Command Line Clipboard Access and Alternatives

Last modified: August 5, 2021

by baeldung (https://www.baeldung.com/linux/author/baeldung)

Administration (https://www.baeldung.com/linux/category/administration)

1. Introduction

Buffers exist at every level of computing. They store data temporarily for future operations. Clipboards are such buffers. In this article, we're going to explain **ways to use a clipboard from a terminal under Linux**. Towards this end, we'll discuss clipboard implementations and terminals.

The code in the article has been tested on Debian 10.10 (Buster) with GNU Bash 5.0.3. It's POSIX-compliant and should work in any such environment.

2. Clipboards

What distinguishes clipboards from other buffers is their ease of use and ubiquity.

Clipboards are part of every major operating system. They allow data transfer in and between applications. Default (often universal) keyboard shortcuts and mouse gestures ease the process.

2.1. System-Wide Clipboard

The system-wide clipboard (or system clipboard) is accessible across all applications. In particular, the X Window System (https://linux.die.net/man/7/x), via its *X Server* component, provides the system clipboard in Linux. It has three parts (or *selections*): *PRIMARY*, *SECONDARY*, and *CLIPBOARD*. *CLIPBOARD* usually functions like what most of us would call "the clipboard".

Importantly, **the system clipboard isn't part of the Linux kernel**. **The *X Server* provides the standard system clipboard available to applications under Linux**.

2.2. Independent Clipboards

Independent clipboards serve a particular application and don't cross application boundaries. Moreover, an independent clipboard can leverage, override, or ignore the system clipboard contents. **Any application can implement its own separate clipboard mechanism**.

As a result, independent clipboards are especially valuable in pure text-based terminals. We'll discuss them below.

3. Terminals

Terminals are a command-line interface (CLI) — they only accept text input.

A terminal was originally only a screen and keyboard. It had just enough memory to remotely communicate with a computer. The closest thing to classical terminals in Linux is the Linux console. It's implemented entirely in the kernel. Because of this, **the Linux console doesn't support a system-wide clipboard on its own**.

Classic terminals have been superseded by terminal emulators. They can run on any local or remote machine. Terminal emulators can add functionality not present in the terminal they are emulating.

4. System Clipboard CLI Access

Standard CLI system clipboard access is done via *xsel* (https://linux.die.net/man/1/xsel) (X selections) and *xclip* (https://linux.die.net/man/1/xclip) (X clipboard CLI). Both depend on the *X Server* running either remotely or locally. Their basic usage is similar and straightforward:

```
user@baeldung:~$ echo Copied via xsel. | xsel --input
user@baeldung:~$ xsel --output
Copied via xsel.
user@baeldung:~$ echo Copied via xclip. | xclip -in
user@baeldung:~$ xclip -out
Copied via xclip.
```

Trying to use *xsel* or *xclip* without *X Server* running (checked via *xset* (https://linux.die.net/man/1/xset)), produces an error:

```
user@baeldung:~$ xset q
xset: unable to open display ""
user@baeldung:~$ echo Copied via xsel. | xsel --input
xsel: Can't open display: (null)
: Inappropriate ioctl for device
```

Tools like *xsel* and *xclip* are the only choice for using the *X Server* system clipboard in a pure text-based terminal.

On the other hand, terminal emulators like GNOME Terminal Emulator (https://help.gnome.org/users/gnome-terminal/stable/introduction.html.en) and Konsole (https://konsole.kde.org/) provide additional options. Their graphical user interface (GUI) nature includes native mouse and, often, keyboard support. Because of this, users can easily select, store, and restore text via the system clipboard. The functionality also extends to remote terminal emulators (such as Putty (https://www.putty.org/)).

5. Independent Clipboard CLI Access

Even when a system clipboard is unavailable, there are options for clipboard-like functionality.

5.1. General-Purpose Mouse (*gpm (https://linux.die.net/man/8/gpm)*)

It turns out that there's a loophole in the "text input only" rule for terminals. It allows for perhaps the most natural way to store and restore data via the clipboard — the mouse.

The *gpm* software package provides a rudimentary mouse driver to any supported terminal. When running the *gpm* daemon, an inverted highlighter character represents the mouse cursor. It roughly reflects the movements of the mouse along with the character matrix.

```
user@baeldung:~$ echo This is the output of a command.
This is the output of a command.
user@baeldung:~$
```

The *gpm* package has clipboard functionality limited to the currently visible data. The user can select text to directly transfer to the prompt position. The data isn't stored anywhere after this operation.

5.2. Screen Manager (*screen* (https://man7.org/linux/man-pages/man1/screen.1.html)) and Terminal Multiplexer (*tmux* (https://man7.org/linux/man-pages/man1/tmux.1.html))

Terminal input can contain complex code symbols (think terminal codes (https://man7.org/linux/man-pages/man4/console_codes.4.html) and signal (https://www.baeldung.com/linux/sigint-and-other-termination-signals#introduction-to-signals) shortcuts). They can, for example, change the cursor position at a specific place on the screen.

Terminal multiplexers like *screen* and *tmux* use these instructions with key bindings (/linux/tmux#key-bindings). They enable the user to have several virtual consoles from different processes. The virtual consoles are placed in window-like portions of the same terminal screen.

Each virtual console in *screen* and *tmux* provides a copy-and-paste mechanism. The user can scroll through the history of a window, select the text, and store it in a paste buffer. The paste buffer's contents are held in the main terminal multiplexer process. Subsequently, they can be restored in any window.

5.3. Vi IMproved (*vi* (https://linux.die.net/man/1/vi))

The *vi* text editor (/linux/vi-editor) also allows for copy-and-paste operations within the application. For example, from *vi* in normal mode (accessed via *Esc*), the user can:

- *yy* – copy (yank) the current line, including the newline character
- *p* – paste (put) the line(s) in the buffer into the text after the current line

5.4. Custom Implementations

Any storage type can serve as a type of primitive clipboard.

In Bash, for example, we can implement a *clipfile*:

```
user@baeldung:~$ echo Copied via clipfile. > /tmp/clipfile
user@baeldung:~$ cat /tmp/clipfile
Copied via clipfile.
```

The drawbacks, however, include inconvenience, lack of standardization, and limited usefulness.

6. Conclusion

In this article, we explained **different ways to use clipboards in a terminal**. First, we defined what a clipboard is and described two different types of clipboards. We then used tools for system clipboard access via the terminal. Finally, we discussed alternative clipboard implementations.

In summary, terminal clipboard access is possible in multiple ways. Depending on the circumstances, it can even be convenient.

CATEGORIES

ADMINISTRATION (/LINUX/CATEGORY/ADMINISTRATION)
FILES (/LINUX/CATEGORY/FILES)
FILESYSTEMS (/LINUX/CATEGORY/FILESYSTEMS)
INSTALLATION (/LINUX/CATEGORY/INSTALLATION)
NETWORKING (/LINUX/CATEGORY/NETWORKING)
PROCESSES (/LINUX/CATEGORY/PROCESSES)
SCRIPTING (/LINUX/CATEGORY/SCRIPTING)
SEARCH (/LINUX/CATEGORY/SEARCH)
SECURITY (/LINUX/CATEGORY/SECURITY)
WEB (/LINUX/CATEGORY/WEB)

SERIES

LINUX ADMINISTRATION (/LINUX/LINUX-ADMINISTRATION-SERIES)
LINUX FILES (/LINUX/LINUX-FILES-SERIES)
LINUX PROCESSES (/LINUX/LINUX-PROCESSES-GUIDE)

ABOUT

ABOUT BAELDUNG (/ABOUT)
THE FULL ARCHIVE (HTTPS://WWW.BAELDUNG.COM/LINUX/FULL_ARCHIVE)
WRITE FOR BAELDUNG (/LINUX/CONTRIBUTION-GUIDELINES)
EDITORS (HTTPS://WWW.BAELDUNG.COM/EDITORS)

TERMS OF SERVICE (HTTPS://WWW.BAELDUNG.COM/TERMS-OF-SERVICE)
PRIVACY POLICY (HTTPS://WWW.BAELDUNG.COM/PRIVACY-POLICY)
COMPANY INFO (HTTPS://WWW.BAELDUNG.COM/BAELDUNG-COMPANY-INFO)
CONTACT (/CONTACT)