



## ***Linux Ubuntu 26.04 Manual Pages on command 'bzless.1'***

**\$ man bzless.1**

BZMORE(1)

General Commands Manual

BZMORE(1)

### NAME

bzmore, bzless - file perusal filter for crt viewing of bzip2 compressed text

### SYNOPSIS

bzmore [ name ... ]

bzless [ name ... ]

### NOTE

In the following description, bzless and less can be used interchangeably with bzmore and more.

### DESCRIPTION

Bzmore is a filter which allows examination of compressed or plain text files one screenful at a time on a soft-copy terminal. bzmore works on files compressed with bzip2 and also on uncompressed files. If a file does not exist, bzmore looks for a file of the same name with the addition of a .bz2 suffix.

Bzmore normally pauses after each screenful, printing --More-- at the bottom of the screen.

If the user then types a carriage return, one more line is displayed. If the user hits a space, another screenful is displayed. Other possibilities are enumerated later.

Bzmore looks in the file /etc/termcap to determine terminal characteristics, and to determine the default window size. On a terminal capable of displaying 24 lines, the default window size is 22 lines. Other sequences which may be typed when bzmore pauses, and their effects, are as follows (i is an optional integer argument, defaulting to 1) :

i<space>

display i more lines, (or another screenful if no argument is given)

`^D` display 11 more lines (a ``scroll"). If `i` is given, then the scroll size is set to

`i`.

`d` same as `^D` (control-D)

`iz` same as typing a space except that `i`, if present, becomes the new window size. Note that the window size reverts back to the default at the end of the current file.

`is` skip `i` lines and print a screenful of lines

`if` skip `i` screenfuls and print a screenful of lines

`q` or `Q` quit reading the current file; go on to the next (if any)

`e` or `q` When the prompt `--More--(Next file: file)` is printed, this command causes `bzmore` to exit.

`s` When the prompt `--More--(Next file: file)` is printed, this command causes `bzmore` to skip the next file and continue.

`=` Display the current line number.

`i/expr` search for the `i`-th occurrence of the regular expression `expr`. If the pattern is not found, `bzmore` goes on to the next file (if any). Otherwise, a screenful is displayed, starting two lines before the place where the expression was found. The user's erase and kill characters may be used to edit the regular expression. Erasing back past the first column cancels the search command.

`in` search for the `i`-th occurrence of the last regular expression entered.

`!command`

invoke a shell with `command`. The character ``!` in `"command"` are replaced with the previous shell command. The sequence `"\!"` is replaced by `"!"`.

`:q` or `:Q`

quit reading the current file; go on to the next (if any) (same as `q` or `Q`).

`.` (dot) repeat the previous command.

The commands take effect immediately, i.e., it is not necessary to type a carriage return.

Up to the time when the command character itself is given, the user may hit the line kill character to cancel the numerical argument being formed. In addition, the user may hit the erase character to redisplay the `--More--` message.

At any time when output is being sent to the terminal, the user can hit the quit key (normally `control-\`). `Bzmore` will stop sending output, and will display the usual `--More--` prompt. The user may then enter one of the above commands in the normal manner. Unfortunately, some output is lost when this is done, due to the fact that any characters waiting

in the terminal's output queue are flushed when the quit signal occurs.

The terminal is set to noecho mode by this program so that the output can be continuous.

What you type will thus not show on your terminal, except for the / and ! commands.

If the standard output is not a teletype, then bzmored acts just like bzcat, except that a header is printed before each file.

## FILES

/etc/termcap      Terminal data base

## SEE ALSO

more(1), less(1), bzip2(1), bzdiff(1), bzgrep(1)

BZMORE(1)