NAME

script − make typescript of terminal session

SYNOPSIS

script [options] [*file*]

DESCRIPTION

script makes a typescript of everything displayed on your terminal. It is useful for students who need a hardcopy record of an interactive session as proof of an assignment, as the typescript file can be printed out later with **lpr**(1).

If the argument *file* is given, **script** saves the dialogue in this *file*. If no filename is given, the dialogue is saved in the file **typescript**.

OPTIONS

Below, the *size* argument may be followed by the multiplicative suffixes KiB (=1024), MiB (=1024*1024), and so on for GiB, TiB, PiB, EiB, ZiB and YiB (the "iB" is optional, e.g. "K" has the same meaning as "KiB"), or the suffixes KB (=1000), MB (=1000*1000), and so on for GB, TB, PB, EB, ZB and YB.

−a, **−−append**
: Append the output to *file* or to **typescript**, retaining the prior contents.

−c, **−−command** *command*
: Run the *command* rather than an interactive shell. This makes it easy for a script to capture the output of a program that behaves differently when its stdout is not a tty.

−e, **−−return**
: Return the exit code of the child process. Uses the same format as bash termination on signal termination exit code is 128+n. The exit code of the child process is always stored in type script file too.

−f, **−−flush**
: Flush output after each write. This is nice for telecooperation: one person does ‘mkfifo foo; script -f foo’, and another can supervise real-time what is being done using ‘cat foo’.

−−force
: Allow the default output destination, i.e. the typescript file, to be a hard or symbolic link. The command will follow a symbolic link.

−o, **−−output-limit** *size*
: Limit the size of the typescript and timing files to *size* and stop the child process after this size is exceeded. The calculated file size does not include the start and done messages that the **script** command prepends and appends to the child process output. Due to buffering, the resulting output file might be larger than the specified value.

−q, **−−quiet**
: Be quiet (do not write start and done messages to standard output).

−t[*file*], **−−timing**[=*file*]
: Output timing data to standard error, or to *file* when given. This data contains two fields, separated by a space. The first field indicates how much time elapsed since the previous output. The second field indicates how many characters were output this time. This information can be used to replay typescripts with realistic typing and output delays.

−V, **−−version**
: Display version information and exit.

−h, **−−help**
: Display help text and exit.

NOTES

The script ends when the forked shell exits (a *control-D* for the Bourne shell (**sh**(1)), and *exit*, *logout* or *control-d* (if *ignoreeof* is not set) for the C-shell, **csh**(1)).

Certain interactive commands, such as **vi**(1), create garbage in the typescript file. **script** works best with commands that do not manipulate the screen, the results are meant to emulate a hardcopy terminal.

It is not recommended to run **script** in non-interactive shells. The inner shell of **script** is always interactive, and this could lead to unexpected results. If you use **script** in the shell initialization file, you have to avoid entering an infinite loop. You can use for example the **.profile** file, which is read by login shells only:

```
if test -t 0 ; then
   script
   exit
fi
```

You should also avoid use of script in command pipes, as **script** can read more input than you would expect.

ENVIRONMENT

The following environment variable is utilized by **script**:

SHELL

If the variable **SHELL** exists, the shell forked by **script** will be that shell. If **SHELL** is not set, the Bourne shell is assumed. (Most shells set this variable automatically).

SEE ALSO

csh(1) (for the *history* mechanism), **scriptreplay**(1)

HISTORY

The **script** command appeared in 3.0BSD.

BUGS

script places *everything* in the log file, including linefeeds and backspaces. This is not what the naive user expects.

script is primarily designed for interactive terminal sessions. When stdin is not a terminal (for example: **echo foo | script**), then the session can hang, because the interactive shell within the script session misses EOF and **script** has no clue when to close the session. See the **NOTES** section for more information.

AVAILABILITY

The script command is part of the util-linux package and is available from Linux Kernel Archive ⟨https://www.kernel.org/pub/linux/utils/util-linux/⟩.