TEXT EDITOR

Diksha Kumari (1DS18CS709)

Gautam Kumar (1DS18CS710)

Harshitha BR (1DS18CS712)

Himanshu Sonwani (1DS18CS713)

ABSTRACT

- **Text editors** are the most commonly used software for writing documentation, plain texts and project source codes. Notepad, Word-pad, etc. are some of the popular text editor tools with various features and options.

- During project design, development and implementation, text editors play an important role. Each text editor has unique features and options of its own.

- Some (old text editors) support just plain text while others (new text editors) support advanced features such as adding images, files, links, and many more.

- After opening the text editor software, we can enter data or type using any keyboard. Editing of existing document can also be done by opening a file (that supports the text editor) from the directory. Save and Save as features can be found in the Menu option. Directory should be selected to save the file; if not selected, the file/document will be saved in default directory.

- A simple Text Editor Application using **Tkinter** is built which is a very good beginner project for Tkinter. This text editor GUI will consist of various menu like file and edit, using which all functionalities like saving the file, opening a file, editing, cut and paste can be done. Now for creating this text editor, Python 3 and Tkinter should already be installed in your system.

INTRODUCTION

- A **Text Editor** is used to edit plaintext files.

- Test editors differ from word processors, such as Microsoft word, in that they do not add additional formatting information to document.

- You might write the paper in word, because it contains tools to change fonts, margins and layout, but word by default puts that formatting and layout information directly into the file, which will confuse the compiler.

- If you open a .doc file in a text editor, you will notice that most of the file in formatting codes.

- Text editors however, do not add formatting codes, which makes it easier to compile your code test editor have a features set different from that of a traditional word processing program.

- For example, most won't let you include pictures, or include tables, or double-space your writing, the features of text editors vary from implementation to implementation

Common features of text editors

- **File operations**: All text editors have a means to save files to disk and to open existing files. Usually there is a menu along the top or bottom of the window. Typically clicking the mouse on the "File" item opens a drop-down menu with Save and Open options.

- **Cursor motion:** Usually you can move the cursor around the screen with the arrow keys; sometimes with the Page up and Page down keys. You can also position the cursor with the mouse; you may or may not need to click the mouse to reposition the cursor. Editors with mouse support usually have a scroll bar too. Most editors have keyboard shortcuts for cursor motion.

- **Cut and paste:** All editors let you select or cut text and copy or move it elsewhere. If you have mouse support, you typically select text by highlighting it: hold down the left mouse button and drag the mouse across the text. Usually the highlighted text is automatically put on the clipboard: paste it elsewhere by clicking the middle mouse button.

- **Search and replace:** Any editor lets you search for specified strings of characters. Decent ones let you replace strings either locally or globally and either with or without confirmation. Good editors can also handle Unix "regular expressions". Look for search and replace on the drop-down Edit menu; there are usually keyboard shortcuts too.

- **Buffers and windows:** Most editors support multiple buffers: temporary storage locations for pieces of text. Good editors also allow you to open multiple windows on the screen (useful for working with two different files or two parts of the same file).

- **Customization:** You can change the appearance and operation of many editors. Look for a Preferences or Options menu entry. Some editors will store a "dot file" in your home directory that records your preferences.

❖ Types of Editors

There are generally five types of editors as described below:

1. **Line Editor**: In this, you can only edit one line at a time or an integral number of lines. You cannot have a free-flowing sequence of characters. It will take care of only one line.
   Ex : Teleprinter, edlin, teco

2. **Stream Editor**: In this type of editors, the file is treated as continuous flow or sequence of characters instead of line numbers, which means here you can type paragraphs.
   Ex : Sed editor in UNIX

3. **Screen Editor**: In this type of editors, the user is able to see the cursor on the screen and can make a copy, cut, paste operation easily. It is very easy to use mouse pointer.
   Ex : vi, emacs, Notepad

4. **Word Processor**: Overcoming the limitations of screen editors, it allows one to use some format to insert images, files, videos, use font, size, style features. It majorly focuses on Natural language.

5. **Structure Editor**: Structure editor focuses on programming languages. It provides features to write and edit source code.
   Ex : Netbeans IDE, gEdit.

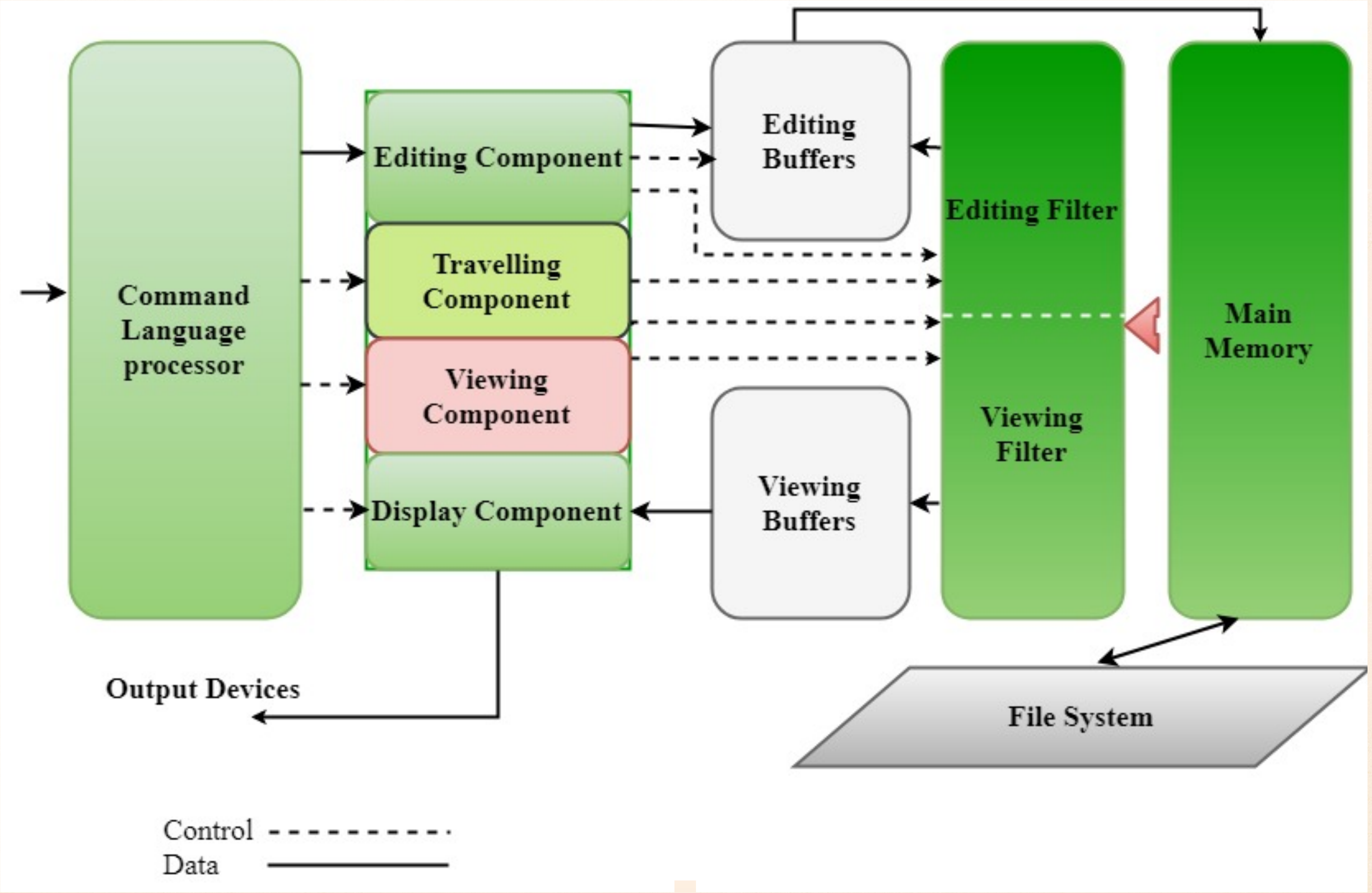
❖ **Tkinter** is a Python Package for creating GUI applications.

- Python has a lot of GUI frameworks, but this is the only framework that's built into the Python standard library.
- It has several strengths; it's cross-platform, so the same code works on Windows, macOS, and Linux.
- It is lightweight and relatively painless to use compared to other frameworks.
- This makes it a compelling choice for building GUI applications in Python, especially for applications where a modern shine is unnecessary, and the top priority is to build something that's functional and cross-platform quickly.

❖ Tkinter Widgets

- There are 19 widgets available in Python Tkinter.
- There are various controls, such as buttons, labels, scrollbars, radio buttons, and text boxes used in a GUI application.
- These little components or controls of Graphical User Interface (GUI) are known as widgets in Tkinter.

BLOCK DIAGRAM



PLAN AHEAD

- We will be using Python and TKINTER library to build our text editor.
- We start by creating a blank window(1200x700) in which we will be writing and editing text.
- Then we will add basic features like : text area, scrollbar, menubar and dropdown menu in which we can open, save and exit our content.
- Also our text editor will be supporting extensions like
    - Text files .txt
    - Python scripts .py
    - JavaScript files .js
    - HTML and CSS files .html, .css
- We will also add status bar which will update the status of our text editor.
- Keyboard shortcuts :- adding keyboard shortcuts to open new file, save a file and save as a new file.

CONCLUSION

- This project which we undertook, will give a big thrust to our technical knowledge as prospective Software professional. It will also be helped us to enhance our skills on the personal front.

- We will try to implement the text editor in a way that it may comprise all the features and functionality of the editor.

- Our text editor can create and revise a document, making sure that there is validity in input as well as functions.

- This full screen text editor is designed to satisfy most of the user requirements such as creating a new file, opening an existing file, saving a file, deleting a file.

- Editing a file is enabled by providing options for inserting or deleting a single character, word or line, as well as support for deleting multiple lines is also present.

SOME REFERENCE LINKS

- https://www.geeksforgeeks.org/editors-types-system-programming/#:~:text=Notepad%2C%20Wordpad%20are%20some%20of,finding%20and%20replacing%2C%20saving%20etc.

- https://www.geeksforgeeks.org/build-a-basic-text-editor-using-tkinter-in-python/

THANK YOU