I.J. Education and Management Engineering **2012, 8, 65-71**
Published Online August 2012 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijeme.2012.08.11

***Available online at* http://www.mecs-press.net/ijeme**

An Example of Course Project of Real Time Multitask Programming

[a] Yu Jianxin, [b] Zhao Ying gong, [c] Li You, [d] Kuang Hongyu

Department of Computer Science and Technology Nanjing University Nanjing, China Zip code 210093

Abstract

This paper states a multi-task programming course project experiment item of real time operating system VxWorks. The project is an emulator of railway ticket vending machine. Its research background is Wuhan-Guangzhou high speed railway line ticket vending machine. The contents of the paper includes process flow, function design and analysis, data flow analysis, task division and definition, semaphore control application, test results, etc.

Index Terms: Embedded system; education; course project; multitask scheduling; emulator; vending machine; semaphor

1. Foreword

In recent years, an embedded system (ES for short) software development course is offered to master students in the Department of Computer Science, Nanjing University (NJUCS for short). Half of that course time is allotted to theoretical knowledge lecture, and the other half is allotted to lab experiment teaching. There are several samples of course projects used in our ES lab to help students develop their own course projects. In this paper we introduce a sample of the ES course projects, which is a railway ticket vending machine emulator (RTVME for short) [1]. The background of this application is the high-speed railway line between Wuhan and Guangzhou, which are two big cities in south China.

The RTVME program is divided into multiple tasks. The whole control flow of RTVME is implemented by Inner Process Communication (IPC for short), which includes giving and taking semaphore or sending and receiving message [2]. Parameters delivery among task functions is mainly implemented by message queue. Now, the prototype of RTVME runs well and can be used as a template for master students to build their own ES course projects. With the help of RTVME, the students can easily engage in research and development of other types of ES application items, such as vending machines for small commodities, cinema ticket vending machine, subway ticket vending machine, airline boarding ticket auto-registration machine, etc.

Corresponding author:
E-mail address: [a] yujianxn@nju.edu.cn, [b] zero1986618@gmail.com, [c] leo86@seg.nju.edu.cn , [d] hector.khy@gmail.com

2. Operation flow analysis

RTVME runs on ARM9 experiment board. The basic operation steps of RTVME are: (1) Start, the LOGO picture and Wuhan-Guangzhou high-speed railway line map appear on LCD screen of the board. RTVME prompts the manager to calibrate clock. It lets the manager to input start station, in which RTVME is installed. (2) Displaying every stop station and standard ticket price between two stations. RTVME lets user input a destination station. (3) Displaying all business trains number between start stations to destination stations. Arriving time and departure time of these trains are listed too. At the same time the user can input selected train number by a 16 keys keyboard which is arranged in 4 rows and 4 columns. (4) The emulator searches its database to find whether enough selected tickets are available. If sold out or not enough, RTVME prompts the user it cannot supply the ticket, and goes back to step (2). Otherwise, RTVME lets user input tickets number he will buy. (5) RTVME searches its database and assigns train car number and seat number of selected trains for selling ticket. (6) After making total price calculation, RTVME refurbishes the LCD screen, draws various RMB currency picture, displays total price, and prompts user to make simulated money payment by touch screen input method many times. Once the accumulated payment is equal or exceeding the required payment, the simulated payment is over and RTVME goes to next step. (7) Calculating the sum of simulated input money. Store each received virtual bill into virtual money database. When simulated payment is exceeding the required payment, RTVME make simulated change calculation. (8) RTVME outputs virtual service results of this virtual ticket sale. At first to make virtual change, then to make virtual sold ticket output. (9) The user executes virtual taking change and virtual taking ticket. (10) RTVME draws service end picture and goes to step (2).

Afterwards, from operation step (2) to operation step (10) forms an infinite loop, which are being executed repeatedly. In the whole circular process, once executing time of any operation step exceeds the restricted time, RTVME will return to operation step 2 so as to restart the virtual vending service again. In this case, if some virtual money had been paid, the RTVME will give user a full virtual refund before going to operation step (2).

3. Basic Function Design

Having accomplished RTVME operation flow analysis, we give a solution to RTVME function design.

(1) LCD display function: Displaying information is such as railway line map (marking all stops), start and destination station selecting input picture, trains number and standard ticket price that stop at start station and destination station, total payment prompting picture, virtual payment picture, virtual change output picture, and virtual ticket output picture. In order to implement these functions, an embedded GUI tool is necessary, here we use windML version 3.0 [3].

(2) LED display function: ARM9 experiment board always installed several LED 7 segment nixie tubes, so we use them to display ticket unit price and purchased ticket number.

(3) Touch screen input and small keyboard input function: Under prompting picture, user can input data such as destination station, train number, purchased ticket number, confirmation for taking virtual payment, and confirmation for taking virtual ticket, etc

(4) Database management functions: There are several database tables built in RTVME. They are train time table, train ticket sales database table, trains car and seats database table, and virtual bill gathering or changing database table, etc. Process function set upon these tables is initialization, data upgrading, outputting datum record which is met search condition, statistics of sold tickets, etc.

(5) Calculation function: Calculation functions RTVME used are listed here. 1) Converting keyboard input or touch screen input into digital value; 2) Purchased tickets total price calculation; 3) Virtual payment accumulation calculation; 4) Virtual money changing calculation. Etc.

(6) Clock process function: Real time clock associated functions includes receiving real time input; saving time into RTC component; outputting current time on LED nixie tubes; etc.

4. Data analysis and database design

RTVME has following database tables:

(1) Train timetable: In this table there are fields such as train number (main key), start station, target station, departure time, arrival time, a hard seat price, a soft seat price, a sleeper bed price, etc. (2) Train ticket table: train number (key), remainder tickets of hard seat, soft seat, and sleeper bed. (3) Train seat table: main fields are train number (key), car number, seat number. (4) Virtual money gathering and changing table. The purpose of this table is to store virtual money paid by user, to provide datum for calculating whether payment is enough, and to make change. The main fields include number of RMB 100 Yuan notes, number of RMB 50 Yuan notes, etc.

5. Data flow analysis

Using Wuhan-Guangzhou high speed rail way line as background, we can draw data flow diagram (DFD for short) of RTVME. Please see Fig. 1.

In Fig. 1, readers can see that when RTVME runs, there are 10 steps in the program execution. Step 1, 2, 3, 4, 6, 9 need man machine interface (MMI for short) operation, step 5, 7, 8, 10 are running at backstage. The main process is in a loop of nine steps, that means RTVME can run 24 hours a day uninterruptedly. And RTVME have some extent simulation fidelity, because user will have the same activity and feeling as if he buys a train ticket at front of real ticket vending machine.

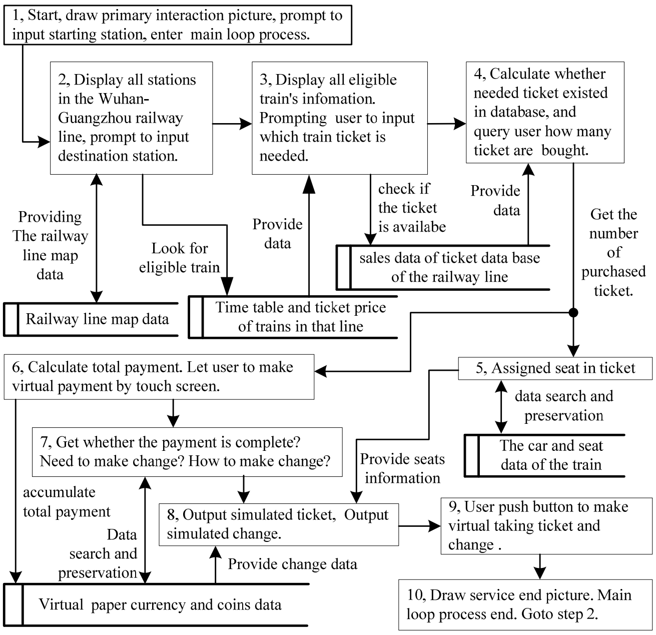
1, Start, draw primary interaction picture, prompt to input starting station, enter main loop process.
2, Display all stations in the Wuhan-Guangzhou railway line, prompt to input destination station.
3, Display all eligible train's infomation. Prompting user to input which train ticket is needed.
4, Calculate whether needed ticket existed in database, and query user how many ticket are bought.
Providing The railway line map data
Look for eligible train
Provide data
check if the ticket is availabe
Provide data
Get the number of purchased ticket.
sales data of ticket data base of the railway line
Railway line map data
Time table and ticket price of trains in that line
6, Calculate total payment. Let user to make virtual payment by touch screen.
5, Assigned seat in ticket
data search and preservation
The car and seat data of the train
7, Get whether the payment is complete? Need to make change? How to make change?
Provide seats information
accumulate total payment
Data search and preservation
8, Output simulated ticket, Output simulated change.
9, User push button to make virtual taking ticket and change .
Provide change data
Virtual paper currency and coins data
10, Draw service end picture. Main loop process end. Goto step 2.

Figure 1. Data Flow Diagram of RTVME

6. Task division and design

The MMI input devices ARM9 experiment board provided are small keyboard, touch screen, and interrupt button, while output devices are LCD, LED nixie light, beeper. According to H.Gomma’s principle, transforming process which has strong dependent relation with I/O device should be divided as a single separate task. So, operation program of keyboard, touch screen, LCD, and LED nixie light are made separate tasks [4].

Then on the basis of task division and according to function cohesion principle, input module, output module, and calculation module should be made as an independent task [4]. In accordance with above two principles, we divided source program of RTVME into 7 tasks. The task IDs are listed as tLCD, tLED, tKeybod, tTouch, tInput, tOutput, and tCalcu. They are corresponds to tasks of LCD, LED, keyboard, touch screen, input, output, and calculation.

RTVME uses a .txt data file and a .jpg picture file, which are stored in flash memory. Writing or reading flash files are executed by their input task and output task, so we does not make a flash file operation program as a task.

Key points of this course project are: (1) Giving and taking semaphores is main method to control the execution sequence of tasks; (2) Sending and receiving message through message queue is the basic method to transmit parameter among tasks.

7. entry function design

In this emulator, entry function is declared as void RetvmStart(void). In booting stage, the entry function is called by usrApplnit function. The pseudo-code list is given as follows.

```
Necessary variables and arrays are declared
Several binary semaphores are created
Some message queues are created
drawWelcome(rStart, a); /*drawing the LOGO*/
wd = wdCreate ( ); /*wd timer is set, then used by tInput task*/
sigInit ();  /*soft signals are initialed*/
initialugl(); /*MMI picture is drawn*/
init_ticketnumber(); /*Ticket database is set*/
Seven tasks are created
while(1)
{
semTake (semRestartId, WAIT_FOREVER);
taskRestart(taskID); /*Restart all seven tasks*/
}
```

8. Semaphore application

In RTVME design, the synchronization and mutual exclusion relationship among tasks are executed and managed by binary semaphores and message queues [5][6]. We use eight semaphores in the program. Fig. 2 illustrates binary semaphores and message queues application in RTVME. In Fig. 2, real line rectangle express task, and dashed line rectangle express functions called by task. Real line arrows express semaphore given and taken relationships among tasks. Semaphore identifiers which are near real line have prefix string "sem", followed by abbreviation name of taking task. For example: semKeybod. It means the semaphore is taken by keyboard task.

Apart from each semaphore corresponds to a task in the infinite loop; In Fig. 2, there is a input operation time limit exceeding semaphore semRestart. A watch dog timer supervises the operation time of current input task. If the user's input operation time exceeds the time limit set by watch dog, the semaphore semRestart is given. So in the infinite loop of entry function, all taskRestart statements, which are pended on semRestart, will get ride of pending state, they run immediately. At this moment, all tasks are re-setup, except entry function. This results in RTVME returns initial MMI picture of ticket selling loop. Please see the pseudo-code list of entry function.

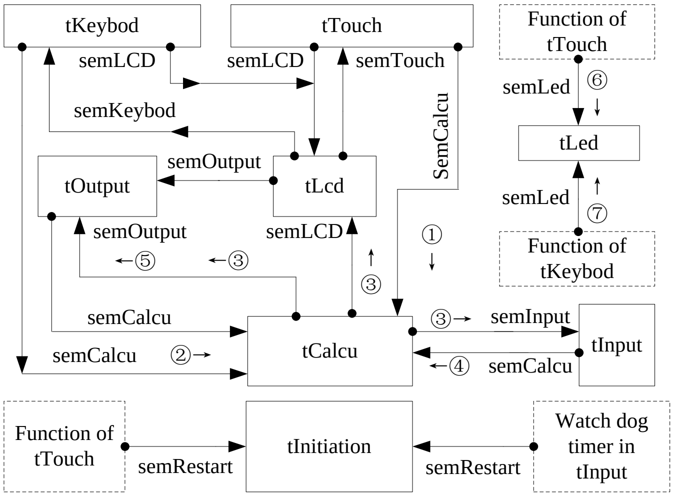

Figure 2. IPC relationships among tasks

9. Message application

Message queue is a mutual communication mean among joint working tasks [7]. In this course project, 5 message queues are used. Message queues identifiers have prefix string "msgQ", followed by shortened form of receiving task name. Their names and purposes are listed as follows.

<1> msgQTou, touch screen input message
<2> msgQCal, user duo to pay money for tickets
<3> msgQInp, to be processed input data
<4> msgQOut, to be processed output data
<5> msgQLed, LED output data

Message delivery functions done by these five message queues are listed below.

(1) tTouch sends unit train ticket price to tCalcu through msgQTou
(2) tKeybod sends purchased train ticket number to tCalcu through msgQTou
(3) tCalcu sends total amount of user's payment to tLCD, tInput and tOutput through msgQCal
(4) tInput sends the virtual money input through keyboard or touch screen to tCalcu through msgQInp
(5) tTouch sends unit train ticket price to tLED through msgQLed
(6) tKeybod sends purchased ticket number to tLED through msgQLed
(7) tTouch sends unit train ticket price to tCalcu through msgQTou

The relationship of message sending and receiving is also shown in Fig. 2. The message delivery operation is marked by number in circle. That number and the number in message delivery function list are coincident to express the message queue operation. Arrows near the circle stand for message transmitting direction. Arrow head direction points the task which receives the message and tail direction represents the task which sends the message.

10. Key task process flow design

LCD and touch screen are main peripherals of ARM9 education experiment board. They are controlled by task tLCD and tTouch respectively. Fig. 3 is the flow chart of tLCD task function and Fig. 4 is the flow chart of tTouch. In initial operation stage, tLCD and tTouch interact three times by two semaphores, so as to get inputs of start station, destination station and train number. Then tLCD and tKeybod execute synchronization operation by semaphore, to get purchased ticket number. And then, by giving and taking semaphore, tLCD, tOutput and tCalcu collaborate to complete user's virtual payment. At last, when virtual payment is up to requirement, tLCD and tOutput cooperate with each other to complete virtual change and virtual ticket output.

In Fig. 3 and Fig. 4, statements within while(1) loop are key processing statements of tLCD and tTouch in one railway ticket auto-selling activity. They correspond to most operation steps within the main loop, which is made up from step 2 to step 10 shown in Fig. 1.

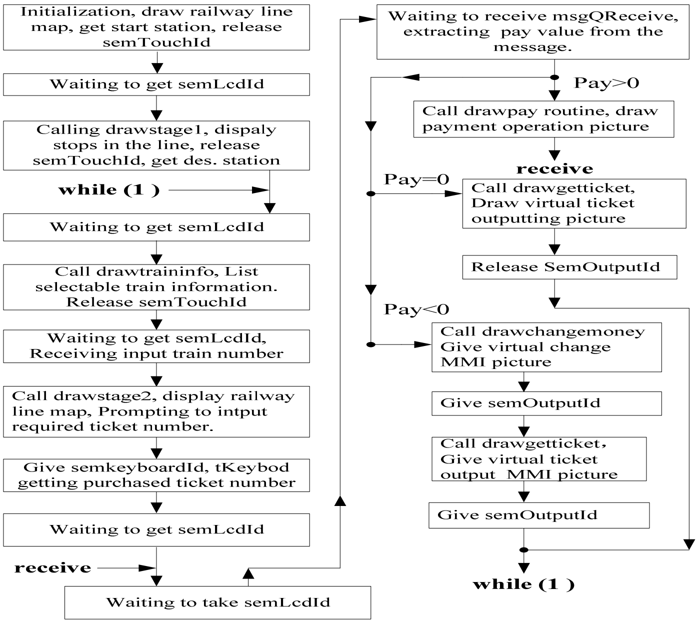
Initialization, draw railway line map, get start station, release semTouchId
Waiting to get semLcdId
Calling drawstage1, dispaly stops in the line, release semTouchId, get des. station
while (1)
Waiting to get semLcdId
Call drawtraininfo, List selectable train information. Release semTouchId
Waiting to get semLcdId, Receiving input train number
Call drawstage2, display railway line map, Prompting to intput required ticket number.
Give semkeyboardId, tKeybod getting purchased ticket number
Waiting to get semLcdId
receive
Waiting to take semLcdId
Waiting to receive msgQReceive, extracting pay value from the message.
Pay>0
Call drawpay routine, draw payment operation picture
receive
Pay=0
Call drawgetticket, Draw virtual ticket outputting picture
Release SemOutputId
Pay<0
Call drawchangemoney Give virtual change MMI picture
Give semOutputId
Call drawgetticket, Give virtual ticket output MMI picture
Give semOutputId
while (1)

Figure 3. Flow chart of tLCD task function

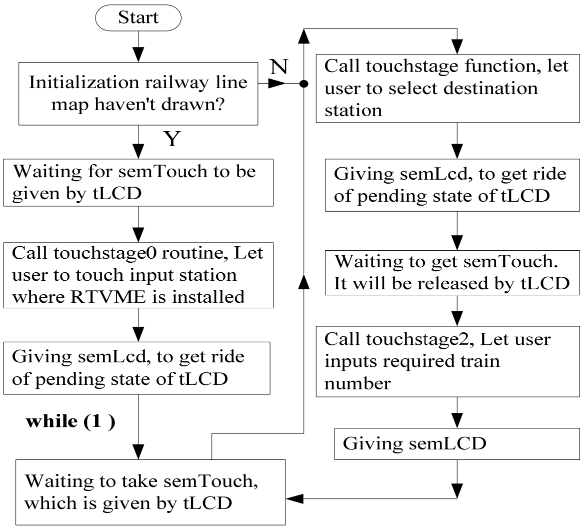
Start
Initialization railway line map haven't drawn?
N
Y
Waiting for semTouch to be given by tLCD
Call touchstage0 routine, Let user to touch input station where RTVME is installed
Giving semLcd, to get ride of pending state of tLCD
while (1)
Waiting to take semTouch, which is given by tLCD
Call touchstage function, let user to select destination station
Giving semLcd, to get ride of pending state of tLCD
Waiting to get semTouch. It will be released by tLCD
Call touchstage2, Let user inputs required train number
Giving semLCD

Figure 4. Flow chart of tTouch task function

The calculation task completes three operations. (1) It computes total price of purchased train tickets. (2) It communicates tInput task to get how much money should be paid by user. (3) It computes whether virtual payment meets the payment requirement. If simulated payment is equal to payment requirement, RTVME make virtual ticket output right away. If simulated payment exceeds payment requirement, RTVME firstly make virtual change calculation, then make virtual change and ticket output. Otherwise, RTVME prompt user to continue virtual payment.

11. Test result

We use Wuhan-Guangzhou high-speed railway as the test case. The information of six stations is used in the test case. These stations are Wuhan, Yueyang, Changsha, Hengyang, Shaoguan and Guangzhou. For testing we write LCD display routines and touch screen button routines of these stations in RTVME program and input valid high speed train number, cars and seats information of the trains and ticket price list of the trains into database of RTVME.

Test results showed that more than 30 test items, which account for 90% of total test items, obtained normal test results, MMI input operation ran well, output data were correct, and all virtual actions were normal.

12. Conclusions

Since three years ago ES software development course has opened in NJUCS, RTVME is used by many graduate students as a course project experiment. Now, this multi task application project has about 2600 lines of ANSI C statements and VxWorks API function calling statements. From the viewpoint of teaching, RTVME plays a good role in training students to boost their real-time multi-task application programming ability. RTVME can be considered as a practicable ES software teaching paradigm. For future work, the documents of RTVME need to be improved, network functions and peripheral, which is closer to real life, need to be added so as to let this course project play an active part in training high quality ES talents.

References

[1] Tang Kun, Wang Mingzhe, Peng Huaijun, “Research and implementation of railway automatic ticketing system,” Railway Computer Application (in Chinese) vol.12, No. 12, page 28, Dec. 2003.

[2] Wind River Systems, Inc. VxWorks Ver 5.5 Programmer's Gue[M], 2nd ed., Mar 2003

[3] Li Hailiang, Shi Pengcheng, “Analysis of WindML graphical interface program frame on VxWorks”, Industrial Control Computer (in Chinese), Vol.20, No. 1, Page 47, 2007...

[4] Luo Guoqing, develop embedded software on VxWorks, 1st ed., China Machine Press (in Chinese), April 2004, Page 35.

[5] Wan Liu, Cai Bin, Guo Yudong, “Analysis of multi-task communication and synchronization based on semaphore in VxWorks,” Journal of Information Engineering University (in Chinese), Vol. 5, No. 2, June 2004, Page 53.

[6] Cong Wei, Wang Yong, Yu Hongkun, “VxWorks Semaphore Applied in Task Synchronization,” Microcomputer Development (in Chinese), Vol. 14, No. 7, July 2004, Page 44..

[7] Kong Xiangying, Gui Baizhi, Embedded Real - Time OS VxWorks and its development environment Tornado, 1st ed., China electric power press (in Chinese), Jan. 2002, Page 152.