Porting the 32-bit Run-Time to GSM (Novell)

1. Introduction

This document describes the software changes required to port the GSM 32-bit run-time software to GSM (Novell). In addition, the immense difficulties of porting the 32-bit run-time software to GSM (DOS) and GSM (BOS) are explained.

2. The GSM 32-bit Run-Time Implementation

The changes to implement the 32-bit run-time environment for GSM (Windows NT) comprise numerous changes to the Cobol and Speedbase software layers; and several significant modifications to the nucleus software (in the GLOBAL.EXE module). Brief details of these changes are described in Appendix F of the GSM V8.1k Notes.

The modifications to the Cobol and Speedbase modules (e.g. the P.$SDLM0 library, the changes to the 32-bit loader to recognise and load 32-bit programs, the 32-bit $SDL32 compiler) are all independent of the host o/s.

The changes to the GSM (Windows NT) GLOBAL.EXE module to support the 32-bit run-time environment are very significant:

- A new 32-bit loader/resolver SVC, SVC-79, is now the largest and most complex SVC within the GSM nucleus. This module is responsible for, amongst many other functions, the dynamic allocation and de-allocation of the 32-bit pages. Nearly every operation supported by SVC 79 (there are currently over 70 functions implemented) depend on the memory allocation/de-allocation primitives of a host operating-system that supports a 32-bit virtual memory-management scheme. Such a scheme is available in the Windows Win-32 API (for Windows-95/ 98, Windows-NT and Windows-2000) and by the various API's for the supported versions of Unix (in particular, the various versions of SCO, HP-UX and AIX)

- In addition, to the introduction of the new SVC 79, the 32-bit run-time implementation required vast changes to the Commercial Code Interpreter (to support the 32-bit Commercial Code Language);

- SVC 10, the central interface to the File, Console, Printer and LAN Executives, has been heavily modified to recognise and convert the 32-bit Page:Offset format addresses that are involved when a 32-bit application performs i/o to a file, screen or printer.

- SVC 18, the TFAM text de-blocking SVC, has been modified to support the 32-bit Page:Offset format addresses that are involved when a 32-bit application uses TFAM or Open TFAM.

- SVC 61, the interface to the host o/s, has been modified to support the 32-bit Page:Offset format addresses that are involved when a 32-bit application uses the Open Direct AM, Open TFAM or makes a call to SVC-61 directly.

- A new SVC, SVC 84, has been implemented to provide a direct interface to the SPD driver for 32-bit applications.

All of the changes to GLOBAL.EXE apply to C sources. These changes have been ported to the GSM (Unix) nucleus (i.e. the glintd component) with very little effort.

3. The GSM (Novell), GSM (DOS) and GSM (BOS) Architectures

Although the various changes to the Cobol and Speedbase modules that have been implemented to support the 32-bit run-time and Development System are architecture-independent, the vast changes to the GSM (Windows NT) nucleus (i.e. the GLOBAL.EXE component) will have to be ported to GSM (BOS), GSM (DOS) and GSM (Novell) in order to support the 32-bit run-time system on these architectures.

The 16-bit GSM (BOS), GSM (DOS) and GSM (Novell) nuclei are all inter-dependent. The GSM (BOS) nucleus (architecture code "J5") is written in 16-bit, real-mode 8086 assembler code. This real mode implementation limits the amount of directly addressable memory to 640Kb (with a theoretical limit of 1Mb). The GSM (DOS) nucleus (architecture "JW") is based on the GSM (BOS) nucleus with 3 main changes: All the "J5" GSM (BOS) "hardware controllers" have been simply renamed to "JW" to allow access to raw peripherals (e.g. TCL Superport); a DOS start-up program. GSMLOAD.EXE, acts as a Bootstrap module; several other "DOS controllers" have been implemented to allow GSM to access DOS peripherals (e.g. SSD disk controller). The GSM (Novell) nucleus (also architecture "JW") is based on the GSM (DOS) nucleus with 3 main changes: A few extra controllers have been implemented to access devices using the SPX protocol (i.e. the SPX LAN and SPX Workstation controllers); the DOS start-up program. GSMLOAD.EXE, has been supplemented by SPEEDBAS.OVL to provide the TLI/SPX facilities that provide access the to the Speedbase Btrieve NLM; finally, two NLM's have been developed to provide file and Speedbase database support on the NetWare server.

Enhancing the 16-bit "J5" and "JW" nuclei to support the 32-bit run-time system is all but impossible:

- A bespoke Virtual Memory Manager would be required. On GSM (DOS) and GSM (Novell) it might be possible to use the VMM provided by an Extended Memory Manager. On GSM (BOS) this could only be implemented by resurrecting the defunct "Protected Mode" implementation. Note that any solution for GSM (BOS) would NOT be applicable to GSM (DOS) or GSM (Novell). For example, switching the CPU into real mode to access memory beyond the 1Mb real-mode limit would crash DOS, Windows 3.11 or Windows-95/98 etc. A conservative estimate for this project for either "J5" **or** "JW" is 12 weeks.

- Porting the 32-bit loader/resolver SVC, SVC-79, from C to Intel assembler (assuming the implementation of a VMM – see above) is also a huge project. Although the C compiler can be configured to generate Intel assembler code such assembler code would be only a starting point for the port to the GSM nucleus. A conservative estimate for this project is 16 weeks. Furthermore, the current "J5" and "JW" nuclei, which are confined to the real mode segmentation limit are already very close to the 64Kb boundary (indeed several projects have been implemented to move isolated modules from the "nucleus segment" to high-memory.

- The above comments for SVC 79 also apply to the Commercial Code Interpreter. A conservative effort to implement the 32-bit changes to the 16-bit CCI is 10 weeks.

- The changes to SVC 10, SVC 18, SVC 61 and the implementation of SVC 84 are less onerous but would still require 4 weeks effort.

Thus, a conservative estimate for the effort to port the 32-bit run-time components to the current GSM (Novell) nucleus is 42 weeks.

A radically different solution is required!

4. The Existing Gateway to Client interfaces

Before describing the proposed port of the 32-bit run-time software to GSM (Novell) a discussion of the various GSM client-server connections is germane. The interfaces are described in chronological order of implementation:

- The GSM (Novell) workstation connection to the GSM.NLM for "File Executive" file access uses low-level SPX/IPX. This protocol is the only one supported by NetWare at the assembler level;

- The GSM (Novell) workstation connection to the SPEEDBAS.NLM for "Speedbase Database" file access uses the high-level Transport Layer Interface (TLI). In theory, TLI

could be used with any low-level protocol but the implementation currently uses SPX/SPX-2;

- The GSM (Windows NT) client connection between GLOBAL.EXE and GLSERVER.EXE for "File Executive" file access uses the protocol-independent Remote Procedure Call (RPC) interface. This protocol, which supports TCP/IP, was chosen to be compliant with the Microsoft Back-Office program.

- The GSM (Windows NT) client connection between GLOBAL.EXE and SPEEDBAS.EXE (and SPEEDSQL.EXE) for "Speedbase Database" file access also uses the protocol-independent Remote Procedure Call (RPC) interface. This protocol was chosen to be compliant with the Microsoft Back-Office program.

5. The Proposed Solution

As described in section 4, we have a total of 3 different client-server interfaces for the GSM (Windows NT) and GSM (Novell) implementations. The proposed solution involves a effective merger of the GSM (Novell) and GSM (Windows NT) architectures and the implementation of a single client-server interface based on TCP/IP.

This solution will limit 32-bit GSM (Novell) to Windows-95 and Windows-98 workstations (i.e. workstations running DOS or Windows 3.1/3.11 will **NEVER** support the 32-bit run-time software). However, the proposed solution will allow workstations running Windows NT-4 Workstation side to run GSM on a Novell network.

The following software modules must be changed in order to implement the hybrid GSM (Windows NT) and GSM (Novell) architecture:

- The GSM.NLM must be modified to support both SPX **and** TCP/IP. The TCP/IP interface will be used to connect to "32-bit capable" workstations running architecture "W1" on Windows-95, Windows-98 or Windows NT. The "legacy" SPX interface will be used to connect on "16-bit only" workstations running DOS, Windows 3.11, 3.11, Windows-95 or Windows 98.

  Important Note: The availability of WinSock compatible TCP/IP on a NetWare server may be dependent on the version of NetWare. Further investigation will be required;

- The SPEEDBAS.NLM must be modified to support both TLI/SPX **and** TCP/IP. The TCP/IP interface will be used to connect to "32-bit capable" workstations running architecture "W1" on Windows-95, Windows-98 or Windows NT. The "legacy" TLI/SPX

interface will be used to connect on "16-bit only" workstations running DOS, Windows 3.11, 3.11, Windows-95 or Windows 98.

Important Note: The availability of WinSock compatible TCP/IP on a NetWare server may be dependent on the version of NetWare. Further investigation will be required;

- The "LAN executive" interface in the GLOBAL.EXE module must be enhanced to support TCP/IP (using the WinSock interface) in addition to the currently supported RPC interface. In the short-term, see below, the existing RPC interface will be used to connect to GLSERVER.EXE servers running on a Windows NT server; the new TCP/IP interface will be required to connect to the "TCP/IP aware" GSM.NLM running on a NetWare server;

- The "SVC-61 Gateway" interface in the GLOBAL.EXE module must be enhanced to support TCP/IP (using the WinSock interface) in addition to the currently supported RPC interface. In the short-term, see below, the existing RPC interface will be used to connect to SPEEDBAS.EXE servers running on a Windows NT server; the new TCP/IP interface will be required to connect to the "TCP/IP aware" SPEEDBAS.NLM running on a NetWare server.

In addition, the following software modules could be changed in order to implement the unified client-server interface:

- The GLSERVER.EXE Server could be modified to support both RPC and "raw" (Winsock) TCP/IP. If this change is implemented the "LAN Executive" code in GLOBAL.EXE could be simplified to use TCP/IP only;

- The SPEEDBAS.EXE Server could be modified to support both RPC and "raw" (Winsock) TCP/IP. If this change is implemented the "SVC 61 Gateway" code in GLOBAL.EXE could be simplified to use TCP/IP only.

The above changes will affect both the GSM (Novell) and GSM (Windows NT) distribution software. Both the hybrid NLM's will be distributed on the GSM (Novell) BACNAT volume. In addition, both the hybrid NLM's and the revised GSM (Windows NT) modules will be distributed on the GSM (Windows NT) BACNAT volumes. Thus, the GSM (Windows NT) software will operate on both Windows NT AND Novell NetWare networks.