**TERMINAL BASED – TEXT EDITOR**

**Kohinoor Chatterjee1**

*1DESH, Vishwakarma Institute of Technology, (India)*

**ABSTRACT**

*A text editor is essentially used for editing and formatting text for multi-purpose usage. This text editor which is a Linux terminal-based editor built by extensive use of data structures in C, encapsulates several important features required for handling text. An elegant interface coupled with familiar shortcuts are some of the key features of the editor that ensure a good user experience and set it apart from other text editors.*

***Keywords:*** *C language, Data structures, Linked list, Programming keys, Ncurses, Linux, Terminal*

**1. INTRODUCTION**

*Text editors are one of the key requirements of a programmer. Editors are like the stethoscope of a doctor, programmers carry their own. Hence, the need arises for an editor, that is not only free of any errors or bugs but also suits the programmers needs and wants. Preferences vary from person to person and hence the wide range of editors available in the software world. This editor focusses on these needs of the programmer and guarantees a user-friendly experience to ensure quick learning and long-term usage by the user. Building a terminal based editor is linked with clever usage of macros and code snippets to display items and essentially control the terminal output. Like any software it consists of a front end and back end where front end is handled by the ncurses library and backend by effective C code.*

**2. DESCRIPTION**

The text editor was built by incorporating the ncurses library for Linux terminal manipulation as well as a doubly circular linked list coded by using C language for efficient handling of the large amount of data associated with the editor.

**2.1 DATA STRUCTURE**

The doubly circular linked list is a good choice for the data structure as a linked list has the worst-case complexity for insertion, deletion, search as well as access as Big O(n). This ensures data handling in negligible amount of time eliminating the risk of glitches or hangs during the course of the application. The doubly as well as circular linking feature further ensures easy access to the predecessor as well as successor and the beginning as well as end of the data.

The linked list consists of nodes where each node was used to store a line in the editor. Lines were further linked by using pointers with a head and tail pointer marking the beginning and end of the file. Start and end pointers were also used to mark those lines which are currently visible to the user due to size constraints of the terminal thus helping in scrolling of text in the editor.

**2.2 NCURSES**

Following this came the task of manipulating the output on the terminal. The terminal in the normal mode essentially prints whatever the user types onto the screen. However, in order to build an editor this had to be disabled. Thus, to access the terminal in the RAW mode the ncurses library was used.

Ncurses incorporates several features that were used in the editor. Originally the noecho() as well as raw() feature was used to bring the terminal into the RAW mode. Further, the library also consists of several macros where every key on the keyboard is mapped to a macro. These were utilized by the back end to code each and every key starting from the basic A-Z’s and 0-9’s followed by complicated combinations like backspace, delete, arrow left-right-up-down, shift as well as control keys combined with other basic keys and tab.

**2.3 FEATURES**

The editor has all the features present in normal editors like notepad, vi, nano and gedit as well as some which are exclusive to it. These include –

* Insertion of any character in the beginning, middle and end
* Backspace key functionality
* Deletion key functionality
* Arrow up, down, left and right
* Mouse click for cursor movement
* Terminal resizing on maximizing/ minimizing the terminal
* Scrolling of data
* Reading, editing and saving a file from the device
* Opening a new file followed by saving the existing file
* Saving the file as another file name
* Searching for a word
* Search and Replace functionality
* Select data by shift and arrow keys movement
* Copy and paste data
* Cut and paste data
* Select all data followed by copying/cutting/backspace/delete
* Tab key functionality
* Automatic indentation feature
* Delete a row
* Copy a row
* Move cursor left and right word by word.

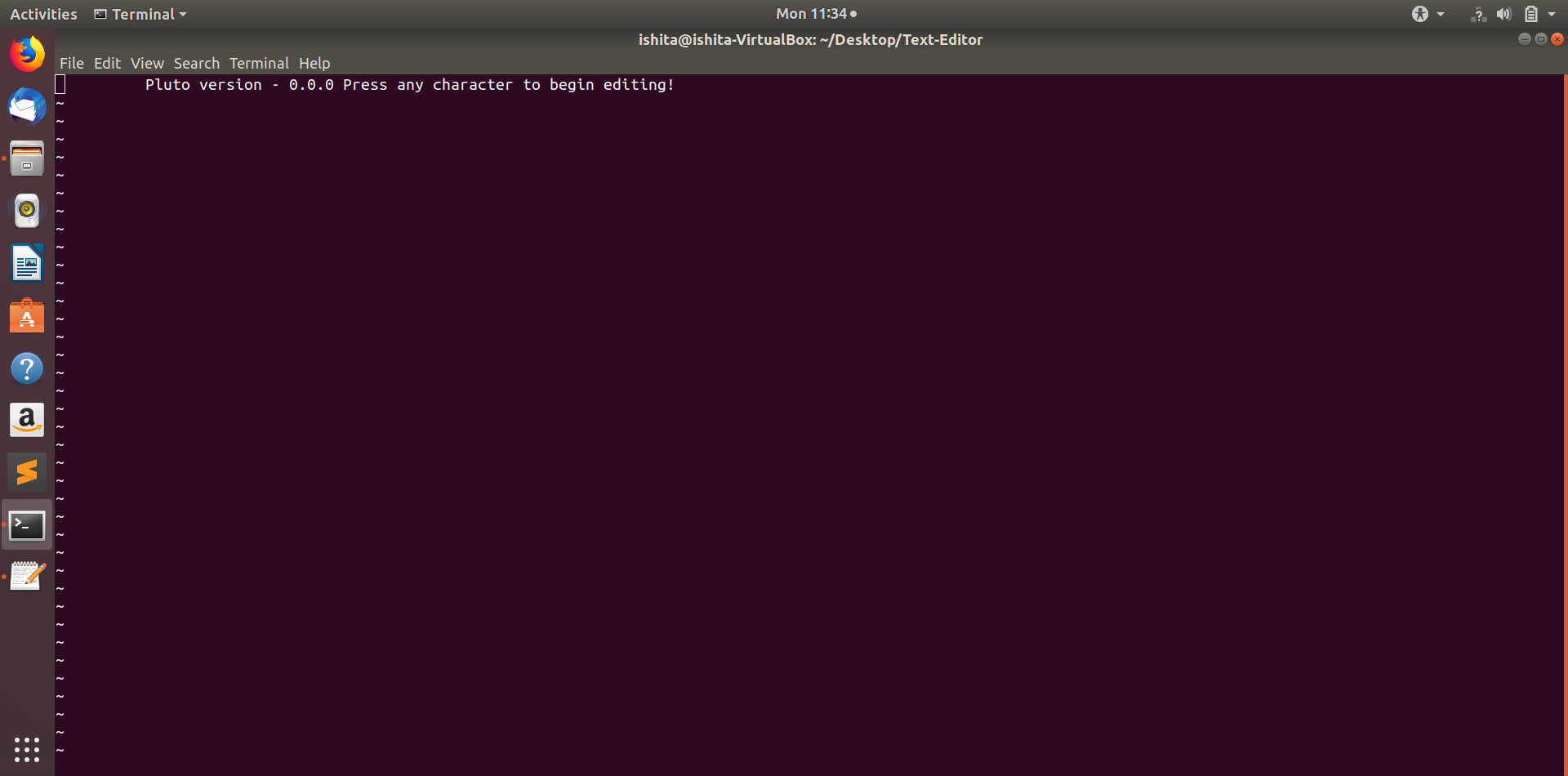
Each of these features were included by manually coding the functionality of these keys in C language.

**3. FIGURES AND TABLES**

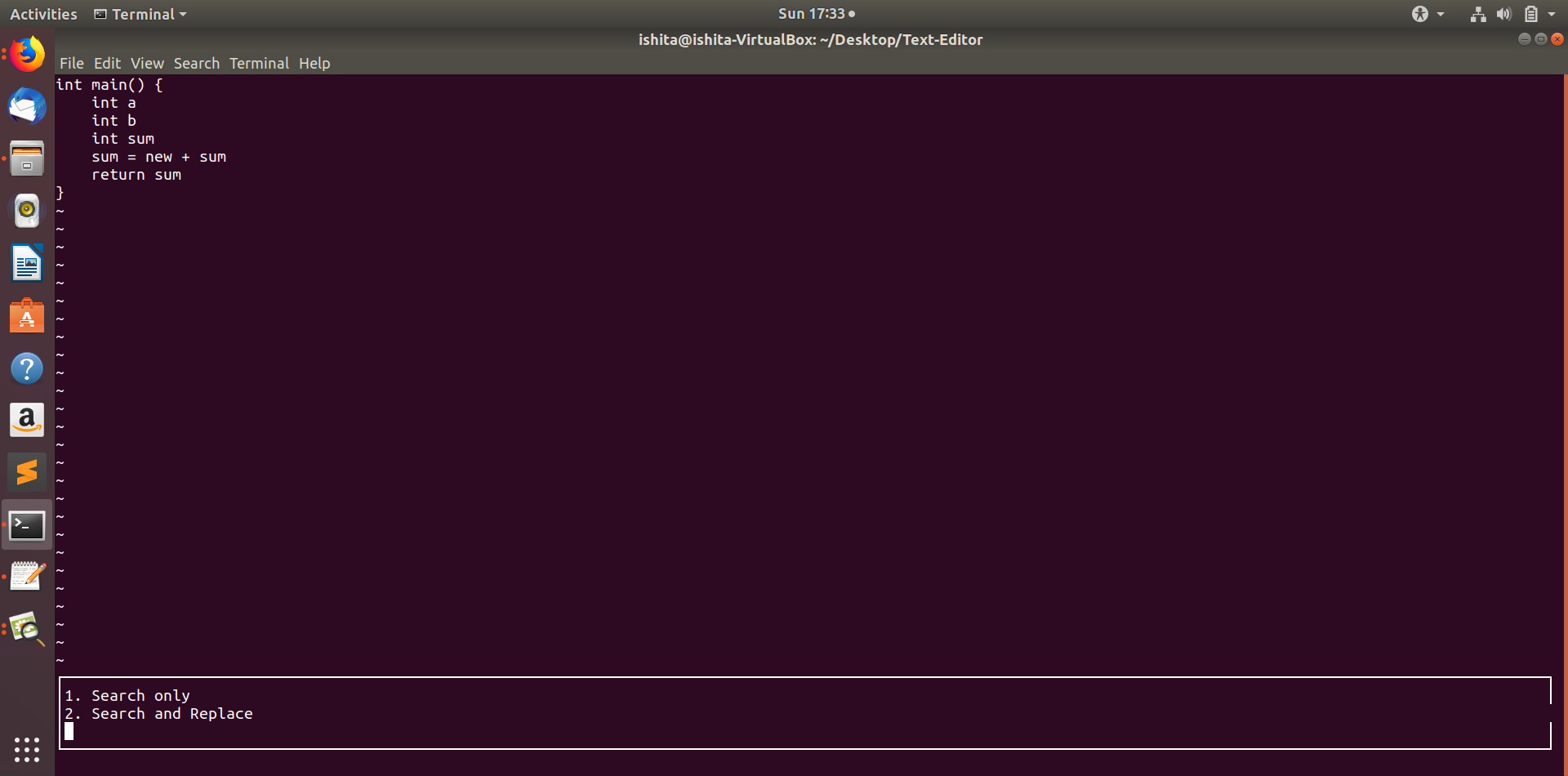
**3.1 USAGE**

|  |  |
| --- | --- |
| Begin Editing | ./pluto file\_name |
| Quit | Ctrl Q |
| Open New File | Ctrl O |
| Save as | Ctrl S |
| Search/Find | Ctrl F |
| Copy | Ctrl C |
| Cut | Ctrl X |
| Paste | Ctrl V |
| Select All | Ctrl A |
| Delete Row | Ctrl D |
| Copy Row | Ctrl Y |
| Move cursor to Next Word | Ctrl R |
| Move cursor to Previous Word | Ctrl L |

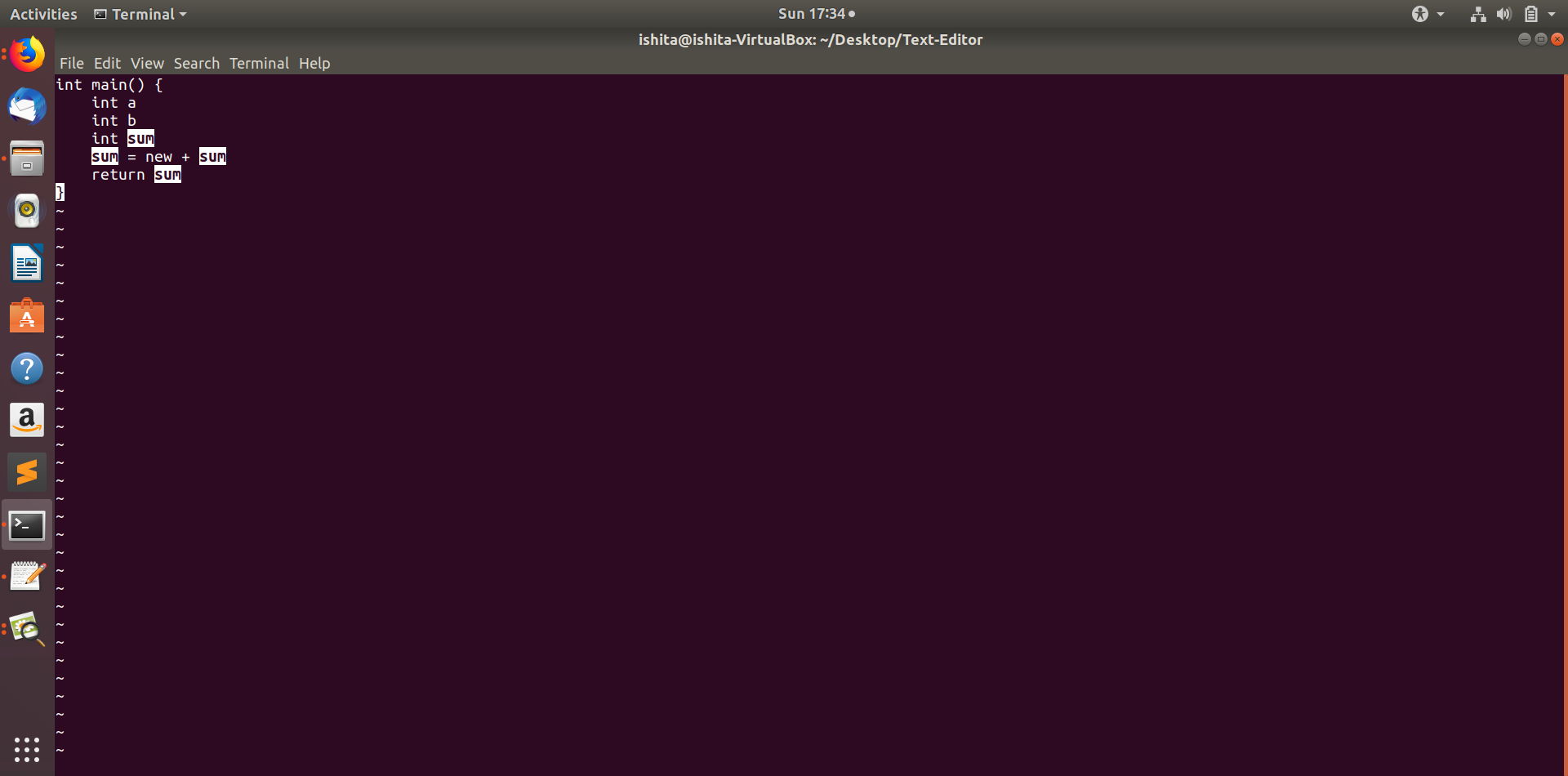
**3.2 SNAPSHOTS**



BASIC LAYOUT



SEARCH/ SEARCH AND REPLACE

SEARCHED ITEM

**4. CONCLUSION**

Text editors, though many due to their usage are highly in demand today. A compact, user-friendly editor is the need of the hour today in order to improve the user’s productivity and provide him a hassle-free experience. Our text editor aims to do the same by incorporating data structures in C and ncurses library together to create a masterpiece.

**REFERENCES**

[1] http://tldp.org/HOWTO/NCURSES-Programming-HOWTO/

[2] http://www.cs.ukzn.ac.za/~hughm/os/notes/ncurses.html