

# THE "LINE EDITOR"

PROJECT REPORT SUBMITTED ON PARTIAL FULFILLMENT OF  
THE REQUIREMENT OF COURSE  
ENGG 102



**Submitted By:**

Ankur Dahal (08)  
Binsan Khadka (19)  
Abhishek Sharma (30)  
Gaurav Banskota (46)  
Sajada Sharma (47)

**Under Supervision Of:**

Mr. Brahmananda Sapkota

Department of Computer Science & Engineering  
School of Engineering  
Kathmandu University  
Dhulikhel, Kavre

June 13, 2000.

## **I. Abstract**

---

This project implements the ideas and the concepts learnt in the course "An Introduction to Structured Programming" offered in the undergraduate course of Computer Engineering in Kathmandu University.

Written in C programming language, the program included in this project deals mainly with the editing of line of text, as per needed of the user. The program is a menu driven and user friendly and is easy to run. User is facilitated with several options for editing and also functions like save, print, open an existing file etc. The user can therefore save the text for future reference as either hard copy or soft copy. The help option aids the user to use the program with comfort and the help function facilitates the short keys for performing different options to edit, save, print etc.

## **II. Acknowledgement**

---

We are grateful to the School Of Engineering, Computer Department for providing us with the project " Line Editor " which is relevant and has a practical application with (C) which helped us to be much more familiar with C language. We are also thankful to Mr. B. N. Sapkota and Mr. Suman Nepal for their valuable suggestions in programming, logic and techniques. We would like to thank Mr. Bhim Prashad Upadhaya for his guidance and co-operation during the lab hours.

Our special thanks to Mrs.Prabhavati Raj Bhandari, Mr. Laxman Gyawali, and Mr Sanat Kumar Bista for their guidance in the preparation and finalization of the report.

## **Table of Contents**

---

I.	Abstract	1
II.	Acknowledgement	2
III.	Introduction	4
IV.	Aims and Objective	5
V.	Limitations	6
VI.	Software Development process	7
VII.	Technical Description	8
VIII.	Program Description	10
IX.	Results/Findings	12
X.	Conclusion	13
XI.	Recommendation	14
XII.	Appendix A (Gantt Chart)	15
XIII.	Appendix B (References/Bibliography)	16

Line Editor, as the name itself suggests, is a user-friendly software, where line can be edited. Several lines of text are written and if there is a mistake in the line, then line editor facilitates the user to make change or edit the line. There are several text editors available but the major difference between a line editor and a text editor is the way lines are edited. Sometimes user needs to edit on an already prepared text, then the line editor provides a way to Edit in it on line by line basis.

After text is edited, it is to be saved for future reference as hard copy or soft copy, in that case Line Editor plays an important role.

## **IV. Aims & Objectives**

---

The main objective of the project is to implement the C programming language that we have studied as our course " COMP 102 " in 1<sup>st</sup> Year 2<sup>nd</sup> Semester of the Bachelor level of Computer Science & Engineering in Kathmandu University.

In the development of computer technology, Internet and e-mail have become more popular. E-mail is an electronic mail sent from one computer to another. The mail to be sent as e-mail has to be composed and to compose a letter, an editor is necessary. This Line Editor provides a way to compose letter, edit it and make ready to send it to concerned person.

The program that was finally submitted has few limitations, which are listed below:

1. Line Editor provides editing of only one line at a time.
2. Number of lines and characters are limited i. e. 60 characters and 20 lines.
3. Special keys that are not in the program may effect the program unknowingly.
4. Use of array (two-dimensional) causes fixed memory to be allocated during program execution.
5. Each file saved in disk uses same amount of disk space( i.e. 1200 Byte).

## **VI. Software Development Process**

---

The objective of Engineering project of COMP 102 is to develop software. To develop software, a detail study is very much essential. Generally the steps to be followed during software development are:

1. Problem definition
2. System analysis
3. System design
4. Program analysis
5. Program preparation
6. Software implementation & maintenance.

Considering the problem, creating a line editor means to built a stand alone program that waits for the user to type the text and whenever required, the program should facilitate the user to make changes or edit on the typed text. For this problem each time a key pressed, it has to be recorded in the memory and if the pressed key is a special key which is required to be edited, then the editing option should be performed. Otherwise if other characters or text is pressed then that text should be added up.

## VII. Technical Description

---

Since the program was implemented on C programming language, most of the functions used are built in functions of C, but there are also some user-defined functions.

The use of data files is found in case of file system. FILE is handled by using fopen and fclose. Fopen and fclose are the built in functions which opens or closes the specified file to read or write in it. The most important and characteristic feature of this software is that it uses the keyboard interface function the bioskey for the interlace between keyboard and monitor. Whichever key is pressed , its hex value is stored in a variable "Key".

There are built in functions gettext( ) and puttext( ). gettext( ) helps to save the characters of specified window along with its attributes in a variable "buffer". Same thing can be put on the screen after some midification in the screen by using puttext( ). Puttext( ) puts the contents of buffer on the specified screen location.

The several user-defined functions and their purpose are listed below:

1. void DrawBox( ) :- draws the outline box of editor screen.
2. void DrawBox1( ) :- draws the box to display help.
3. void Del( ) :- deletes one character on current cursor position.
4. void MenuOption( ) :- display the option of main menu.
5. void FileOption( ) :- display the option of file menu.
6. void EditOption( ) :- displays the option of edit menu.
7. void Highlight(int i, int j )  
:- highlights the j<sup>th</sup> option of i<sup>th</sup> menu where

<u>i:</u>	<u>j:</u>	<u>option:</u>
1	1	File
1	2	Edit
1	3	Tools
1	4	Help
1	5	Exit
2	1	New
2	2	Save
2	3	Load
2	4	Print
2	5	Return

3	1	Cut
3	2	Copy
3	3	Paste
3	4	Delete
3	5	Return

8. void ClearDialog( ) :- clears the dialog box of screen.
9. void MenuSystem( int Choice, int Key) :- performs the functions of main menu where the choice is the choice of function to be performed and Key is for it's confirmation.
10. void New( ) :- sets a new environment for line editor.
11. void Save( ) :- saves the text on specified file on current path.
12. void Load( ) :- loads an existing file from current path.
13. void Print( ) :- sends text to printer for printing.
14. int FileSystem(int Choice1, int Key) :- performs the function of file menu where Choice1 is choice of function to be performed and Key is for it's confirmation.
15. void Cut( ) :- cuts selected line or character from the text.
16. void Copy( ) :- copies selected line or character from the text.
17. void Paste( ) :- inserts the part of text that is recently cut or copied.
18. void Delete( ) :- deletes the selected line or character from text.
19. void EditSystem( int Choice1, int Key) :- performs the function of edit menu where Choice1 is choice of function to be performed and Key is for it's confirmation.
20. void toolssystem( ) :- performs the function of tool menu.
21. void HelpSystem( ) :- displays the help option.

## VIII. Program Description

---

Any editor should have facilities to take the cursor to a desired position, efficiently. To this effect, "Line Editor" provides a number of cursor movement commands like Top of Screen, Beginning of Line, End of Line, Bottom of Screen, Word Left, Word Right.

These movements of cursor are based on the key pressed by the user. If the key pressed is a special key like End, Home, Page Up, Page Down then, the performed operation is the cursor movement to End of current line, Top of current line, Top of screen, Bottom of screen etc. All the cursor movement is handled in the main program itself.

This line editor lets the user cut a word or a line and the contents of text in editor are re adjustable. For example, if line of line numbers is to be cut then the line is saved in a variable CutBuf, then the line is deleted and all the contents of text below line re-shifted above. The logic used to re-shift is:

```
For (j=CurRow; j<RowMax; j++)  
    For (i=0; I<60; I++)  
        Text[j][i] = text[j+1][i] ;
```

where Cursor is positioned in line number equal to value of variable "CurRow " (Cursor Location in Row).

Instead of cut, if delete is chosen then the logic is same but the deleted line is not stored on CutBuf and previously deleted line cannot be pasted.

To paste the line that was previously cut, the saved line from CutBuf is just inserted on the editor.

The program finally prepared is a menu driven program and also can be used by directly utilizing shortcut keys from edit screen. Altogether there are five options in the menu of the program. They are: File, Edit, Tools, Help, Exit.

Each options are discussed sequentially in next page.

## **1. File**

Whenever a document, letter is prepared, it should be converted into hard copy ( print ) or soft copy ( save ). Each document or letter has to be treated as a file. In this file menu there are options to save, open ( existing file ), or print.

For the selection of save or open the file it asks the valid file name and saves or opens the file name, whatever is selected. For the selection of print option, the program sends the file to the printer for a hard copy.

## **2. Edit**

This is the main part of our project which deals with the editing of lines. There are options like cut, copy, paste, delete line, delete to end of line, delete character etc.

After the selection of cut option, we can select the part of the text to be cut from the current position and it can be pasted to any other position by selecting paste option.

If delete character or line option is selected, the selected character or line is deleted. In fact, the character or line to be deleted is shifted to the place of the initial position of character or the line that is to be deleted.

On the selection of insert line, a blank line is seen on the line specified and the remaining line is shifted down.

## **3. Help**

Help facilitates the user with the methods to use the program efficiently. This help option also provides the user with shortcut keys (combination of keys) that are pressed to directly perform the operation.

Finally, if the user wants to shut the session of the line editor, he can exit by selecting Exit menu.

This line editor has been made more efficient by providing the shortcut key facility. This facilitates user to directly perform options for editing without using menu bar.

## **IX. Results/ Findings**

---

The outcome of our project at the end has been excellent. Whatever was hoped in the beginning has been accomplished.

There were two main objectives of doing the project. One of them was to implement the C programming language that was studied in the course COMP102. Since a program more than 30 pages long has been developed, it integrates every possible feature of the C language. The first objective has been fulfilled. The second objective of taking up the project was to build a better team workmanship among the members of the project. While working on the project, the work was divided the work carefully so that everyone was doing something and thus contributing to the project. During the group meetings findings and the proceedings were discussed because of which the various problems encountered were solved with collective decision.

Thus both of the objectives of undertaking the project has been successfully completed.

## **X. Conclusion**

---

In the given period of four months the line editor has been successfully completed. This would not have been possible without the concern and cooperation of all the group members.

Whatever had been thought in the beginning of this project has been accomplished. This project has helped us to implement what had been taught in the course COMP 102 and has also developed our practical approach in C.

## **XI. Recommendation**

---

Before the development of the advanced **TEXT EDITOR**, a programmer would be better equipped if he/she is familiar with **LINE EDITOR**. Additional functions can be put into **THE LINE EDITOR** to make it more user friendly, efficient and advanced.

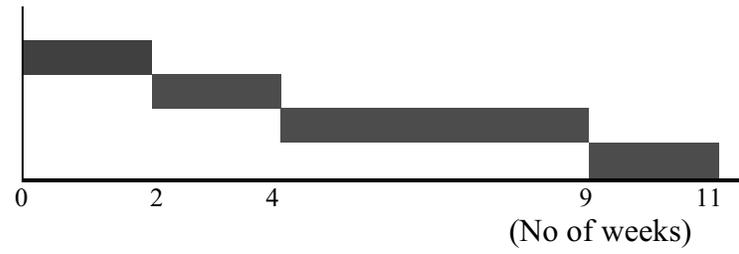
An alternate approach can be used for line editor. Instead of storing each character entered from the input device in an array, the characters entered can be directly saved on specified file, if only the user wants to save it, thus consuming less memory space.

Here, the only input device used is the **KEYBOARD**. The use of mouse can be encouraged for the selection of menu option.

Besides these various other methods can be used to constantly update the line editor to make it easy for the user to use.

**[ GANTT CHART ]**

- 1. Research
- 2. Program Design
- 3. Program implementation & testing
- 4. Report preparation & presentation



### **{ References/ Bibliography}**

1. Gottfried Byron (1998), Programming with C, Preparing & Running Complete C Program, Pp. 101-104, Tata McGraw Hill Edition.
2. Bootle Stan Kelly (1993), Mastering Turbo C, File I/O: Full Stream Ahead, Pp. 258-278, BPB Publications, New Delhi.
3. Kanethkar Yashwant (1995), Writing TSR's Through C, A Small TSR Project, Pp. 221-261, BPB Publications, New Delhi.
4. Khanal R. C. (1996), Computer Concept (part I), Computer Professionals Pp. 170-177, Ekta Publication, Kathmandu.