NAME

unbuffer − unbuffer output

SYNOPSIS

unbuffer *program* [*args*]

INTRODUCTION

unbuffer disables the output buffering that occurs when program output is redirected from non-interactive programs. For example, suppose you are watching the output from a fifo by running it through od and then more.

```
od -c /tmp/fifo | more
```

You will not see anything until a full page of output has been produced.

You can disable this automatic buffering as follows:

```
unbuffer od -c /tmp/fifo | more
```

Normally, unbuffer does not read from stdin. This simplifies use of unbuffer in some situations. To use unbuffer in a pipeline, use the -p flag. Example:

```
process1 | unbuffer -p process2 | process3
```

CAVEATS

unbuffer -p may appear to work incorrectly if a process feeding input to unbuffer exits. Consider:

```
process1 | unbuffer -p process2 | process3
```

If process1 exits, process2 may not yet have finished. It is impossible for unbuffer to know long to wait for process2 and process2 may not ever finish, for example, if it is a filter. For expediency, unbuffer simply exits when it encounters an EOF from either its input or process2.

In order to have a version of unbuffer that worked in all situations, an oracle would be necessary. If you want an application-specific solution, workarounds or hand-coded Expect may be more suitable. For example, the following example shows how to allow grep to finish processing when the cat before it finishes first. Using cat to feed grep would never require unbuffer in real life. It is merely a placeholder for some imaginary process that may or may not finish. Similarly, the final cat at the end of the pipeline is also a placeholder for another process.

```
$ cat /tmp/abcdef.log | grep abc | cat
abcdef
xxxabc defxxx
$ cat /tmp/abcdef.log | unbuffer grep abc | cat
$ (cat /tmp/abcdef.log ; sleep 1) | unbuffer grep abc | cat
abcdef
xxxabc defxxx
$
```

BUGS

The man page is longer than the program.

SEE ALSO

"Exploring Expect: A Tcl-Based Toolkit for Automating Interactive Programs" by Don Libes, O'Reilly and Associates, January 1995.

AUTHOR

Don Libes, National Institute of Standards and Technology