

Documentation on ZCPR - A Z80 Replacement for the CP/M CCP

ZCPR Documentation By RLC

Table of Contents

Introduction	2
Part A: Installation Instructions	4
ZCPR Integration Example	5
Setting the ZCPR Inline Options	8
REL, BASE, CPRLOC, RAS, SUBA, CLEVEL3	8
Customization Symbols	8
N_LINES, WIDE, PGDFLT	8
PGDFLG, MAXUSR, SYSFLG, SOFLG, SUPRES,	
DEFUSR, SPRMPT, CPRMPT, NUMBASE,	9
SECTFLG, FENCE	10
Patching SUBMIT.COM	10
Part B: Usage Instructions and Explanation of Commands	11
The ZCPR Command Hierarchy Search	11
The ZCPR-Resident Commands	14
DIR, ERA	14
LIST, TYPE, SAVE	15
REN, USER, DFU	16
JUMP, GO, GET	17
ZCPR Error Messages	18
Part C: ZCPR Command Levels and How to Use Them	19

Documentation on ZCPR - A Z80 Replacement for the CP/M CCP

ZCPR is a replacement for the CP/M Console Command Processor (CCP) which is designed to run as part of CP/M on Z80-based microcomputers. In most cases it is upward-compatible with the original CP/M Version 2.2 CCP.

ZCPR, however, provides many extensions to the CP/M CCP. Included in these extensions are the following features:

- The TYPE function can be made to page or not page its output at the user's discretion
- A LIST function is available which sends its output to the CP/M LST: Device and does NOT page
- The DIR command has been extended to allow the display of the system files or all files
- The ERA command now prints out the names of the files it is erasing
- The current user number may be included as part of the command prompt; if the user is under a number other than 0, the prompt is of the form 'du>' (like 'A2>' or 'B10>'), and, if the user is under 0, the prompt may be 'd>' or 'd0>' as per his choice
- The SUBMIT facility has been changed in two basic ways:
 - the prompt changes to 'du\$' or 'd\$' when the SUBMIT command is printed
 - the \$\$.SUB is executed from drive A: (note that the original SUBMIT problem now exists, but the new SUB.COM facility corrects it); the CCP-GROUP definition of an Indirect Command File now applies, and this definition is that any sequence of commands which may be issued from the console is also a valid sequence of commands for execution from an Indirect Command File; hence, the sequence:

```
DIR
B:
DIR
A:
```

may be issued from either the console or an Indirect Command File, and the results of the execution of this sequence are the same. Basically, this says that Indirect Command Files are upward-compatible to the console input (but not necessarily that the contents of an Indirect Command File may be issued at the console without modification).

- A command-search hierarchy is now implemented which is executed roughly as follows:
 - the user's command is checked against the CPR-resident commands and executed immediately if a match is found
 - failing that, the current user number on the current disk is scanned for the COM file; the COM file is loaded and executed if found
 - failing that, a default user number (initially 0 but can be reset with the DFU CPR-resident command) on the current disk is scanned for the COM file; the COM file is loaded and executed if found
 - finally, failing that, the default user number on disk A: is scanned for the COM file; the COM file is loaded and executed if found or an error message (COMMAND?, when COMMAND was the user's command name) is printed
- The numeric argument for the SAVE command can be specified in hexadecimal so that the user may employ the values presented by tools such as DDT exactly as they are given
- A GET command which loads a file at a specified memory address and a JUMP command which "calls" the subroutine at a specified memory address have been added; a GO command which "calls" the subroutine at 100H (subset of the JUMP capability) has also been added

This document provides the user of ZCPR with the following information:

Part A: Installation Instructions
Part B: Usage Instructions and Explanation of Commands
Part C: ZCPR Command Levels and How to Use Them

Part A
Installation Instructions

In order to install ZCPR on a target microcomputer (must be currently running CP/M 2.2), the user must know two basic things:

- 1) Where his CCP is currently running in memory
- 2) Where his CCP is located in the SYSGEN image, or, for systems which don't support SYSGEN (such as P&T CP/M 2.2 for the TRS-80 Model II), where his CCP is located on disk and how to place the new ZCPR on top of it

The first question is answered relatively easily. A program, known as either BDOSLOC or BDLOC (for BDOS Locator), is provided with ZCPR. You should assemble this program for your particular computer (change the base ORG if you are running non-ORG-0 CP/M) and execute it. Upon execution, it will provide you with the base address of (1) the BDOS and (2) the CCP for your particular system. BDOSLOC has worked correctly for all systems tested so far, but there is always a chance that it may NOT work for some non-tested system. For the time being, assume that it works correctly and record the starting base page address of your CCP.

The second question is not answered nearly so easily. If you have the ability to SYSGEN your system, it is much easier (commonly) than if you do not. You must, after assembling the ZCPR properly, integrate it into the sysgen (or disk) image of CP/M. This can be done by obtaining a SYSGEN image of your system, scanning it via a debugger such as DDT to find the offset for the CCP, reading the new CPR in on top of the old one, and finally running SYSGEN again to place the resultant system on disk. If you DO NOT have SYSGEN capability, a Disk Utility program is required to locate the CCP on disk and then write the new ZCPR on top of the old one. The net result of this integration is the placement of the new ZCPR onto disk in the proper place so that it will be loaded with the rest of CP/M on cold boot and executed properly.

To find the original CCP, you typically have to locate it by its appearance. It is probably stored contiguously on disk, so, once it is found, a sequential overwrite is all that is required. Probability is extremely high that it is stored contiguously in the SYSGEN image. The CCP starts with two (2) and ONLY TWO jump instructions followed by a buffer area (possibly containing an initial command and/or the Digital Research copyright notice). The Digital Research manuals show the CCP to reside at address 980H in the SYSGEN image, but this may vary with system. To find this image, use DDT or some other such debugger, load the SYSGEN image you can get via SYSGEN, and examine memory starting at around 900H for the two (and ONLY two) jumps described above. If you find an area with more than two jumps (a group of them), you are probably looking at the BIOS and should go lower for the CCP. The CCP will probably start on an even page or half-page address (like 900H, 980H, 1100H, etc).

Now that the location of the CCP has been found, record this address for later. You are now ready for the integration of ZCPR into your system. To do this, perform the following steps using the information of the page address of the CCP (obtained from BDOSLOC and called CPRLOC within ZCPR) and the SYSGEN image address of the CCP (called IMAGE for reference in this document).

1. Edit ZCPR and set the CPRLOC equate to the value obtained from above. Also set any flags and values as you desire (see the section on ZCPR Customization below). When satisfied, end the edit session.

2. Assemble ZCPR with MAC (or equivalent). This assembler is required because of the MACROS used. Only the resultant HEX file is required.

3. Assuming that you can use SYSGEN, obtain a SYSGEN image of your current CP/M system and save it on disk.

4. Load the SYSGEN image into memory with DDT (or equivalent). Once loaded, verify that the original CCP is at the IMAGE address found above and compute the integration offset using the DDT H command:

H<IMAGE adr>,<CPRLOC adr>

The second number displayed gives you the OFFSET value required for step 5.

5. Integrate ZCPR into your SYSGEN image via DDT's I and ROFFSET commands. Use IZCPR.HEX (or the name of your version of ZCPR) to load the FCB and ROFFSET (where OFFSET was computed in step 4) to load the ZCPR.HEX file into memory at the proper location. Check to see that ZCPR is indeed properly loaded by examining the SYSGEN IMAGE area.

6. Place the new system on disk by running SYSGEN and NOT loading the system from disk (use the memory image).

For further clarification of the above process, the following is a sample terminal session which outlines the steps taken.

ZCPR Integration Example

```
B>; Sample terminal session for integrating ZCPR
B>sysgen
SYSGEN VER 2.2
SOURCE DRIVE NAME (OR RETURN TO SKIP)b
SOURCE ON B, THEN TYPE RETURN <-- I hit the RETURN key here
FUNCTION COMPLETE
/ DESTINATION DRIVE NAME (OR RETURN TO REBOOT) <-- and here
B>save 44 cpm56.com <-- We now have a SYSGEN image of CP/M
to work with
```



```
B>ddt cpm56.com           <-- Now to integrate!
DDT VERS 2.0
NEXT PC
2D00 0100
-h1100,bd00           <-- Compute offset for new CPR
CE00 5400           <-- Offset is 5400H
-icpr.hex           <-- Init FCB
-r5400           <-- Read in new CPR with offset
NEXT PC
2D00 0000
-^C           <-- Done!
B>sysgen           <-- Now to SYSGEN onto disk
SYSGEN VER 2.2
SOURCE DRIVE NAME (OR RETURN TO SKIP) <-- Use memory image
DESTINATION DRIVE NAME (OR RETURN TO REBOOT)b <-- onto B:
DESTINATION ON B, THEN TYPE RETURN
FUNCTION COMPLETE
DESTINATION DRIVE NAME (OR RETURN TO REBOOT) <-- Done for now
```

B>

Setting the ZCPR Inline Options

The following are the four basic options available to the user under ZCPR for customization of his package.

Option Name	Function
REL	Configures CPRLOC (CPRLOC equ 0) for integration via MOVCPM rather than the DDT/SYSGEN technique outlined above; set to TRUE for MOVCPM integration or FALSE for DDT/SYSGEN integration
BASE	Base address of your CP/M system; standard CP/M has a base of 0, but some CP/M systems (such as for the TRS-80 Model I and Heath/Zenith H89/Z89) start physical RAM memory at a higher address; equate BASE to the starting RAM memory address of your system
CPRLOC	This is the starting address of ZCPR; set the second CPRLOC equate to the address you obtain from BDOSLOC
RAS	This is an equate which masks out selected ZCPR command functions for security purposes on Remote Access Systems such as Bulletin Boards; the masked out functions currently include SAVE, ERA, REN, JUMP, GO, and GET; set RAS to TRUE to mask these out or FALSE to leave them in
SUBA	This is an equate which determines the drive onto which ZCPR will look for an executing Indirect Command File. If the basic philosophy of the Indirect Command File described above is to be maintained, this symbol should be set to TRUE (look on drive A: for the \$\$.SUB file); if not, this symbol should be set to FALSE (look on the default drive from the \$\$.SUB file). To review, the basic philosophy of the Indirect Command File is that any sequence of commands which may be issued from the console (within reason, which means NOT to erase a \$\$.SUB file) may also be issued from within an Indirect Command File, and the resultant execution should be identical (same functions performed).
CLEVEL3	This equate enables or disables extended Command Level 3 Processing. If set to TRUE, extended Command Level 3 Processing is enabled and the user command line is automatically capitalized, the terminating zero is placed at the end of the buffer, and the internal CIBPTR is set correctly (see later for more information).

Customization Symbols

The following symbols are provided for further customization of ZCPR to a user's particular tastes and hardware facilities.

Option Name	Function
NLINES	Number of lines on the user's CRT for paging
WIDE	This equate is used to select a narrow or wide display under the DIR command; if WIDE is equated to TRUE, each file name is separated by two spaces, a FENCE, and two more spaces; if WIDE is equated to FALSE, each file name is separated by one space, a FENCE, and one more space
PGDFLT	This is the Paging Default flag for the TYPE command; if PGDFLT is set to TRUE, the TYPE command will page its output by default and the P option on the TYPE command (see below) will prohibit paging; if PGDFLT is set to FALSE, the TYPE command will NOT page its output by default and the P option will enable paging
PGDFLG	This sets the option character in the command line for the TYPE command (the 'P' mentioned above); if the user wishes to change this option character, he need only change this equate
MAXUSR	This is the largest user number recognized by the USER command; if the user wishes to protect the higher user areas, he may set this symbol to the highest area normally accessable; 15 is the largest permitted value for MAXUSR
SYSFLG	This is the option character for the DIR command line which is used to specify that DIR search All Files (both \$SYS and \$DIR) for its display; the distributed default for this is 'A'
SOFLG	This is the option character for the DIR command line which is used to specify that DIR search ONLY the \$SYS files for its display; the distributed default for this is 'S'
SUPRES	Set SUPRES to TRUE to suppress printing the user number when the user is under User Number 0 or set SUPRES to FALSE to ALWAYS display the User Number with the CPR prompt; with SUPRES set to TRUE, a user on B: in user 0 sees 'B>' as the prompt, but with SUPRES set to FALSE, a user on B: in user 0 sees 'B0>' as the prompt

DEFUSR	This is the CPR-default user number which is searched in the command hierarchy for the COM files (distributed as 0); the DFU changes this temporarily until a Warm Boot or Cold Boot is done, at which time the search reverts to this value
SPRMPT	This is the CPR prompt character which indicates that a SUBMIT file is in execution; by default it is set to '\$', so prompts like 'A\$' appear during SUBMIT file execution
CPRMPT	This is the CPR prompt character which indicates that the CPR is awaiting a user console command; by default it is set to '>', so prompts like 'A>' appear during user input to the CPR
NUMBASE	This is the escape character used by those commands which require a DECIMAL number as an argument; placing this character after the number argument switches the base to HEXADECIMAL; for example, 'SAVE 15 MYFILE' can be expressed as 'SAVE FH MYFILE' if NUMBASE is set to 'H' (the default)
SECTFLG	This character constant is the suffix option for the SAVE command which specifies that sectors, as opposed to pages, are to be saved; the default value is 'S'
FENCE	This is the character printed to separate entries in a directory listing; it's default value is ' '

Patching SUBMIT.COM

SUBMIT.COM may be patched to run with ZCPR by the following procedure (this is recommended if the user does not have SUB.COM). This patch simply makes it always place the \$\$\$SUB file on Drive A:. Illustrative terminal session follows:

```
A>ddt b:submit.com
DDT VERS 2.0
NEXT PC
0600 0100
-s5bb      <-- Patch is at 5BB Hex
05BB 00 1  <-- Change 0 (default drive) to 1 (drive A:)
05BC 24 .
-d5b0 5cf  <-- See change
05B0 00 00 00 00 00 30 30 31 20 24 01 24 24 24 20 .....001 $.$$$
05C0 20 20 20 20 53 55 42 00 00 00 1A 1A 1A 1A 1A 1A      SUB.....
-^C        <-- Done
A>save 5 newsubmt.com  <-- Save new SUBMIT.COM file
```

Part B
Usage Instructions and Explanation of Commands

The following instructions are written with the assumption that the reader is quite familiar with how to use CP/M 2.2 and its CCP. ZCPR is written as a logical extension of the CP/M 2.2 CCP philosophy and should be addressed as such.

The ZCPR Command Hierarchy Search

The first, and most basic thing, to learn about ZCPR is the order in which it searches for a COM file for execution or a file specified by the GET command. Under the CP/M 2.2 CCP, if the specified COM file command was not found on the current drive in the current user area, the CCP aborted with an error message. ZCPR, however, continues searching from this point a maximum of two more levels. This command hierarchy search was outlined above and is described here in further detail.

1. If the command is of the form 'COMMAND' and NOT 'd:COMMAND', the CPR-resident command list is searched for a match. If the match is found, the CPR-resident command is immediately processed. If the match is not found or the command is of the form 'd:COMMAND', the next step is taken. Note that the 'd:COMMAND' form is good for executing a command COM file which has the same name as a CPR-resident command (such as SAVE or DIR).

2. If the command is of the form 'd:COMMAND', disk drive 'd:' is temporarily logged in for the purpose of the command search. Otherwise, the currently logged-in drive is used.

3. Now the file named COMMAND.COM is searched for. If found, it is loaded into memory starting at 100H and executed. If not, proceed to step 4.

4. Now that the first search for COMMAND.COM has failed, the CPR checks to see if the user is under the current Default User Number. The Default User Number may be that set by the DEFUSR equate in the CPR or that set by the user via the DFU command. DEFUSR is in effect if DFU has not been issued since the last Warm or Cold Boot, and DFU is in effect if it was issued since the last Warm or Cold Boot. If the user is NOT under the current Default User Number, ZCPR temporarily logs him into it and searches the directory. If COMMAND.COM is found, it is loaded as described above and executed. If not, ZCPR proceeds to the next step.

5. The user is now in the Default User Number, and at this point, ZCPR checks to see if the user is on disk drive A:. If not, it temporarily logs into A: and searches the default user number of A: for COMMAND.COM. If found, it is loaded as described above and executed. If not, ZCPR prints the command name as an error message and returns to command input mode, aborting the SUBMIT file if COMMAND came from it.

In all cases of the search above, if COMMAND.COM is found, after it is loaded into memory, ZCPR resets the user to his original disk drive and user number. Hence, the files referenced by the user by default are obtained from this environment.

To illustrate this command hierarchy search, consider the following examples:

Example 1: DEFUSR equ 0 {default user number is 0}

```
B10>           <-- User is on Drive B:, User Number 10
B10>ASM TEST.BBZ    <-- User wishes to assemble TEST.ASM in
                  Drive B:, User 10
<-- At this point, ZCPR looks on B:/10 for ASM.COM, fails,
      looks on B:/0, fails, and finally looks on A:/0; it
      finds ASM.COM here and goes back to B:/10 for the file
```

Example 2: DEFUSR equ 0 and DFU issued

```
B10>           <-- User is on Drive B:, User Number 10
B10>DFU 5      <-- User Selects User 5 as default
B10>ASM TEST.BBZ    <-- As above
<-- At this point, ZCPR looks on B:/10 for ASM.COM, fails,
      look on B:/5, fails, and finally looks on A:/5; it
      fails here also and prints ASM? as an error message
```

Example 3: DEFUSR equ 0

```
B>           <-- User is on Drive B:, User Number 0
B>ASM TEST.BBZ    <-- As above
<-- At this point, ZCPR looks on B:/0 for ASM.COM, fails,
      looks on A:/0, fails, and prints error message
```

Example 4: DEFUSR equ 0

```
A10>           <-- User is on Drive A:, User Number 10
A10>ASM TEST.AAZ    <-- As above, but file on A:
<-- At this point, ZCPR looks on A:/10 for ASM.COM, fails,
      looks on A:/0, fails, and prints error message
```

Another Example:

For example, if the user is logged into Drive B: in User Area 10, the Default User Number is 0, and the following COM files are present as indicated --

```
WM.COM on Drive A: in User 0
MBASIC.COM on Drive A: in User 0 and on
Drive B: in User 0
TEST.COM on Drive B: in User 10 and Drive B:
in User 0
```

then the following happens when the following commands are issued from the console (or Indirect Command File):

```
B10>WM TEST2.TXT
\  \  \_ File to be edited
\  \_ Invoke the WM.COM file (Word Master editor)
\__ User is on Drive B: in User Area 10
```

Results:

ZCPR searches B: User 10, B: User 0, and A: User 0 for WM.COM; it finds WM.COM in A: User 0, loads it, logs the user back into B: User 10, and executes it.

```
B10>MBASIC
\  \_ Invoke the MBASIC.COM file (MBASIC Interpreter)
\__ User is on Drive B: in User Area 10
```

Results:

ZCPR searches B: User 10 and B: User 0 for MBASIC.COM; it finds MBASIC.COM in B: User 0, so it doesn't bother to look on A: User 0. MBASIC.COM is then loaded and executed as described in the previous example.

```
B10>TEST
\  \_ Invoke the TEST.COM file (TEST program)
\__ User is on Drive B: in User Area 10
```

Results:

ZCPR searches B: User 10 for TEST.COM; it finds TEST.COM in B: User 0, so it doesn't bother to look further (if it had, it would have found TEST.COM in B: User 0). TEST.COM is then loaded and executed as described above.

```
B10>TEST2
\  \_ Invoke the TEST2.COM file (TEST2 program)
\__ User is on Drive B: in User Area 10
```

Results:

ZCPR searches B: User 10, B: User 0, and A: User 0 for TEST2.COM; it doesn't find it, so it issues the error message 'TEST2?', which says it couldn't find TEST2.COM.

The ZCPR-Resident Commands

The following pages describe the ZCPR-Resident Commands. These are commands located within ZCPR itself which are executed from within ZCPR. The phrases <afn> and <ufn> refer to ambiguous file name and unambiguous file name as per the CP/M convention.

Command: DIR

Function: To Display a listing of the names of the files on disk
Forms:

DIR <afn>	<-- Displays \$DIR files
DIR <afn> S	<-- Displays \$SYS files
DIR <afn> A	<-- Displays both \$DIR and \$SYS files

Customization Variables:

WIDE	SYSFLG	SOFLG	FENCE
------	--------	-------	-------

Examples:

DIR *.ASM	<-- All \$DIR .ASM files
DIR *.COM S	<-- All \$SYS .COM files
DIR *.COM A	<-- All .COM files

Notes:

If a file is scanned for and no such name exists on disk, the 'No Files' message will appear. However, if a file is scanned for and the name exists as a \$SYS file and \$DIR files are being scanned for, no file name is displayed but the 'No Files' message does NOT appear. For example, if TEST.COM is a \$SYS file and 'DIR TEST.COM' is issued, no message appears. If 'DIR TEXT.COM' is issued and TEXT.COM does not exist on disk, the 'No Files' message is displayed.

Command: ERA

Function: To Erase the specified \$R/W files from disk

Forms:

ERA <afn>	<-- Erase both \$DIR and \$SYS files
-----------	--------------------------------------

Customization Variables:

WIDE	FENCE
------	-------

Examples:

ERA *.ASM	<-- Erase all .ASM files
ERA *.*	<-- Erase all files

Notes:

If a \$R/O file is encountered, a BDOS error message will be displayed and the procedure is stopped. The user is unsure at this time as to which files have been erased and which have not and should check. Sorry for this problem! The ERASE command (to be given to SIG/M by RLC in the near future) is a solution to this problem.

Command: LIST

Function: To Print the specified file on the CP/M LST: device
Forms:

LIST <ufn> <-- Print the file (no paging)

Customization Variables:

-None-

Examples:

LIST TEST.TXT <-- Print TEST.TXT on LST:

Notes:

If the file has a \$SYS attribute, it will be found as well as those with \$DIR attributes.

Command: TYPE

Function: To Print the specified file on the CP/M CON: device
Forms:

TYPE <ufn> <-- Print the file with the paging deflt

TYPE <ufn> P <-- Print the file with the paging deflt negated

Customization Variables:

NLINES PGDFLT PGDFLG

Examples:

TYPE TEST.TXT

TYPE TEST.TXT P

Notes:

When the display pauses during paging, type any char to continue or ^C to abort. ^S also works.

Command: SAVE

Function: To Copy the TPA starting at 100H to disk

Forms:

SAVE <Number of Pages> <ufn> <-- <Number of Pages> in DEC

SAVE <Number of Pages>H <ufn> <-- <Number of Pages> in HEX

SAVE <Number of Sectors> <ufn> S <-- Number of sectors

SAVE <Number of Sectors>H <ufn> S <-- Number of sectors

Customization Variables:

NUMBASE RAS

Examples:

SAVE 15 MYFILE.TXT <-- 15 pages saved

SAVE FH MYFILE.TXT <-- 15 pages saved

SAVE 10H MYFILE.TXT S <-- 16 sectors (8 pages) saved

Notes:

If the file name to be saved already exists, then SAVE will exit with the message 'Delete File?'; if the user REALLY wants to save under this name, he may then type Y or y and the current file will be deleted and then recreated containing the specified part of the TPA.

Command: REN

Function: To Change the name of a disk file

Forms:

REN <ufn new>=<ufn old>

Customization Variables:

RAS

Examples:

REN NEWFILE.TXT=OLDFILE.TXT

Notes:

If <ufn new> already exists, the message 'Delete File?' will be printed and the user may respond with Y or y to delete the current <ufn new> and then rename <ufn old> to <ufn new>.

Command: USER

Function: To Change the current user number

Forms:

USER <User Number> <-- <User Number> in DEC

USER <User Number>H <-- <User Number> in HEX

Customization Variables:

-None-

Examples:

USER 15 USER FH USER 0

USER <-- Same as USER 0

Notes:

-None-

Command: DFU

Function: To Temporarily Change the default user number for the command hierarchy search

Forms:

DFU <User Number> <-- <User Number> in DEC

DFU <User Number>H <-- <User Number> in HEX

Customization Variables:

-None-

Examples:

DFU 15 DFU FH DFU 0

DFU <-- Same as DFU 0

Notes:

See above for explanation.

Command: JUMP

Function: To "call" the subroutine at the specified page address
Forms:

JUMP <Address> <CmdParms> <-- <Address> in HEX

Customization Variables:

NUMBASE RAS

Examples:

JUMP E000 or JUMP E000H <-- Jump to E000H

JUMP <-- Jump to 000H

JUMP 0 <-- Jump to 000H

Notes:

JUMP performs a subroutine "call", so the called routine may return to the ZCPR by either a RET or a Warm Boot.

Command: GO

Function: To "call" the subroutine starting at 100H

Forms:

GO <CmdParms> <-- Execute reentrant at 100H

Customization Variables:

RAS

Examples:

GO *.ASM <-- Assuming XDIR is loaded,
gives directory of *.ASM

Notes:

This command is identical in function to JUMP 100H; JUMP, however, leaves the address as the first entry in CP/M BASE + 80H (the input line buffer), while GO has no such address.

Command: GET

Function: To load a file from disk into memory starting at the specified page

Forms:

GET <Address> <ufn> <-- <Address> in HEX

Customization Variables:

NUMBASE RAS

Examples:

GET 8000 TEST.80 <-- Load TEST.80 starting at 8000H

GET 100 TEST.80 or GET 100H TEST.80 <-- Load TEST.80
starting at 100H

GET 0 TEST.80 <-- Load TEST.80 starting at 000H

Notes:

GET searches for the specified file according to the same command hierarchy search employed by the ZCPR command scanner. Hence, if the user is on B:/10 and the file is on A:/0 with the current default user number at 0, GET will search from B:/10 to B:/0 to A:/0 in looking for the file.

ZCPR Error Messages

The following are the error messages issued by ZCPR and their meanings.

Message Meaning

? Printed after a command or an argument means that such was invalid

No File From DIR, this means that DIR did not locate any files
Also from ERA with the same meaning

All? Issued in response ERA *.* , asks the user is he really wants to erase all the files. Unlike under the original CP/M 2.2 CCP, single character input is required (Y or y for yes and anything else for no) with NO <CR> to end the line

Full From SAVE, means that there is not enough space on disk
From GET or command load by CPR, means that there is not enough space in memory

Delete File?

From REN or SAVE, means that the file specified already exists on disk and the user may type Y or y to delete it and proceed with the REN or SAVE function

Part C
ZCPR Command Levels and How to Use Them

ZCPR Version 1.0 and beyond supports three distinct command levels in its implementation. Each level constitutes a different way to issue a command for ZCPR to process.

Command Levels 1 and 2 are common to all implementations of CP/M and CP/ZM from CP/M Version 1.4. Command Level 1 is that command level in which the command is issued by the user from his console terminal. The prompt 'd>' or 'du>' appears on the terminal, and the user is allowed to enter the command with editing from the terminal. Command Level 2 is that command level in which the command is entered from an executing \$\$.SUB file.

In both cases, the command is stored in the internal ZCPR buffer called CIBUFF (Command Input BUFFer). Under both Command Levels 1 and 2, the command is placed into this buffer, the characters of the command line are capitalized, a character count which indicates the number of characters in the command line is stored in CBUFF (the byte before CIBUFF), an ending binary 0 is placed after the last character in the command line, and the internal pointer CIBPTR (Command Input Buffer PointeR) is set to point to CIBUFF (the first character of the command line).

Command Level 3 is an extended concept to Command Levels 1 and 2 which is specifically supported by ZCPR Version 1.0 and beyond. This command level allows a transient program to place a command line into CIBUFF and the character count into CBUFF and have this command line executed by ZCPR. Once control is transferred to ZCPR to execute the command line, the transient program which placed the command line loses control and the command is executed exactly as though it had been typed by the user at his console terminal.

In order for a transient program to utilize the Command Level 3 facility, this program MUST do the following:

1. Locate the ZCPR. Since the ZCPR is ALWAYS 2K bytes in size and located directly under the BDOS, the transient can locate the ZCPR by examining the BDOS entry page address at location 7 and subtracting 8 from this number (8 pages = 2K bytes). The resulting number is the base page address of ZCPR.

2. Store the command line in CIBUFF and the character count in CBUFF. Knowing the base page address of ZCPR, the following information is useful in doing this:

```
ORG  CPRLOC    ;Base Address of ZCPR
JMP  CPR       ;Enter ZCPR and Execute Default Cmd
JMP  CPR1      ;Enter ZCPR and Don't Execute
MBUFF:  DB  BUFSIZE ;Size of CIBUFF in bytes
CBUFF:  DS  1      ;Number of Bytes in Command Line
CIBUFF: DS  BUFSIZE ;Buffer for Command Line
          DS  1      ;Buffer for Ending 0 (set by ZCPR)
CIBPTR: DS  2      ;Address of CIBUFF (set by ZCPR)
```

3. Obtain the User/Disk Flag. Location 4 contains this number, but the user may select a flag of his choice. This flag is one byte long, and the high-order nybble (4 bits) contains the user number and the low-order nybble contains the disk number to process the command from. The User/Disk Flag is to be passed to ZCPR in the C Register.

4. When ready, transfer control to ZCPR to process the command by JMPing to the base address of ZCPR. The first JMP in the JMP Table given above is at this address. At this time, ZCPR will log in the user and disk in the User/Disk Flag and process the Command Level 3 Command Line.

The following is a sample program which illustrates the steps outlined above:

```

;
; Demonstration of Command Level 3 Facility by RLC
;
udflag equ 4      ;Address of User/Disk Flag
bdos   equ 5      ;Address of BDOS Entry Point

org 100h

lxi d,prmpt ;Print User Prompt
mvi c,9    ;PRINT function
call bdos

lhld bdos+1 ;Get address of BDOS
mov a,h   ;High-Order Address in A
sui 8     ;A=High-Order Address of CPR
mov h,a   ;HL=Address of CPR
mvi l,0
shld cpr  ;Save address in buffer

lxi d,6   ;Point to command line buffer
dad d    ;HL points to command line buffer
xchg    ;DE points to command line buffer
mvi c,10  ;READLN into this buffer
call bdos

lhld cpr  ;Get Address of CPR
lda udflag ;Get User/Disk Flag
mov c,a   ; ... in C
pchl    ;Run Command Line

cpr: ds 2      ;CPR Address buffer
prmpt: db 'User Command? $'

```

Enjoy using ZCPR!

-- RLC