



# Need of Advanced Clipboard Manager

Sarang B. Kimmatkar

Assistant Professor, Department of Computer Technology  
Kavikulguru Institute of Technology and Science  
Ramtek, Nagpur (MS), India  
sarangk067@gmail.com

**Abstract**—Clipboard is a temporary storage or buffer which is used by an Operating System for storing temporary data. Clipboard is also defined as a software facility used for short-term data storage and/or data transfer between documents or applications, via copy and paste operations. It is most commonly a part of a graphical user interface (GUI) environment and is usually implemented as an anonymous, temporary data buffer, sometimes called the paste buffer, that can be accessed from most or all programs within the environment via defined programming interfaces. Any operating system has its own clipboard. The functionality provided by operating system, considering clipboard, are very limited for users. If the system is to be used for operation requiring fast use, say faster development of programs and structures, then the ordinary clipboard will not provide this supportive environment. Hence there is need of some software mechanism, called as clipboard manager, which will increase functionalities of clipboard so that ordinary clipboard can be used for satisfying needs of the faster working environment.

**Keywords**—Clipboard; Clipboard Manager; Buffer of sytem; Cliped data; Paste buffer;

## I. INTRODUCTION

A typical application accesses clipboard functionality by mapping user input (key bindings, menu selections, etc.) to these interfaces. The data content in clipboard is the data which is copied or cut within the OS environment. Data may be text, image, any file or directory which user wants to move from one location to another. In human-computer interaction, cut-paste and copy-paste are related commands that offer a user-interface inter-process communication technique for transferring data. The cut command removes the selected data from its original position, while the copy command creates a duplicate; in both cases the selected data is placed in a clipboard. The data in the clipboard is later inserted in the position where the paste command is issued. Many clipboards provide only one buffer, overwritten by each new copy or cut operation. For example, suppose any user copies an audio file, the information about that audio file is stored in the clipboard. Later, when user copies any video file, clipboard is overwritten with information about video file. So when paste function is invoked, video file is pasted by the system. In case of Linux and Windows operating system, when files are copied within the system, the locations of these files are stored in clipboard so that when required, these files can be fetched from that location. But in case of textual data, when copied, the data is stored in the clipboard which is fetched on invoking paste function.

Computer-based editing can involve very frequent use of cut-and-paste operations. Most software-suppliers provide

several methods for performing such tasks, and this can involve (for example) key combinations, pull-down menus, pop-up menus, or toolbar buttons.

1. The user selects or "highlights" the text or file for moving by some method, typically by dragging over the text or file name with the pointing-device or holding down the Shift key while using the arrow keys to move the text cursor.
2. The user performs a "cut" operation via key combination Ctrl+x, menu, or other means.
3. Visibly, "cut" text immediately disappears from its location. "Cut" files typically change color to indicate that they will be moved.
4. Conceptually, the text has now moved to a location often called the clipboard. The clipboard typically remains invisible. On most systems only one clipboard location exists, hence another cut or copy operation overwrites the previously stored information. Many UNIX text-editors provide multiple clipboard entries, as do some Macintosh programs such as Clipboard Master and Windows clipboard-manager programs such as the one in Microsoft Office.
5. The user selects a location for insertion by some method, typically by clicking at the desired insertion point.

The development of a clipboard organizer/manager begins with proper understanding of copy paste operations[1]. Apperley, Fletcher and Rogers [3] provided an overview on copy-paste cycle. Copy and paste operations are a fundamental part of the graphic style of user interface. Copy and paste are amongst the most commonly used commands and are usually implemented so as to provide quick and convenient use. The only drawback is the fact that the clipboard (paste buffer) is usually invisible. No user can see directly that what data is present in clipboard. If there is some delay between copy and paste, say a need to edit the destination document in readiness for receiving the cutting, then difficulties can arise and it may slow the process. It is often all too easy to inadvertently issue another copy command and overwrite the information waiting in the buffer but buffer is overwrite. Sometimes user needs to know which data is copied in the clipboard during any operation. To overcome this data in the clipboard can be fetched and stored in a permanent storage that can be used when required.

A paste operation takes place which visibly inserts the clipboard text at the insertion point. The paste operation does

not typically destroy the clipboard text, it remains available in the clipboard and the user can insert additional copies at other points.

The Microsoft Office Clipboard allows user to copy up to 24 multiple text and graphical items from Office documents or other programs and paste them into another Office document. For example, a user can copy text from an e-mail message, data from a workbook or datasheet, and a graphic from a presentation and then paste them all into a document. By using the Office Clipboard, anyone can arrange the copied items the way that is required in the document [7]. In Windows operating system, for some applications, such as Microsoft word, the clipboard can be used to copy multiple textual data. If during editing document some textual data is to be used many times by the user after copying some other text, this functionality is useful. Also it provides options for resuming editing if user has not saved the document due to some reason. But these functionalities are not sufficient for the users requiring faster access to clipped data. Some security also might be needed for buffered data by the user point of view. The Linux clipboard is also much like Windows but it doesn't provide such extra options. Hence there is a need of some software which will extend functionalities of clipboard.

## II. NEED OF CLIPBOARD MANAGER

The LINUX clipboard is much like the windows equivalent. The clipboard is a software facility used for short-term data storage and/or data transfer between documents or applications, via copy and paste operations. It is most commonly a part of a GUI environment and is usually implemented as an anonymous, temporary data buffer that can be accessed from most or all programs within the environment via defined programming interfaces. A typical application accesses clipboard functionality by mapping user input (key bindings, menu selections, etc.) to these interfaces. Clipboard of any OS doesn't have ability to maintain history of copied content, to copy multiple contents at different locations at a time or to paste multiple contents at multiple locations. Hence development of a new clipboard manager is needed for many users to support different operations.

Anyone may need clipboard manager for following reasons

### A. See What is Clipped within system

When user copy something, it's hard to tell: nothing visually happens. Information is sent to the clipboard, and the only way user can find out what's there is by pasting it. A clipboard manager gives a quick way to see what is copied along the system. Simple but very useful, if anyone want to quickly review.

### B. Recover From Distractions

The clipboard being fundamental to most computer workflows, sometimes user may accidentally copy over some other text or file losing previously copied data. In this case user has to search along many windows for recovering previously copied data. Clipboard managers keep an ongoing record of things users have copied, allowing anyone to scroll through clipboard history and find back any missing object.

### C. Use as a Highlighter

If user is researching something online, user may require a virtual highlighter. Simply highlight and copy any important bits of information as user reading an article, the clipboard manager will store them all in a buffer. When users have done reading, users can look through everything copied and paste what's still useful into a document.

### D. Review Your Day

Scrolling through the things anyone has copied today is a great way to remember what work is performed within the system. This could remind users to get back to work on a particular project, or just give a way to fill in gaps in records.

### E. Bookmark URLs

If users find themselves with 4,382 tabs open, the system probably not running all that well. Instead of opening all those tabs, consider collecting the URLs and tidbits that might come in handy by copying them. When work is done, head back to the clipboard manager to compile them. After researching user should have a pretty good idea of what's still useful, so scrolling through what is copied and finding back what actually needed. Bookmarking URLs hence become helpful.

A number of independent software vendors have extended the capabilities of local clipboard. One good example is making the local clipboard work with online applications by saving the clipboard data to the online location upon a copy or cut event, making this data available to online applications for pasting. Another example is a case when the software does not support copying from clipboard or does not allow one to paste a text from clipboard. To make storage of clipboard permanent and to see what user have copied, one buffer on hard disk is created. This buffer is used to store the data which is copied by the user when using system. During each copy operation buffer is modified. Tracks of each copy and cut operation are maintained in log file stored in a buffer.

## III. ABOUT CLIPBOARD MANAGER

A clipboard manager is a computer program that adds functionality to an operating system's clipboard. Many clipboards provide only one buffer, overwritten by each new "copy" operation. The main task of a clipboard manager is to store data copied to the clipboard in a way that permits richer use of the data. Most clipboard managers allow the user to keep multiple clipped objects, available for later use. Some keep a clipping history by automatically making a new buffer for each new cut or copy operation. Some applications have an internal copy history feature. Clipboard managers enhance the basic functions of cut, copy, and paste operations. The default Microsoft Windows clipboard manager enables pasting the copied data, even after an application is closed, by maintaining the clip buffer itself. Its copying and pasting operations are very versatile in what they permit to be transferred between applications. A range of cells clipped from an Excel sheet can be pasted as a table into MS Word, or OpenOffice.org. Formatted text clipped from a web page will become cells in an Excel sheet, a table in MS Word, or plain text in Text Edit. Windows does not offer a copy history feature. Users wanting this function use a third-party clipboard manager that replaces the default clipboard. The UNIX desktop environment KDE ships with Klipper. GNOME provides a basic clipboard manager function as part of the gnome-control-center (accessed

via the `gnome-settings-daemon`), that supports the freedesktop.org Clipboard Manager Specification. Clipboard managers enhance the basic functions of cut, copy, and paste operations with one or more of the following features:

1. Multiple buffers and the ability to merge, split, and edit their contents and selecting in which buffer "cut" or "copy" operations should store data.
2. Text, data objects, media content, and URLs can be separately monitored.
3. Saving copied data to long term storage which can be used at any time when required.
4. Indexing or tagging of clipped data so that the data can be identified easily.
5. Searching option may be provided for saved data.

### *Conclusion*

The clipboard manager will extend functionality of default clipboard. It will provide a buffer for storing large amount of clipped data. This buffer will be permanent, easily accessible and manually movable to any user defined storage. A user accessible list of files must be maintained by clipboard manager which stores name of each file which is stored in buffer. Also user can search a file and its location within the buffer. The secured GUI will be provided to user containing additional options like set and retrieve data in clipboard and set password. Many other options can be included that are required for improving any systems performance.

### *References*

- [1] Mark Apperley, Dale Fletcher, and Bill Rogers, "Breaking the Copy/Paste Cycle: The Stretchable Selection Tool," Computer Science Department, Waikato University Hamilton, New Zealand.
- [2] K. T. Stolee, S. Elbaum, and G. Rothermel, "Revealing the Copy and Paste Habits of End Users," IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 2009.
- [3] Olivier Chapis and Nicolas Roussel, "Copy-and-Paste Between Overlapping Windows," INRIA and LRI - Univ. Paris-Sud & CNRS 2INRIA B<sup>at</sup>. 490, F-91405 Orsay, France.
- [4] Sarang B. Kimmatkar, "Collecting clipped data from default clipboard in Linux," IEEE Sponsored 2nd International Conference on Innovations in Information Embedded and Communication Systems ICIECS'15, Karpagam College of Engineering, Coimbatore, Tamil Nadu, India, 19th and 20th March 2015.
- [5] Jeffrey Stylos, Brad A. Myers and Andrew Faulring, "Citrine: Providing Intelligent Copy-and-Paste," Computer Science Department and Human-Computer Interaction Institute Carnegie Mellon University.
- [6] Sarang B. Kimmatkar, "Extending functionalities of default clipboard," (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6), 2014.
- [7] Shaobo Li, Shulin Lv, Xiaohui Jia, and Zhisheng Shao, "Application of Clipboard Monitoring Technology in Graphic and Document Information Security Protection System," Third International Symposium on Intelligent Information Technology and Security Informatics.
- [8] M. Wang and Z. Qui, "Research of Anti-copy and Plagiarism Monitoring System," First International Workshop on Education Technology and Computer Science, IEEE 2009.
- [9] <https://support.office.com/en-us/article/Copy-and-paste-multiple-items-by-using-the-Office-Clipboard-714a72af-1ad4-450f-8708-c2931e73ec8a> [Accessed on dated 14 September 2017], at hrs. 10:13.
- [10] <https://wiki.archlinux.org/index.php/clipboard> [Accessed on dated 13 September 2017], at hrs. 21:30.
- [11] <http://www.webopedia.com/TERM/C/clipboard.html> [Accessed on 14 September 2017], at hrs. 10:30.
- [12] [https://msdn.microsoft.com/en-us/library/windows/desktop/ms649014\(v=vs.85\).aspx#\\_win32\\_Cut\\_and\\_Copy\\_Operations](https://msdn.microsoft.com/en-us/library/windows/desktop/ms649014(v=vs.85).aspx#_win32_Cut_and_Copy_Operations) [Accessed on dated 11 September 2017], at hrs. 17:00.
- [13] [https://msdn.microsoft.com/en-us/library/windows/desktop/ms649014\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms649014(v=vs.85).aspx) [Accessed on dated 11 September 2017], at hrs. 17:10.