

**MỘT SỐ VẤN ĐỀ TRONG XÂY DỰNG HỆ THỐNG
GỬI/NHẬN SMS DÙNG MÔ - ĐEM GSM
SOME PROBLEMS IN BUILDING THE SMS SENDING
AND RECEIVING SYSTEM USING GSM MODEMS**

**ThS. NGUYỄN HẠNH PHÚC
KS. NGUYỄN PHẠM TUẤN**

Khoa Công nghệ Thông tin, Trường ĐHHH

Tóm tắt:

Các mô-đem GSM hỗ trợ tập lệnh AT chuẩn để giao tiếp với ứng dụng trên máy tính. Ngoài ra, mô-đem có thể có tập lệnh mở rộng. Nếu ứng dụng máy tính sử dụng tập lệnh AT chuẩn để giao tiếp với mô-đem thì ứng dụng đó không phụ thuộc vào thiết bị. Nghiên cứu này cho phép xây dựng các ứng dụng gửi/nhận tin nhắn SMS và đề nghị mô hình dịch vụ máy chủ SMS sử dụng đồng thời nhiều mô-đem GSM khác nhau.

Abstract:

GSM modems help AT files in communication with its application on the computer. Besides, modems may have open files. When the computer application uses the standard AT to communicate with modems, the application doesn't depend on the device. The analysis allows building SMS sending and receiving system.

1. Đặt vấn đề

Tin nhắn SMS (Short Message Service) đầu tiên được truyền đi trên mạng GSM của Châu Âu và nó đã đặt một dấu mốc lớn trong lịch sử điện thoại di động (ĐTDD). Ngày nay, điện thoại di động là thiết bị không tách rời trong cuộc sống. Đặc biệt, SMS là dịch vụ không thể thiếu đối với giới trẻ. Năm 2002, sóng dịch vụ tin nhắn đa phương tiện (MMS) lần đầu tiên được đưa vào sử dụng. SMS được sử dụng rộng rãi trong hầu hết các lĩnh vực kinh tế xã hội như tin nhắn cá nhân, các dịch vụ thông tin, thông báo về thông điệp âm thanh và fax, chat, định vị (GPS), đọc thông số và điều khiển từ xa. MMS mở ra những cơ hội kinh doanh và được kỳ vọng có một vị trí tốt đẹp trong sự phát triển của các kênh phân phối nội dung thương mại như video, nhạc, tin tức,...

Có một số cách xây dựng hệ thống gửi/nhận SMS.

1) Dùng Simple Network Pager Protocol – trước đây phổ biến ở Mỹ và một số nước, hiện nay không còn được dùng rộng rãi nữa, vì hầu hết sở hữu ĐTDD chứ không phải máy nhắn tin.

2) Dùng TAP/UCD Protocol – gửi SMS thông qua kết nối dialup, vừa chậm và vừa không ổn định.

3) Dùng mô-đem GSM – phụ thuộc vào tốc độ xử lý SMS của mô-đem. Đa số các mô-đem GSM tích hợp sẵn trong ĐTDD được cho là không hoàn toàn là một mô-đem GSM hoàn chỉnh, có tốc độ gửi/nhận SMS chậm, khó có thể đáp ứng chế độ làm việc liên tục 24/24 vì nhiều lý do, trong đó có lý do liên qua đến vấn đề của pin. Các mô-đem GSM chuyên dụng có giá không cao, tốc độ gửi/nhận trung bình 1.000 SMS/giờ và đáp ứng tốt chế độ làm việc liên tục.

4) Thông qua dịch vụ HTTP – SMS được gửi đến/nhận từ một địa chỉ IP và máy chủ ở đó đảm nhận việc trung chuyển SMS. Có thể sử dụng SMS gateway trung gian theo cách này và thông thường là miễn phí đối với chủ dịch vụ và người dùng cuối sẽ phải trả phí SMS theo quy định của dịch vụ SMS gateway đó.

5) Xây dựng SMS gateway bằng cách sử dụng các giao thức khác nhau, chẳng hạn Short Message Peer to Peer Protocol (SMPP). Giải pháp này cho phép gửi/nhận SMS với tốc độ cao. Tuy nhiên, nó khá đắt đỏ trong cả đầu tư ban đầu, cả phí hỗ trợ hàng tháng và phức tạp trong thủ tục đăng ký. Đối với các tổ chức kinh tế/xã hội có nhu cầu ở mức độ trung bình trong việc gửi/nhận SMS, giải pháp sử dụng các mô-đem GSM tối ưu nhất.

Hiện nay trên thị trường đã có nhiều phần mềm cũng như các mô-đun cho phép tích hợp khả năng gửi nhận SMS cũng như phần mềm máy chủ dịch vụ SMS với giá cao. Dù các lệnh AT mở rộng của từng dòng sản phẩm hầu như không giúp tăng tốc độ làm việc với SMS, một số nhà sản xuất điện thoại có cung cấp API để xây dựng các ứng dụng làm việc với thiết bị, nhưng chủ yếu hỗ trợ các thiết bị sản phẩm của họ. Tuy nhiên, chưa gặp một tài liệu nào phân tích và trình

bày tương đối hoàn chỉnh cách thức xây dựng các hệ thống hoặc các mô-đun cho phép gửi nhận SMS dùng mô-đem GSM bằng hệ thống lệnh AT thông qua cổng giao tiếp COM, đặc biệt trong việc kiểm soát chặt chẽ quá trình gửi/nhận từng SMS cụ thể.

2. Các kỹ thuật cơ bản

2.1. Lệnh AT

Để gửi/nhận SMS, cần kết nối thiết bị là Mô-đem GSM vào cổng COM của máy tính. Nếu mô-đem kết nối vào máy tính bằng cổng USB thì cần phải biết tên của thiết bị trong hệ thống hoặc thiết bị đã được kết nối qua cổng COM emulated nào. Chương trình máy tính và thiết bị trao đổi dữ liệu thông qua hệ thống lệnh AT (Attention commands) chuẩn. Tuy nhiên, tùy vào thiết bị và nhà sản xuất, mỗi mô-đem có thể có hệ thống lệnh AT mở rộng nhằm tối ưu và nâng cao khả năng kết nối của thiết bị với máy tính [2, 4, 6, 7, 11].

LỆNH AT	CHÚ THÍCH
+CMGF	Chọn chế độ làm việc
+CPMS	Chọn lưu trữ
+CSMP	Thiết đặt thông số trong chế độ văn bản
+CSAS	Lưu giữ các thiết lập SMS
+CSCS	Chọn kiểu mã hoá dữ liệu
+CSDH	Xem thiết lập SMS
+CMGR	Đọc SMS xác định từ thiết bị
+CMGL	Đọc tất cả SMS theo loại từ thiết bị
+CMGS	Gửi SMS
+CMGW	Ghi SMS vào bộ nhớ
+CMSS	Gửi SMS đã lưu trong bộ nhớ
+CMGD	Xoá bộ nhớ

Trong chương trình, đầu tiên cần tạo một kết nối cổng COM cho mỗi mô-đem, sau đó gửi đến cổng COM những lệnh AT tương ứng với biểu sau và đọc kết quả thực thi lệnh AT từ cổng COM. Cần kiểm tra kết nối và mô-đem bằng cách sử dụng nhóm lệnh: AT, +CPIN, +CSCA, +CGMI, +CGMM, +CMEE, +CSMS, +CSQ, +CBC trước mỗi phiên làm việc. Nhóm lệnh AT trong biểu dùng để làm việc với SMS.

Để đọc thiết lập hiện tại, dùng lệnh AT có thêm ký tự '?'. Để xem những giá trị nào có thể thiết lập, dùng lệnh AT có thêm 2 ký tự '=?''. Để thiết lập giá trị thông số mới, dùng lệnh AT có thêm ký tự '=', và theo sau đó là những giá trị thông số mới. Để gửi một nội dung đến chỉ một khách hàng, sử dụng lệnh + CMGS là tối ưu nhất. Tuy nhiên, có những nội dung cần gửi đến nhiều khách hàng khác nhau. Trong trường hợp này nên dùng lệnh +CMGW ghi SMS lên bộ nhớ của mô-đem, sau đó dùng lệnh +SMSS để gửi SMS đó đến các khách hàng khác nhau. Cách này cho phép nâng cao tốc độ làm việc của mô-đem nhờ giảm thiểu trao đổi thông tin giữa mô-đem và chương trình.

Có thể gửi SMS theo hai chế độ văn bản (text mode, +CMGF = 1) và chế độ mặc định PDU (Protocol Data Unit, +CMGF = 0). Giá trị các thiết lập thông số cho chế độ văn bản và PDU có khác nhau cho một số lệnh AT. Chẳng hạn, với lệnh đọc tất cả các tin nhắn +CMGL tiếp nhận các thông số "REC UNREAD", "REC READ", "STO UNSENT", "STOSENT" và "ALL" trong chế độ văn bản; trong khi đó, trong chế độ PDU sẽ là các giá trị 0 – 4. Ngoài ra, không phải tất cả các Mô-đem GSM đều hỗ trợ chế độ văn bản. Thử nghiệm cho thấy không chỉ những điện thoại lạc hậu, mà một số điện thoại hiện đại thuộc loại bậc nhất hiện nay, chẳng hạn W580, cũng không hỗ trợ chế độ văn bản khi làm việc với các chương trình trên PC. Trong khi đó, chế độ PDU thì tất cả các mô-đem hỗ trợ và chế độ này cho phép gửi hình ảnh và nhạc chuông. Suy ra, khi xây dựng một chương trình làm việc với các Mô-đem GSM, cần phải nghiên cứu tài liệu kỹ thuật của từng loại mô-đem để có thể thiết lập đúng những thông số mà mô-đem đó hỗ trợ, và trong mô-đun làm việc với các mô-đem, cần xác định loại và mô-đem mô-đem, sau đó sử dụng những thông số mà mô-đem đó hỗ trợ; hoặc dùng lệnh AT có thêm '=?' để kiểm tra, những giá trị nào các thông số tương ứng của một lệnh AT cho một mô-đem cụ thể có thể tiếp nhận. Tất cả các mô-đem đều phải hỗ trợ tập hợp các lệnh AT chuẩn. Nếu mô-đun sử dụng tập hợp AT chuẩn để làm việc với các mô-đem, thì hệ thống sẽ không bị phụ thuộc vào thiết bị được sử dụng.

2.2. SMS

Thông thường mỗi SMS có độ dài tối đa 160 ký tự trong bảng chữ cái 7 bit. Các SMS 8 bit có độ dài tối đa 140 ký tự thường là SMS thông minh chứa hình ảnh và nhạc chuông hoặc là các thiết lập WAP. Các SMS chứa thông điệp gồm các ký tự unicode 16 bit (UCS2) có độ dài tối đa 70 ký tự. Các SMS có độ dài lớn hơn độ dài tối đa vẫn có thể được truyền tải dưới dạng ghép nối nhiều SMS phân đoạn với độ dài chuẩn. SMS cũng có thể phân chia thành các loại sau: SMS-SUBMIT, SMS-DELIVERY, SMS-STATUS-REPORT, SMS-SUBMIT-REPORT, SMS-DELIVERY-REPORT, SMS-COMMAND. Cấu trúc của một SMS được mô tả chi tiết theo từng loại trong [1]. Ba loại SMS đầu tiên chúng ta quan tâm nhất khi xây dựng hệ thống tra cứu thông tin. Có thể kiểm tra lý do gửi/nhận SMS không thành công thông qua hai loại SMS sau cùng. Sau đây là ví dụ dùng lệnh AT gửi SMS trong chế độ văn bản bằng ngôn ngữ C#:

```
//portName,baudRate,parity,dataBits, stopBits //SMS-SUBMIT, thời gian delivery 3 ngày, xem
SerialPort sp = new SerialPort("COM4", //octet đầu tiên
460800, Parity.None, 8, StopBits.One); sp.write("AT+CSMP=49,169,0,0\r");
//Nếu kết quả trả lại là ERROR, nghĩa là có lỗi //Gửi tin nhắn đến số +841234238986, nếu
//kết nối với mô-đem, nếu không sẽ là OK //thành công: +CMGS n, với n – từ định danh
sp.write("AT\r"); //của SMS vừa được gửi (ví dụ +CMGS:136),
//Sử dụng chế độ văn bản, text mode. +CMGF: //ngược lại: +CMGS ERROR m, với m – mã lỗi.
//ERROR - nếu có lỗi, ngược lại +CMGF OK sp.Write("AT+CMGS="+841234238986+"\r");
sp.Write("AT+CMGF=1\r"); sp.Write("gsmmsmms"+(char)(26));
```

Trong mọi chế độ, thông tin gồm các ký tự Unicode 16 bit được mã hoá thành các cặp đơn vị thông tin 8 bit. Mỗi đơn vị 8 bit gọi là octet. Mỗi cặp octet tương đương với một ký tự Unicode 16 bit ở dạng HEX. Để gửi một tin nhắn chứa các ký tự Unicode 16 bit, cần phải chọn kiểu mã hoá dữ liệu là UCS2 hoặc HEX.

Ví dụ thông điệp “Tiếng Việt” được mã hoá thành: 005400691EBF006E00670020005600691EC70074. Với thông tin Unicode 32 bit, mỗi ký tự chiếm 4 octet.

Tuy nhiên, với một số mô-đem gửi và một số điện thoại nhận SMS có hỗ trợ Unicode, khi gửi SMS có nội dung chứa các ký tự Unicode trong chế độ văn bản, nội dung SMS hiện ra không đúng, và thậm chí không thu được nội dung đúng khi đọc lại SMS đã gửi/đã nhận trực tiếp từ mô-đem/điện thoại trong cả hai chế độ văn bản và PDU. Thử nghiệm cho thấy điều đó đúng với các điện thoại Nokia 6230 và Sony Ericsson W580. Thậm chí đúng với SMS gửi từ trang web của Vinaphone trong chế độ Unicode. Vì thế, khi cần làm việc với SMS qua mô-đem, đặc biệt với các SMS có nội dung chứa ký tự Unicode, tốt nhất nên chọn chế độ PDU.

Khi dùng chế độ PDU, thông tin theo bảng chữ cái 7 bit (septet) thường được mã hoá thành những octet để gửi đi. Và khi nhận, cần phải giải mã nó để hiển thị nội dung SMS cho người dùng. Sau đây là ví dụ mã hoá thông điệp “gsmmsmms” gồm 9 ký tự 7 bit thành các octet.

KÝ TỰ	g	s	m	s	m	s	m	m	s
MÃ	103	115	109	115	109	115	109	109	115
BIN	1100111	1110011	1101101	1110011	1101101	1110011	1101101	1101101	1110011
OCTET	11100111	01111001	01111011	11011110	10011110	10110111	11011011		11100111
HEX	E7	79	7B	DE	9E	B7	DB		73

Và như vậy, thông điệp “gsmmsmms” được mã hoá thành E7797BDE9EB7DB73 bằng cách chuyển số lượng bit cần thiết từ cuối ký tự kế sau (gạch chân) sang đầu của ký tự kế trước để có thể tạo thành một octet 8 bit. Đối với ký tự ‘s’ cuối cùng, vì không còn ký tự nào đứng sau nữa nên được giữ nguyên và tạo thành một octet.

Để xử lý SMS được đúng, đầu tiên phải xác định đó là loại SMS nào dựa vào lệnh đọc SMS từ mục nào và 2 bit số 0, 1 của octet đầu tiên (00 – SMS-DELIVER, 00 –SMS-DELIVER-

REP, 10 – SMS-STATUS-REP, 10 – SMS-COMMAND, 01 – SMS-SUBMIT, 01 – SMS-SUBMIT-REP). Tùy theo cấu trúc của từng loại SMS, phân tích và so sánh SMS-SUBMIT/SMS-SUBMIT-REP và SMS-DELIVER/SMS-DELIVER-REP theo từng cặp để kiểm soát độ chính xác khi gửi/nhận. Tương tự, SMS-SUBMIT/SMS-STATUS-REP để kiểm soát SMS đã được nhận thành công hay chưa. Đối với SMS-SUBMIT, không cần các octet chứa thông tin về Trung tâm dịch vụ tin nhắn (SMSC) khi gửi SMS. Thay vào đó, octet độ dài của thông tin về SMSC sẽ chứa giá trị 0x00 vì thông tin về SMSC đã được cài đặt trong mô-đem. Và octet đó sẽ không được tính vào tổng số octet khi gửi SMS đến mô-đem. Tuy nhiên, thông tin này thường được kèm theo khi đọc PDU của SMS đã gửi từ mô-đem.

Ví dụ, thông điệp “gsmmsmms” gửi đến số điện thoại +84914780898 trong chế độ PDU là 069148192000502100 0C91482143329868000009E7797BDE9EB7DB73 được phân tích như sau:

THÔNG SỐ	OCTET	CHÚ THÍCH
SMSC Info	06	Độ dài của thông tin về SMSC là 6 octet, bao gồm kiểu và số điện thoại
	91	Kiểu số điện thoại quốc tế
	4819200050	Số điện thoại của SMSC là 8491020005
First Octet	21	SMS-SUBMIT, 0x21 = 0010 0001
TP-Message-Reference	00	
TP-Recipient-Address	0C	Độ dài của số điện thoại là 12 chữ số
	91	Kiểu số điện thoại quốc tế
	482143329868	Địa chỉ người nhận 841234239986
TP-Protocol-Identifier	00	SMS thông thường
TP-Data-Coding-Scheme	00	Mã hoá dữ liệu theo bảng chữ cái 7 bit. Nếu là 04 – 8 bit, 08 – UCS2
TP-User-Data-Lenght	09	Độ dài dữ liệu người dùng là 0x09 septet
TP-User-Data	E7797BDE9EB7DB73	Dữ liệu người dùng: gsmmsmms

Số điện thoại và thời gian được mã hoá dưới dạng semi octet đảo ngược theo từng cặp, 84 12 34 23 99 86 thành 48 21 43 32 98 68. Nếu octet kiểu số điện thoại có giá trị 81, số điện thoại sẽ là 01234238986. Khi đó số chữ số của số điện thoại là 11 – số lẻ, thì F sẽ được thêm vào số điện thoại: 1032249389F6. Trong chế độ PDU, độ dài nội dung bao gồm các ký tự Unicode UCS2 được tính theo số octet chứ không phải septet như trong trường hợp thông điệp bao gồm các ký tự 7 bit. Ví dụ, PDU của SMS “Tiếng Việt” gửi đến số điện thoại +84914780898 sẽ là:

0001000B914819740898F8000814005400691EBF006E00670020005600691EC70074(0x14= 0₁₀).

Một báo cáo về tình trạng của SMS đã được gửi đi có từ định danh 136). 10 (0x88) trong chế độ văn bản:

CMGL:23,"RECUNREAD", 6, 136,,,"08/08/07,00:54:54+28", "08/08/07,00:54:54+28", 0 và trong chế độ PDU có dạng như sau: 06 91 4819200050 06 88 0C 91 482143329868 808070 00454582 808070 00454582 00. MCDV SMS có thể xác định một SMS gửi đi đã được nhận thành công hay chưa dựa vào từ định danh n=0x88 (13610) mà lệnh AT+CMGS đã trả lại khi gửi SMS và giá trị của TP-Status. TP-Status = 00 có nghĩa người nhận đã nhận SMS, 01 – không thể xác định đã nhận. Và SMS với thông điệp “gsmmsmms” nhận từ số điện thoại +84914780898 trong chế độ văn bản:

+CMGL:5,"REC UNREAD","+84914780898",,"08/08/07,00:04:29+28", gsmmsmms trong chế độ PDU sẽ là: 06 91 4819200050 24 0B 91 48197408 98F8 00 00 80807000409282 09 E7797BDE9EB7DB73. E7797BDE9EB7DB73 chính là nội dung “gsmmsmms” đã được gửi đi.

Trong trường hợp thông điệp gửi đi hay tiếp nhận không thể được gói gọn trong một SMS chuẩn (chẳng hạn nội dung văn bản dài, hoặc chứa thông tin mô tả các giai điệu, ảnh tĩnh, hình ảnh động), thì chúng được tách ra thành nh ững phân đoạn và mỗi phân đoạn được gói gọn trong một SMS. Trong trường hợp đó, giá trị của octet đầu tiên sẽ được cộng thêm giá trị 64 và các octet 050003RFMXSQ sẽ được chèn vào trước octet cách thức mã hoá dữ liệu. Trong đó RF (8bit) – ký hiệu của thông điệp, MX – số phân đoạn và SQ – số thứ tự của phân đoạn. Nếu số lượng phân đoạn lớn hơn 256, thì các octet được chèn vào sẽ là 050004RFRFMXSQ, trong đó RFRF (16 bit)

là số phân đoạn. Tuy nhiên, các ĐTDĐ ngày nay đa số hạn chế số lượng ký tự có thể thể hiện trên màn hình.

Các bit 0 và 1 của TP -DCS xác định lớp của SMS. Nếu chúng chứa giá trị 00, SMS thuộc lớp 0 – Immediate display message. Khi đó, nội dung SMS sẽ hiện ngay lập tức trên màn hình của người nhận. Lớp SMS này thường dùng để gửi các thông điệp khẩn cấp. Để phân tích các tin nhắn đọc từ mô-đem, cách tốt nhất là dùng chế độ PDU. SMS có thể chứa nội dung là một hình ảnh tĩnh và hình ảnh động (animation) hoặc một giai điệu. Thông tin hình ảnh hoặc âm thanh được mã hoá thành các octet [1, 8, 9, 10] và thường có độ dài lớn hơn độ dài tối đa tiêu chuẩn của SMS, nên thường được chia thành nhiều phân đoạn như trong trường hợp thông tin văn bản. Tuy nhiên, chức năng này không phải là quan trọng cho một hệ thống như đang nghiên cứu. Và đặc biệt, hình ảnh và giai điệu được chuyển tải qua SMS có chất lượng kém. Hình ảnh có số màu tối đa 64, kích thước 255x255 pixel và giai điệu định dạng MIDI. Trong khi đó, hầu hết ĐTDĐ đang được sử dụng ngày nay đều cho phép gửi/nhận các tin nhắn đa phương tiện có thể chứa nội dung bao gồm cả văn bản, hình ảnh, âm thanh và thậm chí video chất lượng cao với các định dạng khác nhau. Tốt hơn, nên nghiên cứu tích hợp chức năng gửi/nhận MMS.

THÔNG SỐ	OCTET
SMSC Info	06 91 4819200050
First octet	06
TP-Message-Reference	88
TP-Recipient-Address	0C 91 482143329868
TP-Service-Center-Time-Stamp	808070 00454582
TP-Discharge-Time	808070 00454582
TP-Status	00
THÔNG SỐ	OCTET
SMSC Info	06 91 4819200050
First octet	24
TP-Originator-Address	0B 91 4819740898F8
TP-Protocol-Identifier	00
TP-Data-Coding-Scheme	00
TP-Service-Center-Time-Stamp	80807000409282
TP-User-Data-Length	09
TP-User-Data	E7797BDE9EB7DB73

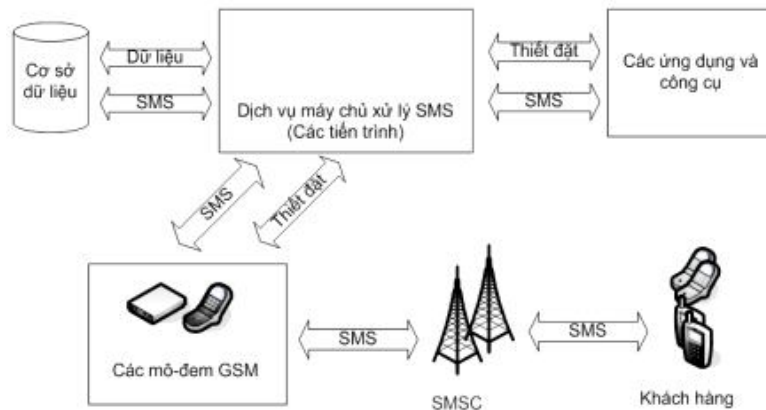
3. Kiến trúc hệ thống

Một hệ thống dịch vụ máy chủ SMS có thể có kiến trúc như trên hình vẽ.

Quá trình làm việc của hệ thống có thể diễn ra theo những kịch bản như sau:

1) Các ứng dụng và công cụ cần gửi những thông báo đến những khách hàng trong một danh sách nào đó. Ứng dụng gửi nội dung cần gửi hoặc những lệnh nào đó đến dịch vụ máy chủ xử lý SMS (MCDVSMS).

Tùy vào loại thông điệp, MCDVSMS có thể đọc thông tin từ cơ sở dữ liệu (CSDL), tạo SMS và ra lệnh cho mô-đem gửi SMS đến SMSC. SMSC sẽ phân tích SMS và gửi nó đến khách hàng cần thiết.



2) Khách hàng gửi một SMS chứa nội dung nào đó hoặc lệnh để tra cứu thông tin. SMSC chuyển SMS đó đến mô-đem, và MCDVSMS sẽ đọc SMS

từ mô-đem. Nó sẽ phân tích SMS nhận được, lưu SMS vào CSDL hoặc chuyển sang cho các ứng dụng theo thiết lập, nó có thể truy xuất dữ liệu từ CSDL, tạo SMS trả lời và ra lệnh cho mô-đem gửi đến SMSC để đến người dùng. Trong mọi trường hợp, nó phải kiểm soát sự chính xác trong việc gửi/nhận và việc khách hàng đã nhận thành công hay thất bại đối với từng SMS, để trong trường hợp cần thiết, nó gửi lại cho khách hàng chính xác một SMS nào đó.

Trong cấu trúc hệ thống, có lẽ vấn đề đáng quan tâm nhất là cấu trúc và cách thức làm việc của MCDVSMS. Ứng dụng MCDVSMS nên được xây dựng dưới dạng một dịch vụ hệ thống. Đối với hệ điều hành Windows họ NT, đó là dịch vụ NT. Mục đích xây dựng MCDVSMS dưới dạng dịch vụ hệ thống là để đảm bảo sự ổn định trong quá trình vận hành và không cần phải đăng nhập

hệ thống để chạy chương trình. Có thể thiết lập để dịch vụ NT chạy theo những thông số cho trước với những quyền hạn khác nhau.

Công việc của MCDVSMS là gửi lệnh AT đến mô-đem, đọc trả lời từ mô-đem, xử lý kết quả đó và có thể gửi lệnh đến mô-đem tùy thuộc vào kết quả xử lý. Cần giải quyết việc đồng bộ hoá quá trình gửi lệnh/nhận kết quả và xử lý kết quả để không xảy ra lỗi và tranh chấp trong hệ thống MCDVSMS. Các môi trường phát triển phần mềm theo mặc định thường tạo ra một chương trình mới chỉ bao gồm một tiến trình và quá trình xử lý câu lệnh trong chương trình xảy ra một cách nối tiếp. Khi đó, chương trình có thể làm việc không hiệu quả và đặc biệt có thể gây ra lỗi trong một số trường hợp.

Cách giải quyết là xây dựng chương trình đa tiến trình (multi-threading) và dùng các đối tượng giúp đồng bộ hoá quá trình truy cập đến một tài nguyên dùng chung của nhiều tiến trình khác nhau như mutex, semaphore, event, interlock trong thư viện WINAPI. Các môi trường lập trình ngày nay đều có các lớp khác nhau triển khai các chức năng tương ứng. Một ví dụ đơn giản cho trường hợp không tổ chức đồng bộ hoá giữa các tiến trình. Nếu chương trình có hai tiến trình T1 và T2 sử dụng chung một biến n – chẳng hạn, số lượng SMS cần gửi. Để tăng giá trị của n lên 2 đơn vị, T1 phải đọc n vào thanh ghi, tăng giá trị thanh ghi lên hai đơn vị, sau đó lưu n lại trong bộ nhớ với giá trị mới.

Trong thời gian đó, tiến trình T2 tăng n lên một đơn vị và lưu vào bộ nhớ trước khi T1 lưu n vào bộ nhớ. Như vậy, kết quả của T2. Nếu MCDVSMS chỉ làm việc với một mô-đem thì đơn giản hơn nhiều so với trường hợp đồng thời nhiều mô-đem. Tuy nhiên, cũng cần xây dựng ít nhất hai tiến trình khác nhau, một phụ trách việc gửi lệnh đến mô-đem, một phụ trách đọc kết quả trả lại từ mô-đem và xử lý khi làm việc với chỉ một mô-đem. Nếu không thực hiện tốt, chương trình có thể gửi đến mô-đem các lệnh khác nhau trong khi chưa đọc hết hoặc chưa kịp xử lý kết quả từ mô-đem. Điều này sẽ gây nên mất kết quả hoặc/và gây nên các lỗi xung đột trong sử dụng mô-đem, chẳng hạn tràn bộ nhớ đệm, mô-đem xử lý không kịp,... Thậm chí có thể gây ra lỗi truy cập đến các tài nguyên dùng chung khác, chẳng hạn danh sách lệnh gửi đến mô-đem bị mất và giá trị n hiện có trong bộ nhớ không đúng.

Tương tự, nếu MCDVSMS làm việc với đồng thời nhiều mô-đem, thì mỗi phân hệ làm việc với một mô-đem đều phải có ít nhất hai tiến trình. Ngoài ra, để MCDVSMS sử dụng các mô-đem được hiệu quả, cần phải tổ chức phân bổ công việc giữa các mô-đem. Nghĩa là tối thiểu, các phân hệ sẽ sử dụng chung danh sách SMS cần gửi. Cũng có thể tổ chức hệ thống theo kiểu mỗi phân hệ đọc kết quả từ mô-đem và gửi nó đến CSDL hoặc mỗi phân hệ là một dịch vụ NT khác nhau, các phân hệ khác khi rồi sẽ truy cập CSDL để đọc danh sách kết quả và xử lý. Như thế, vấn đề tương tranh giữa các phân hệ trong việc đọc danh sách kết quả cũng như danh sách SMS cần gửi sẽ được hệ quản trị CSDL giải quyết thay. Tuy nhiên, như vậy sẽ tăng ít nhất gấp đôi số lượng truy cập đến CSDL từ hệ thống. Hệ quản trị CSDL phải xử số lượng lệnh rất lớn để xử lý một truy cập, có thể tương đương với tổng số lượng lệnh của cả hệ thống MCDVSMS, dù là truy cập đơn giản. Khi đó, hiệu quả tổng quan không cao. Vì thế, tổ chức MCDVSMS theo kiểu đa phân hệ đa tiến trình, các tiến trình sử dụng những tài nguyên dùng chung, để phân bổ công việc giữa các mô-đem là giải pháp tối ưu nhất.

4. Kết luận

Thông qua nghiên cứu vấn đề, chúng tôi nhận thấy:

- Các dịch vụ trao đổi thông tin đơn giản bằng SMS là rất hiệu quả và tiện lợi, đặc biệt dịch vụ tra cứu thông tin.

- Nghiên cứu đã chỉ ra cách thức tạo, gửi, nhận và phân tích một số loại SMS đáng quan tâm nhất khi xây dựng một hệ thống tra cứu thông tin cho cả hai chế độ văn bản và PDU. Phân tích và đi đến kết luận, nên sử dụng chế độ PDU để gửi/nhận và phân tích SMS.

- Nghiên cứu cho phép xây dựng các ứng dụng gửi/nhận SMS không phụ thuộc vào thiết bị và đề nghị mô hình dịch vụ máy chủ SMS đa tiến trình, sử dụng đồng thời nhiều mô-đem GSM khác nhau và phân bổ công việc giữa các thiết bị nhằm sử dụng các thiết bị một cách tốt nhất.

TÀI LIỆU THAM KHẢO

- [1]. G-2403R (GSM/GPRS MODEM) Shenzhen Baihui Technology Co.,Ltd
 [2]. <http://www.inet.vn>

Người phản biện: ThS. Nguyễn Văn Thủy