

table of Contents

table of Contents

Chapter OneOne ticket and one control specification for bill printing.....	1
1.1 introduction.....	1
1.2 Analysis of reasons for losing tickets.....	1
1.3 Solution	2
1.4 Host computer software processing flow	3
1.4.1 Cashier task flow.....	3
1.4.2 Data sending task flow	4
1.5 New command	5
1.5.1 One ticket one control data block start instruction.....	5
1.5.2 One vote one control data block end instruction	5
1.6 Encapsulated print data block format.....	6

Chapter OneOne ticket and one control specification for bill printing

1.1 introduction

In traditional POS printers, there is often a loss of orders, especially in the background kitchen printing. In order to solve this problem, this document regulates a process to make printing stable and reliable.

1.2 Analysis of reasons for losing tickets

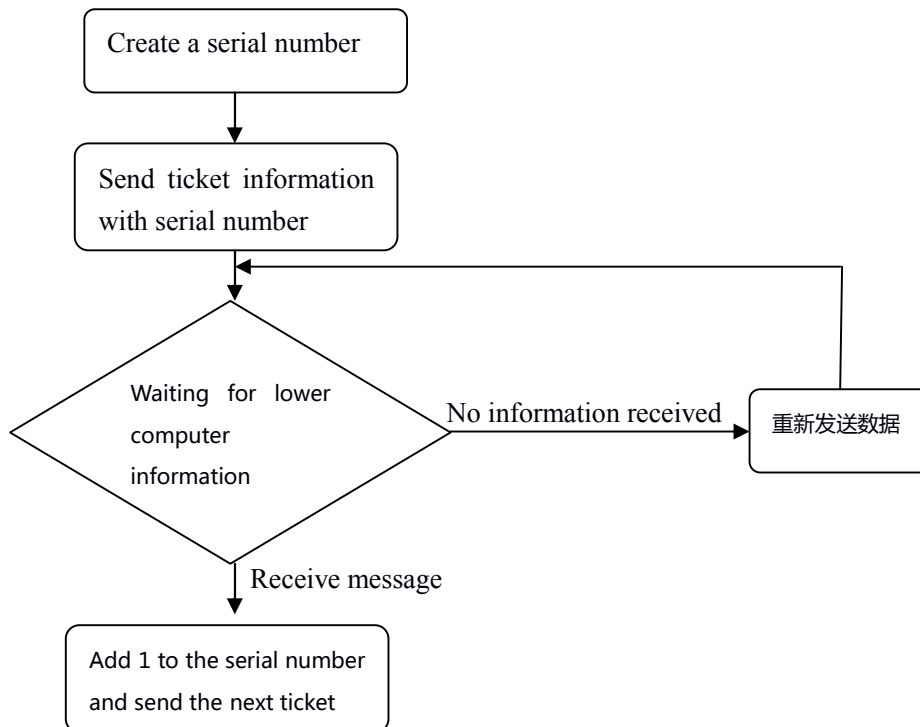
The main reason for POS printers losing orders is that the print data channel is only one-way data, that is, the upper computer sends print data to the lower computer, and the lower computer prints the data. The status of the lower computer, such as the buffer of the lower computer is full, or the printing of the printing receipt is completed, cannot be fed back to the upper computer in real time. In this way, after the data from the upper computer is sent, has the lower computer received it? Is it printed after receipt? I never knew.

1.3 Solution

The solution is to control each ticket and send a single ticket. The host computer needs to perform the following steps.

1. A serial number is established for each printed ticket.
2. A bidirectional data channel is recommended. The lower computer can send data to the lower computer, and the lower computer can return information to the upper computer.
3. When the host computer sends data to the host computer, the serial number is sent first, and then the print data is sent.
4. After the serial number is received by the lower computer, it will be cached first. After printing, the serial number will be sent to the upper computer to indicate that the bill has been successfully printed.
5. After the upper computer sends a piece of print data, wait for the serial number returned by the lower computer.
6. If within 10 seconds, you do not wait for the serial number sent back by the lower computer, it means that the lower computer failed to receive the ticket, and the upper computer reissued the ticket, the serial number remains unchanged.
7. If the host computer fails to retry 3 times, please check the hardware.
8. If the serial number returned by the lower computer is successfully received, it means that the ticket has been successfully printed. The serial number is incremented by one, and the upper computer sends the next ticket.

Figure Chapter.1 One ticket one control host computer single bill sending process

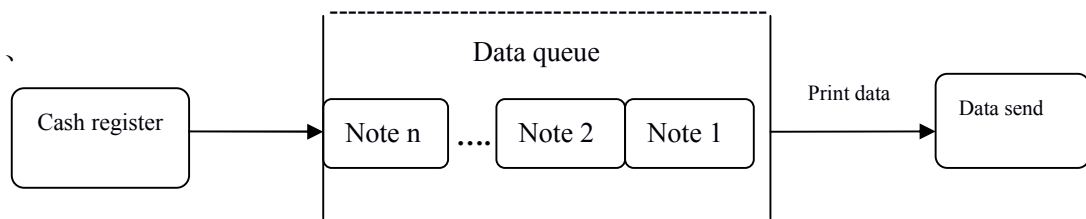


1.4 Host computer software processing flow

The POS host computer software is mainly divided into two parts: cash register and data sending. After adopting the method of one ticket and one control, the data sending process of the host computer software needs to add a sending data queue.

The relationship between the cashier task and the data sending task is as follows Figure Chapter.2 Shown.

Figure Chapter.2 Relationship between cash register task and data sending task



Then the cash register task and data sending task need to be implemented according to the following steps.

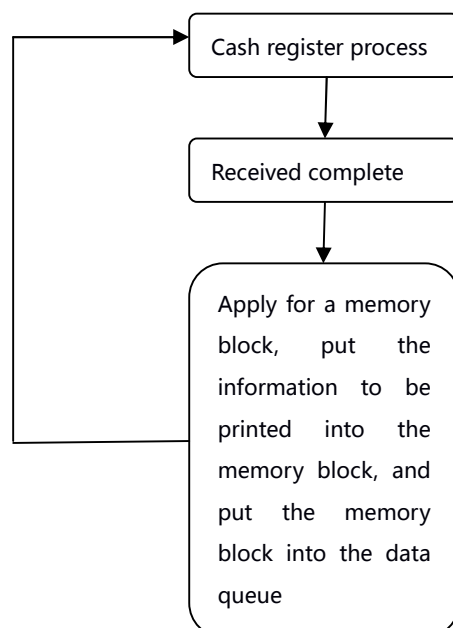
1.4.1 Cashier task flow

1. Normal cash register process
2. Cashier completed
3. Apply for a memory block, copy the data to be sent into the memory block, and put the memory block in the queue.

Note: The data queue here is not in bytes, but in blocks.

4. After putting the data to be sent into the sending queue, continue to collect cash.

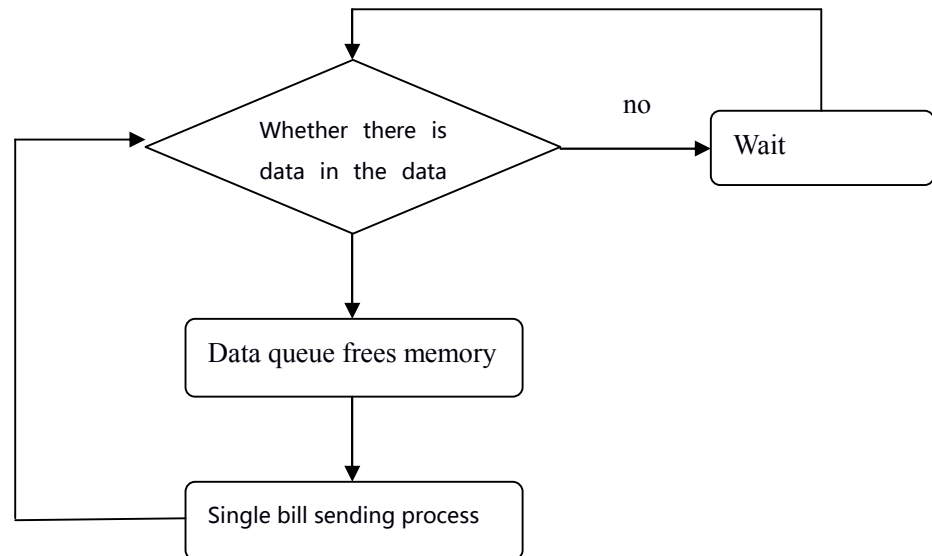
Figure Chapter.3 Cashier task flow



1.4.2 Data sending task flow

1. Determine whether the data queue is empty
2. If empty, wait for data
3. If there is data, enterFigure Chapter.1 One ticket one control host computer single bill sending process

Figure Chapter.4 Data sending task flow



1.5 New command

In order to realize one ticket and one control, it is necessary to add a data block start instruction and a data block end instruction to encapsulate the print data block.

1.5.1 One ticket one control data block start instruction

【format】 ASCII GS ## S SN

Hex 1D twenty three 53 SN

Function: Mark the beginning of the print data block, and the subsequent content of the print data

SN4BYTE serial number

It is recommended that the SN serial number be expressed as a decimal number, byTable Chapter.1It consists of 4 parts shown.

Table Chapter.1 Serial number composition

2 people	2 people	2 people	4
year	month	day	serial number

For example, the SN of the first ticket on August 13, 2013 is 1308130001 in decimal,

The hexadecimal representation is 4D F8 7A D1,

Because the data is in little-endian mode, the corresponding hexadecimal command is

1D 23 53 D1 7A F8 4D

1D 23 53 4E 61 BC 00

1.5.2 One vote one control data block end instruction

【format】 ASCII GS ## E

Hex 1D twenty three 45

Function: Mark the end of the printing data block, the lower computer needs to send back the response information after receiving the instruction,

table Chapter.2 The lower computer sends back the information content

ASCII	GS	##	E	SN (4BYTE)
Hex	1D	twenty three	45	SN (4BYTE)

1.6 Encapsulated print data block format

Encapsulate the data content of the start and end commands of a ticket and control data block, as followsFigure Chapter.5Shown.

Figure Chapter.5 Encapsulated print data block format

One ticket one control data block start instruction
Print data block Content
One ticket one control data block end instruction