

SuperSoft Utility Pack II

Commands in utility packs for the most part have a consistent command syntax. Each command consists of a command name followed by arguments. The arguments are separated by spaces (not tabs). If spaces are desired in an argument, they must be placed in double quotes. The arguments tend to be options, filenames, or input/output redirections. The options are preceded by a dash ('-') in most cases. Filenames must be unambiguous ('?' and '*' are not treated specially).

These commands make use of a "standard input" and a "standard output." Normally, the standard input is equivalent to the console keyboard and the standard output is equivalent to console output. However, these assignments can be changed by adding an input/output redirection argument to the command line. If an argument of the form >file is added to the command line, the standard output is changed to file. If an argument of the form <file is added to the command line, the standard input is changed to file. (These arguments must be preceded by a space).

Ambiguous file specifications are not (as yet) accepted by any utility command. Also there is no provision for redirection the standard input/output to a device such as PRN: .

The commands in the utility pack is described in the following pages. Each command is given in synopsis form, with square brackets ('[' and ']') indicating optional forms. Upper case forms in the synopses are typed by the user exactly as they appear. Lower case forms are to be replaced by some item such as a filename.

There are several inconveniences associated with CP/M command line processing that affect utility packs: first command lines are forced into upper case, making entry of lower case impossible; second, command lines are limited to a mere 126 bytes; third, the maximum number of arguments is sixteen.

CAT -- Concatenate files.

Synopsis:

```
CAT [ - ] [ filename ] ...
```

CAT concatenates the named files. If a minus ('-') is specified the standard input is read. Control-Z is considered end of file.

See also:

PR

DATE -- Display or set the system date.

Synopsis:

```
DATE [ date_string ]
```

DATE displays or sets the date string contained in file 'DATE.DTE'. The date string is entered into 'DATE.DTE' as the new system date. If date_string is absent, DATE will display the information found in 'DATE.DTE' up to the first Control-Z. If 'DATE.DTE' does not exist DATE will display '00/00/00'. No format testing is done on date_string, so any string is allowed as the date.

Bugs:

IO redirection is not available.

See also:

PR

DC -- On-line desk calculator.

Synopsis:

DC

DC is a thirteen BCD digit, floating point arithmetic calculator.

DC accepts commands from the standard input after prompting with a '>'. Input to DC is in Reverse Polish Notation.

The following is a list of DC commands:

<number>	- Push number onto stack.
'h' or 'H'	- Display online help.
'l'nn or 'L'nn	- Pop top of stack into register nn.
'p'	- Display top of stack.
'P'	- Display top of stack and pop.
'q' or 'Q'	- Leave DC.
's' or 'S'	- Swap top two stack elements.
'w'nn or 'W'nn	- Push register nn onto stack.
'x' or 'X'	- Display entire stack.
'z' or 'Z'	- Clear stack.
'\$'	- Extended function flag (see below).
<carriage ret.>	- Start processing.

The following is the list of basic mathematic functions that DC performs. These functions operate by popping the top two stack elements, performing the operation, and then pushing the result back onto the stack.

'+'	- Addition.
'-'	- Subtraction.
'*'	- Multiplication.
'/'	- Division.
'%'	- Modulus.
'^'	- Exponentiation by an integer.

There is also a set of extended math functions. These functions are accessed by first entering the extended function flag ('\$') and then the function name. All of these functions pop the top element of the stack, compute the result, and push the result back onto the stack.

All of the functions, except for 'I', approximate their result by evaluating a power or Taylor series in "single precision," so it is possible for some results to be less precise than thirteen digits. All trigonometric functions return radians.

The functions and their names are:

'a' or 'A'	- Compute the natural antilog.
'c' or 'C'	- Compute the cosine (radians).
'i' or 'I'	- Compute the integer part.
'l' or 'L'	- Compute the natural log.
'q' or 'Q'	- Compute the square root.
's' or 'S'	- Compute the sine (radians).

't' or 'T'	- Compute the tangent (radians).
'!'	- Factorial.

The following functions are also extended math functions, and are accessed the same way, they however operate on the top two stack elements. These functions return '1.000000000000' if the expression is true and '0.000000000000' if false.

'<'	- Less than.
'>'	- Greater than.
'<='	- Less than or equal to.
'>='	- Greater than or equal to.
'='	- Equal.
'#'	- Not equal.

Up to eighty characters may be entered on the command line at a time. Numbers must be terminated by a space or a command.

Bugs:

IO redirection is not available.

Sample session:

The following is a sample DC session.

```
>00001 2. 3PPPX
3.000000000000
2.000000000000
1.000000000000
> P
Stack empty.
>1
>2
>3*+p
7.000000000000
>5 6 7x
7.000000000000
6.000000000000
5.000000000000
7.000000000000
>$=x
0.000000000000
5.000000000000
7.000000000000
>q
```

DIFF -- Compare two files line by line.

Synopsis:

```
DIFF [ -A ] file1 file2
```

DIFF does a line by line comparison of file1 and file2. If a line is found in file1 but not in file2 the line is displayed, prefaced by '^*' on the standard output device. The line is displayed, prefaced by '^.' if in file2 but not file1. If option -A is selected all other lines are displayed, prefaced by '^ '.

Note:

DIFF reads the file into memory. If the file won't fit, DIFF dies. In order to make this less of a problem, DIFF is supplied in four versions: DIFF.COM (24k version), which handles 2,000 lines and 10,000 bytes of source; DIFF32.COM (32k version), which handles 3,000 lines and 17,000 bytes; DIFF48.COM (48k version), which handles 4,000 lines and 24,000 bytes; and DIFF56.COM (56k version), which handles 5,000 lines and 31,000 bytes. (Subtract 0E00h (about 4k) from these figures to get the required TPA size).

SPLIT can also be used to keep DIFF alive.

See also:

CMP (Utilities I), SPLIT (Utilities II).

INUSE -- Reserve a terminal.

Synopsis:

```
INUSE
```

INUSE clears the terminal screen and displays 'IN USE' in large letters across the screen. INUSE ends on any keystroke.

INUSE should request a password to prevent unauthorized access to the terminal. However, under CP/M, control-C cannot be ignored by a program waiting for input.

INUSE does not use IO redirection.

PR -- Output formatter.

Synopsis:

```
PR [ -Lnn -Wnn -nn -T -PR -PU -Hstring1 ] [ filen ... ]
```

PR prints the list of files on the standard output with optional header/trailer lines. The header contains a page number, the filename of the file being printed, and the date_string (if present--see DATE). PR expands tabs and form feeds using the appropriate linefeed or space characters.

-Wnn sets the line length to nn (default is 79). -Lnn sets the page length to nn (default is 66). -nn causes the output to be formatted into nn columns (default is 1). -T deletes the header/trailer lines. -PR sends the output to the printer device. -PU sends the output to the punch device. -Hstring1 substitutes string1 for the filename in the header.

See also:

CAT (Utilities I), DATE (Utilities II), WIDTH (Utilities I).

Bugs:

IO redirection is not available.

RPL -- Substitute strings.

Synopsis:

RPL string1 string2 file1 ...

RPL replaces all occurrences of string1 with string2 in the list of files. The original of each file is renamed using the extension of '.BAK'.

Bugs:

IO redirection is not available. Files with the extension of '.BAK' cannot be specified to RPL.

See also:

GREP (Utilities I), TR (Utilities II).

SLEEP -- Pause.

Synopsis:

SLEEP nn

SLEEP executes for nn time units (a time unit is one second on a 4MHz system). This is useful in a 'submit' file when a pause for operator interaction is needed, such as when a disk needs to be mounted, or forms need to be changed in a printer.

SPLIT -- Split a file.

Synopsis:

SPLIT [-nn] file1

SPLIT copies file1 into multiple files of one hundred lines each. These files are names XAA, XAB, and so on. If -nn is specified, the file is split at multiples of nn lines.

A maximum of 26 files will be created.

Bugs:

IO redirection is not available.

TR -- Translate bytes.

Synopsis:

```
TR string1 string2 file1 [ file2 ...]
```

TR performs a byte by byte translation. string1 and string2 are made up of ASCII characters or '\ ' followed by three decimal digits which will be taken to be the decimal value of the byte's code. string2 must at least contain as many byte values as string1. '\ ' represents a single slash, and '\F' indicates that the rest of the string is to be taken as the name of a file to read in order to form the translation information. The file is taken to contain pairs of strings: each a string1-string2 pair, carriage returns, line feeds, spaces, tabs, and control-Z's are string separators.

In the following example 'A' is translated into 'Z', byte code 5 into byte code 6, and byte code 10 into 'P' for the files 'TEST1.DAT' and 'TEST2.DAT'.

Note that the command line processing of CP/M requires that lower case ASCII characters be entered to TR via the '\ ' escape convention.

```
TR A\005\010 Z\006p TEST1.DAT TEST2.DAT
```

Byte codes must be 3 digits long, padded with leading zeros if necessary. Valid byte codes run from 0 to 255.

Note that text files usually are padded at the end with control-Z's, therefore translating control-Z produces files that are hard to manipulate with standard CP/M programs.

Bugs:

IO redirection is not available. Files with the extension of '.BAK' cannot be specified to RPL.

See also:

RPL