exists, it will be deleted. All special characters that the user may have patched into NULU (see below, NULU MODIFICATION) will be sent to the file as well, with the exception of the EOF character, 26 (1ah). The only output not echoed to the file will be output caused by viewing or printing a member file. Even if the console has been turned off by an NCF file, all console output will continue to be sent to this file.

The filename passed must be unambiguous.

Under the second form of this command, the current output file, if any will be closed. When NULU is caused to terminate, the current output file is closed along with the current library, if any.

ERROR HANDLING

NULU should never crash to CP/M anymore. When NULU signs on, it redirects the warm boot BIOS vector, thereby preventing NULU termination due to BDOS errors. Under CP/M Plus, it goes even further, shutting down the console error messages, handling them internally. Therefore, if NULU attempts to perform some operation on a drive where no disk is mounted, one could reasonably expect (on a 2.2 system) for the BIOS to issue some error message. The user then hits RETURN or ^C and NULU should recover. It may think that the disk was full, when there really wasn't one there at all, but it should still take the proper course of action needed to recover.

ERROR MESSAGES

Most NULU error messages are of the form:

```
ERROR XX: Explanation
```

where "XX" is some number from 0 to 255 and "Explanation" is a clue as to the nature of the error. Each error message is listed below, along with an explanation. Suggestions about ways to handle the error are given where appropriate.

ERROR 1: reading unwritten data
Problem: A sector required to gain access to a file is indicated by CP/M to be unwritten.
Solution: The library directory or perhaps the library itself has become trashed at some point. Deleting the entry and KRUNCHing will probably render it harmless, but the safest course to take is to obtain a fresh copy of the library.

ERROR 2: disk full
Problem: The library disk became full during file addition or KRUNCHing or the destination disk of an extraction command became full.
Solution: If the error occurred during KRUNCHing, NULU should have recovered by itself, reopening the old library. If it did not recover, then you removed the original disk before the KRUNCH was complete or a serious read error occurred.

If the error occurs during file addition to the library, operation should not be impaired, but no files that required additional disk space can be added to the library until some disk space is cleared. Note that deleted entries can still be overwritten by incoming files as long as the file sizes match. The incoming file that caused the error will be recorded in the library directory as a deleted entry. If the error occurs as a result of attempting to replace a member, the original library member should be unharmed and left as an active member.

If the error occurs during file extraction or unsqueezing the destination file is deleted because it is an incomplete file. Extraction can continue as before.

ERROR 4: reading unwritten data
See ERROR 1. This is the same except that BDOS mentioned that an entire extent was missing.

ERROR 5: can't make file
Problem: An attempt was made to create a file on a disk with no free directory entries.

ERROR 10: media changed
Problem: You pulled the diskette out in the middle of a crucial operation.
Solution: Don't do that. It is probably an unrecoverable error.

ERROR 11: disk i/o error
Problem: You are probably running under CP/M Plus if you get this message. It means that the operating system detected a physical problem with your disk and the read or write failed.
Solution: Try it again. I have recovered from this type of error by persistence.

ERROR 12: disk r/o
Problem: Your operating system probably discovered that you have a write-protect tab on your diskette.
Solution: Remove it.

ERROR 14: bad drive
Problem: An invalid drive for your system was chosen (see below, NULU MODIFICATION) or a letter higher than "P" was indicated.
Solution: Choose another drive or re-patch NULU.

ERROR 17: password error
Problem: You tried to access a file for which a password was required. NULU does not support access to such files.

ERROR 18: file exists
Problem: You attempted to KRUNCH a library from one user area into a different one in which a file with the same name already resided.

ERROR 19: ambiguity error
Problem: This means that you entered an ambiguous filespec where an unambiguous filename was required, or that the destination filespec in an extraction or unsqueeze command was less ambiguous than the source filespec. See -E EXTRACT MEMBERS for complete details.

ERROR 70: file table full
Problem: You attempted to open one more NCF file than your nesting patch supported. See NULU MODIFICATION.

ERROR 74: checksum error
Problem: During the unsqueezing of a file, the internal checksum generated did not match the explicit checksum stated in the file's squeeze decoding table.
Solution: Get a new copy of the member file. Yours has been trashed somehow.

ERROR 75: CRC error
Problem: When each member file is added to a library, a CRC is created for it and stored in the library directory. If, upon extraction or unsqueezing, the CRC generated at THAT time does not match the original CRC in the directory, that means that the sectors wherein that library entry resides have become corrupted. There is an outside chance that the CRC entry in the library directory has become damaged instead, however that condition would probably also generate ERROR 107.

ERROR 77: not enough memory
Problem: An attempt was made to open a library with a directory too large to be accounted for with available memory.
Solution: Open the library on a computer with a larger TPA and reorganize it so it can be handled by the smaller system.

ERROR 78: user cancel
Problem: No problem. NULU is simply letting the user know why an operation has been cancelled. Most operations which work on ambiguous filespecs may be interrupted by entering a ^C during their operation. (You may have to hold the key down a while.) The KRUNCH operation may be so interrupted as well.

ERROR 83: bad syntax
Problem: Improper syntax was used to attempt some operation.

ERROR 85: file not found
Problem: The filespec indicated for some operation could not be located.

ERROR 86: incompatible cp/m version
Problem: An attempt was made to run NULU on a system with a CP/M version number of less than 2.0.

ERROR 100: bad library directory
Problem: The file specified could not be opened as a library file.
Solution: The directory could be bad, but most likely the file simply isn't a library. Check it out with some disk editor like SPZ or EDFILE.

ERROR 107: library directory CRC error
Problem: The CRC that was calculated and stored for the directory the last time the library was modified does not match the directory CRC calculated as the library is opened this time.
Solution: With most libraries, this error will be valid. In that case, the only remedy is to obtain a new copy of the library. However, there are some libraries that were created with LU, and then modified but never KRUNCHED by NULU 1.0. In that case, it is likely that the old, original CRC stored in the library directory would still be there, even though it is way out of date. My suggestion is, when you first get NULU 1.5, go through all of your .LBR files, if you were using NULU 1.0 previously, and KRUNCH them. This way you will know later on that the CRCs stored in each library should all be accurate.

ERROR 116: squeeze decode table error
Problem: A file has a squeeze decoding table, but at some point the table is trashed or perhaps is missing.
Solution: None really; get a new copy of the file. If your Data Buffer Size has fallen below 9 sectors, then the reason might be that the entire decode table could not be read into memory.

ERROR XX: undefined
Problem: Unknown.
Solution: Write down all information presented on the terminal. The number following the word ERROR is especially important. Get the information to me. Mail it or call and I will try to help.

The most serious, non-recoverable error is:

NULU CRC error

which can occur when NULU 1.5 is first loaded. This means that NULU failed to match its own CRC. You need to get a fresh copy. Or perhaps you have tried some patch not described below...?

NULU MODIFICATION

Patches are listed in the order in which they appear in NULU.COM. Each patch is explained and the default values are listed immediately next to the patch address. All addresses are absolute hexadecimal.

Patch	Default	Explanation
00163h	50h	Number of characters each full video line can contain. Not necessarily the same as the number you can see.
00164h	50h	Number of characters visible on a line at any one time.
00167h	07h	This byte indicates the number of files that NULU can have open at any one time. The base number is 4 and should never fall below that. The value of 3 has been added to it to allow 3 level nesting of NCF files. If the NCF value is increased or decreased, identical changes should be made to this byte. (See Patch at 00220h.)
00168h	0010h	This obscure word controls the number of relocatable tasks that can be controlled by the NULU memory management routines. Because each file is a relocatable task, the same modifications for the previous byte apply here. The base value of this word is 000dh and should never fall below that point. (See Patch at 00220h.)
0016ah	10h	Number of contiguously numbered disk drives in your system. If you have drives A and B, set this value to 2. It will prevent you from accessing an invalid drive.
0016bh	10h	Number of contiguous user areas that NULU will be allowed to access. Under normal CP/M 2.2, Turbo-Dos or a ZCPR system, I understand that this number can be as high as 20h. For CP/M Plus or most systems that use time/date stamping this number should not exceed 10h.
0019ch	0005h	This is the address of the BDOS vector. All operating system calls go through this address. If you have some weird, non-standard CP/M implementation you can patch this to try to make NULU work with it.
0021ch	00h	Set to a non-zero value to make NULU READ/ONLY. No operators will be executed that could change the library or disk directories in any way.
0021dh	00h	LU.COM was set up to exit automatically if a command tail was used when LU was invoked. With the release version of NULU, it takes the -X command to exit. Set this byte to a 01h to get it to act like LU in this respect. That is, it will automatically exit after executing the passed command tail, if one was present.

0021eh	01h	Set to a 00h to start NULU with BRIEF mode ON (long messages not printed).
0021fh	01h	Set to zero to prevent users from being able to ESC,RET from the PROMPT AND CHAIN TO PREVIOUS MODE prompts in NCF files. (Be sure to also change the values at 00167h and 00168h.)
00220h	03h	This byte indicates the nesting level allowed for NCF files. I have no idea what the upper limit is, but watch that Data Buffer Size!
00221h	'-Y *LBR -O',0	Starting here are 39 bytes in which any valid NULU operators may be stored. This line will be automatically executed when NULU is loaded, even before any other parameters passed on the command line. This might be used to create a version of NULU that, after loading, sets the drive/user defaults to B0:, prints the menu, scans the drive for library files, then switches to the open library mode. That command line would look like this: -U B: -H -Y *LBR -O Notice that all characters in thisline MUST be UPPER CASE. The entire line MUST be terminated by a null byte. Only 39 bytes are available for text. not a byte more. The 40th byte is for the null terminator.
0026fh	3ch	This is the number of lines that NULU will print on each page during member file printing. Change this value in accordance with the line printer patch below. Set it to zero to have NULU ignore line counts altogether.

Starting at 0024ch are 5 data areas used to define strings needed to control the console attributes. Each string is preceded by an identification byte and a byte indicating its active length. Then 5 bytes are available for the actual definition. Therefore a total of 7 bytes is consumed for each video control. The attributes are, in order:

```
      DIM or REVERSE VIDEO ON      (DIM is preferred)
      DIM or REVERSE VIDEO OFF
      UNDERLINING ON
      UNDERLINING OFF
      CLEAR SCREEN
```

Each control becomes active as soon as it is patched in.

Likewise, there is one string defined for the LST: device thatwill be issued immediately before printing a file. It begins at 00271h and has the same format as the strings above.

FINAL NOTES

Please report any bugs or problems to Martin Murray, P.O. Box 35972, Dallas, Tx 75235 or call (214) 351-6117, EVENINGS AND WEEKENDS ONLY. Please specify the version of NULU with which you are working.

As always, donations will be gratefully accepted. It took a lot of work to produce NULU and if you enjoy it and it benefits you, don't hesitate to show your appreciation with a check! Note: I no longer work for SYSTEM SOLUTIONS, so please, no checks made out to them. I am now working for INOVA, a company founded specifically to support 8-bit computers running CP/M. Look for new and exciting retail products from that source. (The preceding has been an official plug.)