The design of the C mobile earth station simulator based on C language

Jun Li

1 Navigation Department, Qingdao Ocean Shipping Mariners College, Qingdao, China No.84, Jiangxi Road, Qingdao City, China

*email: li.jun8@coscoship.com.cn

Abstract. The communication equipment software of the Global Maritime Distress and Safety System is based on DOS platform, and the equipment operation simulator used for training and assessment is generally developed by VC, VB and FLASH technology. There are a lot of difference between the simulator and the real machine on the interface and the operating response. The operation of the simulator is done with mouse, while the operation of the real machine must be done with the keyboard only on board ships. The Inmarsat-C mobile earth station simulator has being developed by C language. The simulator terminal operating interface and response is identical with the real machine operation because the developing language of the simulator is the same as the real machine. Because the adoption of the doubly-linked list data structure, program structure is clear and code is simplified.

1. Introduction

Inmarsat-C Mobile Earth Station (MES) is one of the radio installations on board ships that required by the International Maritime Organization (IMO). It is widely used for route communications and emergency communications by ocean going vessels. In The Manila Amendments to the International Convention Standards of Training, Certification and Watch keeping for Seafarers, 1978(STCW) [1], the proper use of the Global Maritime Distress and Safety System (GMDSS) radio equipment and the navigation instrument is mentioned a few times. Scientific and efficient training and assessment of the GMDSS radio equipment and navigation instruments for the students is very important, because it is concerned with the safety of ship’s navigation. The software interface and operation mode of the MES is different from those developed on WINDOWS platform because it is programmed by Turbo-C language on the DOS platform. The Mouse is not provided with the MES so it can be operated only with the keyboard.

At present, the equipment operation simulator used for training and assessment is generally developed by VC, VB and FLASH [2] technology. There’s a lot of difference between the simulator and the real machine on the interface and the operating response. The operation of the simulator is done with mouse, while the operation of the real machine must be done with the keyboard only. The Inmarsat-C MES simulator has three parts, the host operating interface, the assessment system and the terminal interface. Software development of the simulator terminal interface adopts C, which is the same language used by the real machine, so the operating interface and response is identical with the real machine operation. The other two parts of the simulator are developed by C# [3], which improved the development efficiency and reduced the development cost.
2. The software components of the Inmarsat-C MES simulator

The Inmarsat-C MES simulator system is a LAN (Local Area Network) composed by a number of computers and the network model is C/S (server/client) [4] mode. The advantages of both client and server hardware environment are made full use, reasonable roles are assigned to the two ends, so reduce the costs for the system communication. The server of the Inmarsat-C MES simulator system is the network management terminal. Its main function includes the client's login, editing tutorial exercises, posting evaluation questions and exam management, etc. The client of the Inmarsat-C MES simulator system is the training and testing terminal for students. An interface is provided for every operator to participate in the training and assessment examination. The software development platform of the system is C# and Turbo-C. The Data initialization, training and the assessment can be completed by the simulator system.

3. The implementation of the Inmarsat-C simulator system

3.1. The implementation of Inmarsat-C MES terminal software

The main functions of the simulator include telex text edit and handle, address book, telex transmission, E-mail, X.25 protocol, position reporting, Enhanced Group Calling(EGC) setting, Global Position System(GPS) interface control, distress setting and distress alerting, etc. The user interface of terminal shows in figure 1. Because the Technique of Direct Screen Writing, digital timer, transparent window shade and doubly-linked list data structure, the display speed is significantly accelerated and the reliability and stability of the program is improved.

3.2. The Technique of Direct Screen Writing.

The Technique of direct screen writing is to direct access to the display cache to obtain screen display. The CPU busy time is saved when display characters on screen using the technique of direct screen writing instead of using the functions of the C program language, so the display speed on the screen is accelerated obviously. The display speed is largely depends on the speed of graphics card. Compared to the interrupt calls system of the DOS system (including system calls, DOS calls and BIOS calls) it has not only the advantages of simplicity and rapidity and can be applied to the interrupt service routine programming, such as the interrupt and the “terminate and stay resident” (TSR) program. Because of the Irreaccessibility of DOS in the DOS5.0 version or early, the system calls of DOS and some BIOS calls cannot be called in the interrupt service routine programs. So it is convenience to display characters on the screen by means of the technique of direct screen writing.

3.3. Display cache. The display buffer is used to record the screen display information.

The screen is divided into 25 lines and 80 characters per line in the text display mode, so one screen can display 2000 (80 * 25) characters. Each character in the display buffer area take two bytes, one byte is the character's ASCII code, the other is the character of the display properties, the color display, so a total of 4000 byte buffer is taken by 2000 characters. They correspond to the characters of every row and column of the screen from left to right and top to bottom. The font information is recalled by the ASCII code and the video hardware maps the dot matrix to the screen. The characters displayed on the screen will be changed if the ASCII code in the cache is changed.

The character color, the foreground color, is depended on the lower four bit (0-3 bit) of the characters display properties. The background color is depended on the 4-6 bit and the character blinking properties is determined by the highest bit (bit 7).)

3.3.1. Display method.

The MK-FP ( ) function of Turbo C generates a far Pointer directing to a specified location. The display buffer address of the color monitor is 0B800H in the “Text display mode”. To create a pointer directing to the video buffer using the char far * p = MK_FP (0xb800, Ox0000) and the pointer directs to the first character on the upper left corner of the screen. The corresponding characters will be
displayed on the screen as long as the attributes of the characters and bytes in the memory directed by the pointer are written.

3.3.2. The applications of the technique of direct screen writing in the Inmarsat-C MES simulator.

The transparent window shade:
The Menus, submenus and dialog boxes of the Inmarsat-C MES simulator are rectangular window and the active window has a transparent shadow. The characters covered by the shade is uncertain, it may be a menu or message in the text editor. See figure 1. Because in the C language there isn’t a function of changing the character color in specified location on the screen so we can use the display buffer characteristics to change the color attributes of characters on the specified areas in the display buffer. The ASCII value of character remains the same, the color properties of the foreground color is changed to dark gray and the background color is changed to black, then the effect of a transparent shadow is realized. The function named shadow can be called by all windows in the project. The main codes to do this are shown below.

```c
void shadow(int x, int y, int width, int height, int on)   //window shadow function, on=0.
{
    //eliminating the shadow; on=1, painting shadows
    char far *p;
    char far *q;
    int value, i;
    if (on==0) value=31;
    else value=8;
    q=MK_FP(0xb800,0x0000);
    p=q+80*2*height+2*x+1;
    if (height<25) for (i=0; i<(width-x)+1; i++)
    {
        *p=value;
        *p++;
    }
    p+=160;
    *p=value;
    p=q+80*2*(y-1)+2*width+3;
    if (width<79) for (i=0; i<(height-y); i++)
    {
        *p++;
        *p+=value;
    }
}
```

Figure 1. The terminal interface of the Inmarsat-C MES simulator.
In the function “shadow”, the parameter “x” and “y” represents the columns and rows of the upper-left corner of the window respectively, the parameter “width” and “height” represents the columns and rows of the lower-right corner of the window. When the parameter “on” is 1, the character’s color attribute is that the foreground color is dark gray and the background one is black, then the shadow is painted. When the parameter “on” is 0, the character’s color attribute is that the foreground color is white and the background color is blue, that is to eliminate the shadow and restore the original screen. Because the foreground color and the background color is mostly white and blue respectively in the Inmarsat-C MES simulator program, the original screen can be restored when the parameter “on” is 1. The other characters’ original color can be restored by using the function gettext( ) and puttext( ).

In the function shadow, q=MK_FP(0xb800,0x0000). The first address of the display buffer is assigned to the parameter “q”. The location in the buffer of the shadow characters can be determined because the offset of the distance between the characters to be changed to “q” be calculated according to the values of the parameters of (x, y) and (width, height). The color value is dark gray 8 when eliminating the shadow and the value is white 31 when painting the shadow.

The direct screen writing in telex editor.
The main codes to do this are shown below.
```c
void PrintAt(char*Text, int x, int y, int bc, int fc)
{
    int Color=fc+bc*16;
    char far*Addr=&screen[(x+y*80)*2];
    while (*Text)
    {
        *(Addr++)=*(Text++);
        *(Addr++)=Color;
    }
}
```

3.4. The data structure and algorithm of the text editor of the Inmarsat-C MES simulator
The text editor of the Inmarsat-C MES simulator is mainly consisted of some functional units, they are respectively the file operation unit, the text edit unit, cut and paste unit, the menu and other control unit and help unit, etc. The function of the text editor unit is introduced as follow.

The mainly function of the text editor unit are: insert, delete, backspace, cut, find, replace, copy, paste, insert system time, insert position information, delete the word where the cursor is positioned, and delete line, etc. see figure 2.

![Figure 2. The text editor unit of the Inmarsat-C MES simulator.](image-url)
The text editor is the most commonly used tools to create and edit documents. Recently, with the development of computer and computational science, many kinds of text editor are brought out. It has been a common control in the advanced programming language development platforms, such as VC, VB, C#, etc. The realization of the required program function can be achieved conveniently by changing the properties of the controls. In the C language development environment, programmers have to write the code to obtain the function of the text editor unit. So we develop a text editor unit for the Inmarsat-C MES simulator which has less memory footprint and run more efficient.

The doubly-linked list data structure [5] is adopted. The data domain of each node is a line of characters. There are 70 characters in a line. The last one is “\0” which marks the end of the string.

typedef struct Line
   {   char lineString[70];
   int charNumbers;  //the number of characters in the line (including the ending sign of the string “\0”).
   struct Line *next;
   struct Line *prior;
}                               //Line_link;
Using two lines of data as an example, a schematic diagram for doubly-linked list data structure is illustrated in figure 3.

In Figure 3, the data represents the two domains, “char lineString[70]” domain and charNumbers domain. The prior domain in the first line and the next domain in the last line are all NULL.

The doubly-linked list data structure has many advantages, such as simple structure and algorithm, data in every line using one dimensional array, saving storage compared with pointer, obtaining scrolling conveniently, etc. Because the costs of the charNumber domain’s two bytes reduce the complexity of the algorithm significantly, unnecessary function calls will be avoided.

The code realizing the functions of the text editor unit is a singly loop, described as follows.

```c
{    window(2,6,71,22); // The editing area.
    clrscr();
    q=p;
    for (i=y-1;i>0;i--)
    {  if (q->prior!=NULL)
       q=q->prior;
    }
    i=1;
    while ((q!=NULL)\&\&(i++<=17))
    { gotoxy(1,i);
      cprintf(q->lineString);
      q=q->next;
```

The code realizing the functions of the text editor unit is a singly loop, described as follows.
The following statements defines the main variable for the text editor unit of Inmarsat-C simulator:

```c
Line_link *p;   //The pointer of the node which located in the current line of the link list in the editing area.
int x, y;    // The cursor position in the current editing window.
int chars;    // The total number of characters in the editing area.
```

Scrolling operation is supported in the text editor unit of Inmarsat-C MES simulator. In the editing area, the first line in the editing area may not be the first line of the text which is being edited. The first line of the editing area will be located easily according to the number of line, expressed in terms of y, which the cursor is currently positioned, and the doubly-linked list data structure of the text editor unit.

The rational design of the doubly-linked list data structure can achieve compact and effective algorithm. For example, to add a character to the text. In the functions of add or insert characters, firstly, it is needed to locate the position where the cursor is positioned by the check(Hhead, wherey(), wherex()) function if the singly-linked list is adopted; secondly, to apply the space of pointer node [6] for the character, and finally assign a value to the pointer node. All these lead to increase system overhead. For this simple example in the text editor of the Inmarsat-C MES simulator, it’s fine to use p-> lineString[x-1] to assign a value. Although the characters located after the cursor should be moved to insert or delete characters and raise some costs, in general, the benefits of reasonable date structure far outweigh its costs.

4. Conclusions
The development of this simulator adopts different platform according to different requirements. Because the developing language of the simulator is the same as the real machine, the simulator terminal operating interface and response is more effective and authentic. These is little difference between the real machine operations on board the ship and on the simulator. Software development of the host operating interface adopts C#, which improved the development efficiency and reduced the development cost.

References
[1] Li, W., (2013) A study on practice teaching concerning navigation major in vocational colleges based on manila amendments to the stcw convention. Journal of Nantong Vocational & Technical Shipping College, 12: 122-125.
[2] Wei, W., (2010) Development and application of marine vhf radio simulator. NAVIGATION OF CHINA, 33: 16-19.
[3] Lin, Ch.Ch., Hong, Y.Zh., and Dong, F., (2007) Design and implementation of a gmess simulator based on visual c#.net . Journal of System Simulation, 19: 1256-59.
[4] Liu, Sh., Shi. G.Y., and Zhang, Y.Q., (2010) Design and implementation of communication software based on tcp/ip protocol and multithreading technology. Computer Engineering and Design, 31: 1417-20.
[5] Zhu, L., Zhang, X.H., and Xia, Zh., (2013) Optimization of polygon intersection & union algorithm based on operation on doubly-linked list” . Computer & Digital Engineering, 41: 463-465.
[6] Jin, L., (2009) Use c language's pointer insert linked lists in data structure. Computer Knowledge and Technology, 5: 4575-76.