

COMPUTER ORGANIZATION & ASSEMBLY LANGUAGE

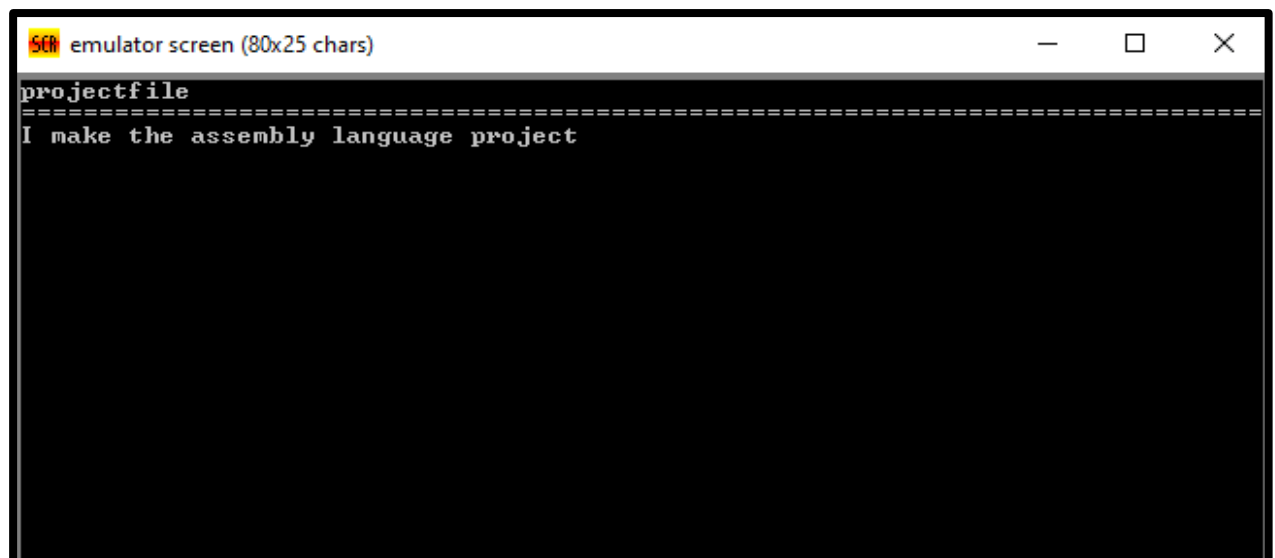
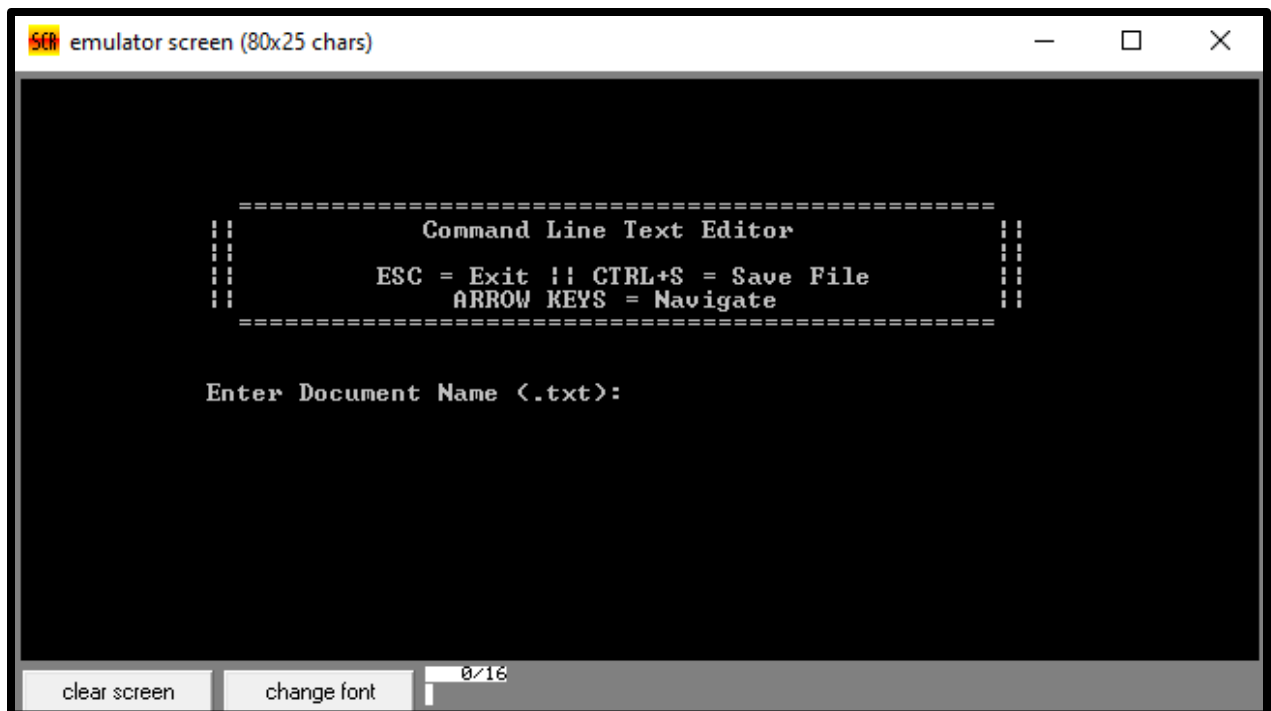
Project:

Command Line Text Editor

Roll No:

23BSAI048

The functionality and flow of a **Command Line Text Editor** implemented in 8086 Assembly Language. The editor allows users to create, read, write, open, and save text files using keyboard input. It operates in a basic command-line interface (CLI) and is built to support standard text manipulation features with custom keyboard inputs like arrow keys, Enter, Backspace, Ctrl+S (Save), Ctrl+O (Open), and Esc (Exit).



Project Features

1. Text Creation:

- Users can input and display characters.
- Text input is handled by scanning for key presses using BIOS interrupts, which are interpreted and displayed accordingly on the screen.

2. Cursor Movement:

- Arrow keys move the cursor up, down, left, and right. The cursor's position is adjusted to follow the user's navigation, and it updates the screen based on the current text location.

3. Text Editing:

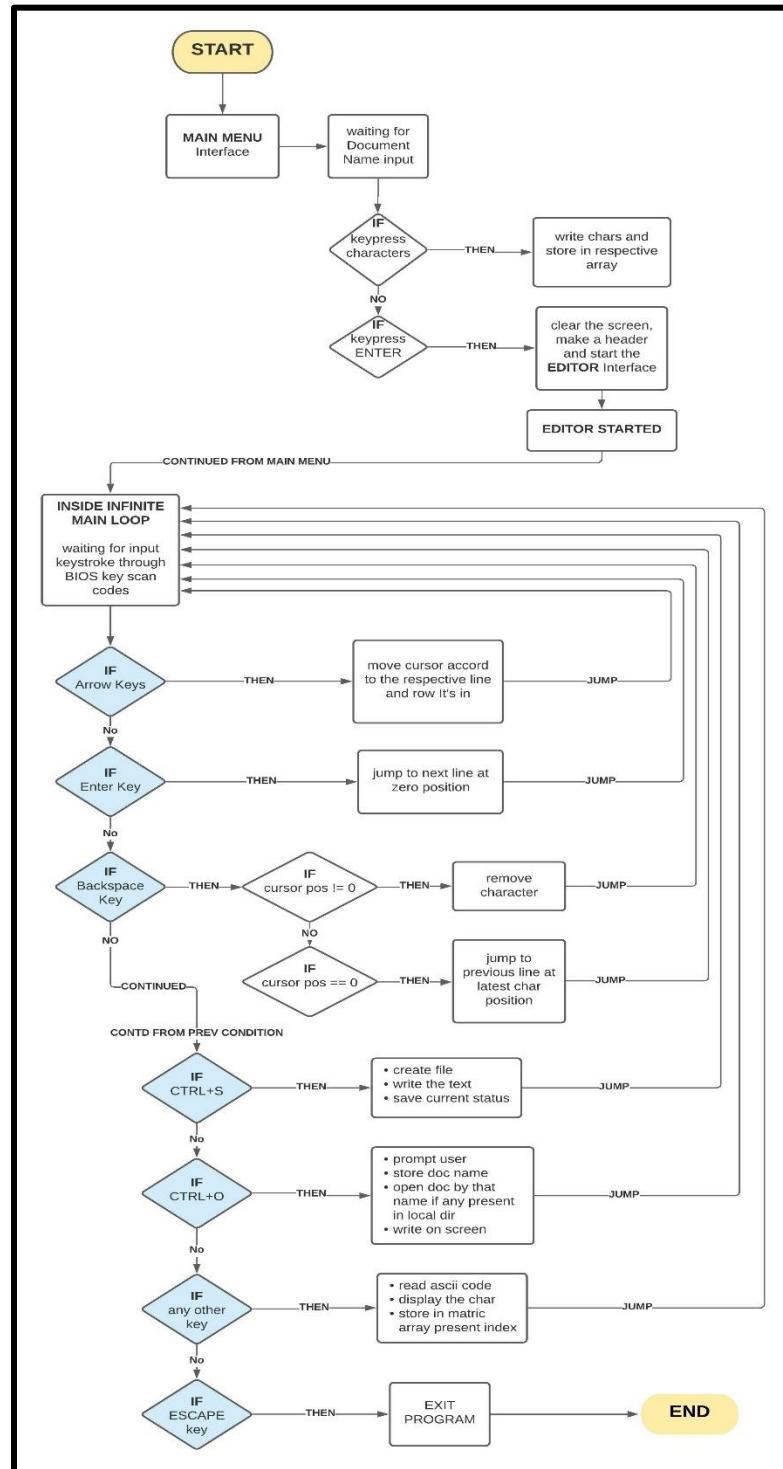
- The Backspace key allows for character deletion, adjusting the line and cursor position.
- Enter key moves the cursor to the next line, starting from position 0 on the new line.

4. File Operations:

- **Ctrl+S**: Save the text to a file, prompting the user to enter a file name.
- **Ctrl+O**: Open a file by name, displaying its contents if the file exists in the directory.
- **Esc**: Exits the editor and terminates the program gracefully.

Flow Diagram

The text editor's flow, from startup through main menu operations to continuous text input and file management.



Detailed Flow:

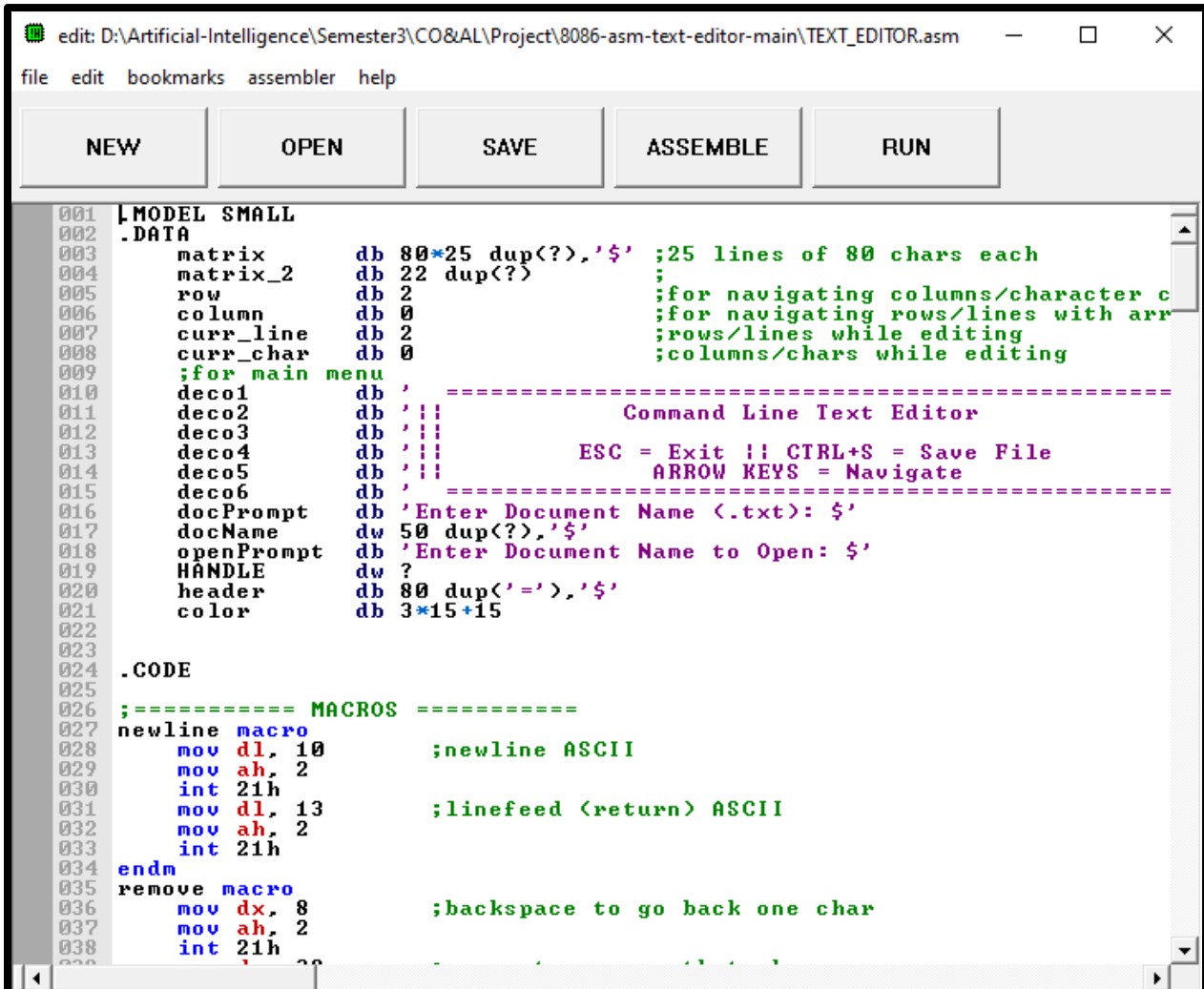
1. Main Menu Interface:

- Starts by asking the user to input a document name.
- Character inputs are stored, and when Enter is pressed, the editor starts with a header display and clears the screen.

2. Infinite Main Loop:

- Continuously scans for keyboard input using BIOS scan codes.
- The loop checks for different key presses:
 - **Arrow Keys:** Move the cursor.
 - **Enter Key:** Move to the next line, positioning the cursor at the start of the new line.
 - **Backspace Key:** Removes the character to the left of the cursor or moves to the previous line if at position 0.
 - **Ctrl+S:** Prompts for a file name and saves the current text.
 - **Ctrl+O:** Prompts for a file name and opens the respective file if it exists.
 - **Any other key:** Displays the ASCII equivalent and stores it in memory.
 - **Esc Key:** Exits the program

Code Segment: This section refers to the specific blocks of the 8086 Assembly code that handle the keypress events and file management.



The screenshot shows a window titled "edit: D:\Artificial-Intelligence\Semester3\CO&AL\Project\8086-asm-text-editor-main\TEXT_EDITOR.asm". The window has a menu bar with "file", "edit", "bookmarks", "assembler", and "help". Below the menu bar are five buttons: "NEW", "OPEN", "SAVE", "ASSEMBLE", and "RUN". The main area displays assembly code with line numbers on the left. The code is organized into sections: `.MODEL SMALL`, `.DATA`, and `.CODE`. The `.DATA` section defines variables like `matrix`, `matrix_2`, `row`, `column`, `curr_line`, `curr_char`, and prompts for document name and open file. The `.CODE` section contains macros for `newline` and `remove`. Comments in green provide explanations for the code.

```
001  [MODEL SMALL
002  .DATA
003      matrix      db 80*25 dup(?),'$' ;25 lines of 80 chars each
004      matrix_2    db 22 dup(?)
005      row         db 2 ;for navigating columns/character c
006      column      db 0 ;for navigating rows/lines with arr
007      curr_line   db 2 ;rows/lines while editing
008      curr_char   db 0 ;columns/chars while editing
009      ;for main menu
010      deco1       db '=====
011      deco2       db '||| Command Line Text Editor
012      deco3       db '|||
013      deco4       db '||| ESC = Exit !! CTRL+S = Save File
014      deco5       db '||| ARROW KEYS = Navigate
015      deco6       db '=====
016      docPrompt   db 'Enter Document Name (.txt): $'
017      docName     dw 50 dup(?),'$'
018      openPrompt  db 'Enter Document Name to Open: $'
019      HANDLE      dw ?
020      header      db 80 dup(' '),'$'
021      color       db 3*15+15
022
023
024  .CODE
025
026  ;===== MACROS =====
027  newline macro
028      mov dl, 10 ;newline ASCII
029      mov ah, 2
030      int 21h
031      mov dl, 13 ;linefeed <return> ASCII
032      mov ah, 2
033      int 21h
034  endm
035  remove macro
036      mov dx, 8 ;backspace to go back one char
037      mov ah, 2
038      int 21h
```