MICROSHELL™ NEWSLETTER

SPRING 1983

In this our first MicroShell newsletter, we will report the current status of the MicroShell release and discuss some of the questions most frequently asked by MicroShell users.

Version 2.0 of MicroShell has been shipping now since December 1982 and the general response to the added features has been very good.

INTERACTION BETWEEN VARIOUS EDITORS AND MICROSHELL

As far as we know, none of our users has found an editor that doesn't work with MicroShell. There are a couple of things to watch out for. Some editors – notably Mince, by Mark of the Unicorn – do not fill up the last sector of the file with control Z's. MicroShell gets confused when it is executing a shell file built with an editor that does this. The fix is to fill up the last sector in the shell file with control Z's. This can be done most easily by just having MicroShell redirect the output from typing the file into another file, e.g.:

type command.sub>command1.sub

MicroShell fills the last sector with control Z's during output redirection.

Also note that some editors *write* to a scratch file (again Mince does.) MicroShell's file search feature is not designed for file writes, only for reads. One thing you can do to create cross-user links is to use one of the MicroTools, "ln", to make a link to the scratch file in user 0.

EQUAL SIGN HANDLING

The equal sign (=) is a special character to MicroShell because of its use in separating the arguments in a rename command. MicroShell puts in a space on each side of the equal sign. This works fine for the rename command but causes problems with other programs that require an equal sign in the command line (e.g. Sorcim's ACT assembler, MicroSoft's linker, etc.) The temporary fix is to preceed the equal sign with a backslash (\) to cause MicroShell to ignore the special meaning of the equal sign. If you are running the command from a shell file, you must preceed the equal sign with 2 blackslashes (\ \). We will correct this problem in the next release of MicroShell.

INTERACTION BETWEEN PIP AND MICROSHELL

Digital Research's Pip utility for file transfer requires one precaution when operating under MicroShell. Normally MicroShell automatically recognizes when the user is running Pip and temporarily turns off automatic file search. This is necessary to prevent Pip from erasing files on the wrong drive! The user can override MicroShell's recognition that Pip is being run by prepending a disk drive designator, e.g. a: pip. MicroShell will not recognize that Pip is being run if the program name is not just plain "pip". Beware also of a public domain patch to pip that renames it as RPIP; that will cause the same problem.

DOUBLE QUOTES (") AND MICROSHELL

The double quote character is treated specially by MicroShell to group arguments to a shell file. For example, if you had a shell file "command.sub" that had a $1 argument substitution in it, then saying:

command "this is the argument"

would substitute the four words inside the double quotes for $1 in the shell file and strip the double quotes out of the command line passed to the program. This is undesirable when a program needs to see the double quotes to do its own argument grouping. The solution we recommend is to make the following patch to MicroShell which causes MicroShell to treat single quotes (') as it used to treat double quotes. Then double quotes are no longer a special character to MicroShell and they are passed through to the program to handle.

Double Quote to Single Quote Patch, File: SH.COM Vers: 2.0

LOCATION	WAS	CHANGE TO
0FF9	22	27
109A	22	27 (ALL VALUES IN HEX)
1A15	22	27

MULTIPLE COMMAND SEPARATOR

MicroShell uses a semicolon (;) to separate multiple commands on a command line. Some users have indicated a desire to use a different character. The following patch will do that:

Multiple Command Separator Patch, File: SH.COM Vers: 2.0

LOCATION	WAS	CHANGE TO
0FEA	3B	XX (ALL VALUES IN HEX)
1A18	3B	XX

where XX is the hex value of the desired command separator. Be sure not to use one of the characters that have special meaning to MicroShell.

EDITING THE COMMAND LINE

Some users have experienced a problem in executing a command line after editing it. The following patch will probably correct the problem:

Command Line Editing Patch, File: SH.OVR Vers: 2.0

LOCATION	WAS	CHANGE TO
21EE	71	74 (ALL VALUES IN HEX)

NEW GENERATION SYSTEMS, inc.

2153 Golf Course Drive
Reston, VA 22091
(703) 476-9143

PASSING VALUES TO A SHELL FILE FROM A PROGRAM

MicroShell version 2.0 has two extended shell commands to pass a shell file a string or a number from a program – %memstr and %memnum respectively (see page 45 of the MicroShell manual.) The %memnum function ran into interference passing a number in location 80 and 81 Hex. The following patch changes the location to OFE and OFF Hex:

%memnum Patch, File: SH.OVR Vers: 2.0

LOCATION	WAS	CHANGE TO
14.64F	80	FE (ALL VALUES IN HEX)

The program should now put the number in OFE and OFF hex (low byte in OFE). The number must be binary not ASCII.

THE %RETURN EXTENDED SHELL COMMAND

The %return statement in a shell file will cause processing to return to the previous level – to the last shell file, if the current shell file was invoked from a shell file (i.e. a "nested" shell file) or the command level if there is no nesting. On page 42 of the MicroShell Version 2.0 manual, the specification of the %return statement allows an optional argument. This argument assigns a value to a variable if the %return is executed. This optional feature does not work and should not be used. %return by itself, works correctly.

RENAME ERROR MESSAGE

A few of the first copies of Version 2.0 of MicroShell shipped before 12/13/82 had a bug in the rename (REN) function which caused the wrong error message to be printed when a rename error occurred. If you get the right error message when you try to rename a nonexistant file or when the new name already exists, you don't need this patch:

Rename Patch, File: SH.COM Vers: 2.0 (prior to 12/13/82)

LOCATION	WAS	CHANGE TO
15A3	CA	D0
15CB	C9	D5 (ALL VALUES IN HEX)
15D1	D5	C9
15EB	D0	CA