Computer Science and Systems Analysis

Computer Science and Systems Analysis Technical Reports

Miami University *Year* 1991

System Programming - The Human and the Machine

Michael Kline
Miami University, commons-admin@lib.muohio.edu

MIAMI UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE
& SYSTEMS ANALYSIS

TECHNICAL REPORT: MU-SEAS-CSA-1991-004

System Programming - The Human and the Machine

Michael T. Kline

School of Engineering & Applied Science | Oxford, Ohio 45056 | 513-529-5928

System Programming - The Human and the Machine

by

Michael T. Kline

Systems Analysis Department
Miami University
Oxford, Ohio 45056

Working Paper #91-004 12/91

Systems Programming - The Human and the Machine

Final Report

In Partial Fulfillment of the Requirements for

The Masters Degree in Systems Analysis

from

Miami University

Oxford, Ohio

By:

Michael T. Kline
1012 North Maple Street
Eaton, Ohio 45320

1991

Review Committee:

Mr. Douglas Troy

Dr. Thomas Schaber

Dr. Alton Sanders

Systems Programming -
The Human and the Machine

Michael T. Kline
1012 North Maple Street
Eaton, Ohio 45320

1991

VM/MK

Table of Contents

Systems Programming - The Human and the Machine

1. Introduction

Systems programming is an occupation which is often misunderstood. Its most obvious definition might be "one who programs systems". However, it has come to have a more specific meaning. In today's computing environment a systems programmer is one who is responsible for the installation, customization, and maintenance of the software and hardware of a computer system that is necessary for application development to be done. According to IBM's Dictionary of Computing, a systems programmer is one who "plans, generates, maintains, extends, and controls the use of an operating system with the aim of improving overall productivity of an installation" [1]. This definition seems to indicate that a systems programmer might do everything on a system **except** the application development. One might assume that the systems programmer must possess many skills to do this and that the software products involved must be very flexible and well documented to allow the systems programmer to make all modifications needed.

The purpose of this paper is to document my experiences in planning, generating, and modifying the IBM VM/SP operating system (systems programming), survey literature on systems programming, and to draw conclusions as to what makes a successful systems programming experience. I will explore the skills necessary for the systems programmer to perform the tasks, as well as discuss aspects of the system itself (hardware, software, and documentation) that affect the success of any systems programming effort. This work is intended to serve as a case study of a VM/SP systems programmer working on VM/SP-compatible hardware. Judgments as to how these skills and conclusions may apply to other platforms are left to the reader.

Note that this paper contains many terms specific to IBM mainframes, the VM/SP operating system, and computer systems in general. Wherever appropriate, the definitions for such terms are provided in the text. In addition, a glossary of terms is provided in Appendix A of this report.

2. Background/History

This section presents background information necessary for this discussion including: (1) how the project originated, (2) history of the IBM hardware, and (3) history of IBM's operating system that I used.

2.1. This Project

The following sections discuss my interest in systems programming, how the project became a reality, and finally, how Miami would benefit from the project.

2.1.1. My Interest in Systems Programming

I graduated from Miami's undergraduate program in 1985 with a degree in systems analysis. I used my analysis and programming skills to secure a job as an application programmer and have worked in this area for the past six years. During this time I developed a curiosity about systems programming and have been interested in learning more about this area.

As a prospective graduate student to Miami's Systems Analysis program, I learned that Miami was in the process of applying for a gift from International Business Machines (IBM) that would include one or more small mainframes and that it would be possible for me to perform the systems programming for these systems (providing the gift was secured). I entered Miami's graduate program in August, 1990 with intent to do this work for my master's project.

2.1.2. Making the Project a Reality

Negotiations with IBM for the acquisition had begun in the spring of 1990 and continued throughout the remainder of the year. The primary questions being explored included what Miami could do with the machines and how it would benefit IBM. Once a plan had been agreed upon, contracts were approved in February of 1991 for two machines and various software. Miami would receive a 9375 and a 9371 mainframe, various terminals and printers, and almost any required software product.

In January, 1991, I was introduced to Mr. Michael Snyder, an account systems engineer for IBM, who would be available to teach me about the systems being received. I worked with him to learn about the hardware, systems software, and the basic skills necessary to prepare a software development laboratory for the Systems Analysis department. One could say that I apprenticed with him to learn what only his years of experience could offer.

It was decided that since I would be the first one to work with the new hardware, I would work primarily with the larger mainframe (the 9375) because this machine was scheduled to be used in courses in the fall semester. My systems programming efforts would begin in the spring, finishing up by the end of summer so that the machine could be used by classes for programming projects in the fall.

2.1.3. Miami's Interest in the Project

The Systems Analysis department saw several potential uses for the machines. First, they could provide a laboratory for graduate students to study aspects of IBM mainframes without impacting the mainframe serving the academic community of the university. Second, once the system was operational, the software development tools could be used by other graduate or undergraduate classes for programming projects. Finally, faculty could use the systems for research or simply for enhancing their own skills.

The goal of my work on this project was to do the necessary planning and systems programming to make one of the machines usable for the above purposes. My work on this project would not only satisfy the research requirement for my own degree, but also would leave the department with a fully operational system that could be used by other students and faculty.

2.2. IBM 9375 Mainframe

IBM's 9375 mainframe was developed at the Glendale Development Laboratory in Endicott, New York. The architecture of the machine is descendant from the IBM/370, popular since the 1970s. It is an air-cooled small mainframe that was intended to bring 370-architecture mainframes out of the corporate headquarters and into the branch office setting. The initial intent was to use this type of rack mounted processor in a distributed system connected to a large mainframe in a central location. Since then it has also been realized that this machine works well in a smaller company as its primary system.

The 9375's overall processing power is perhaps no more than ten percent of the larger 370 class mainframes, but it runs identical software. It provides this capability in an air-cooled, rack-mounted configuration, taking no more floor space than a large desk. Another important feature is that this class of machine does not require an environmentally controlled room. An office atmosphere with average temperature and humidity is perfectly acceptable.

The machine that Miami received was a 9375 model 50, providing 1.5 mips (million instructions per second) of mainframe power with 16 megabytes of main memory. Also included were an operator's console, eight 400 megabyte 9335 disk drives and one 9347 tape drive capable of handling standard one-half inch tapes at a 1600 bpi (bits per inch) capacity. Various adapter cards were installed including Ethernet and token-ring network boards, and work station adapters to connect terminals and printers. Also received were ten 3279 color terminals and a 3287 printer.

2.3. IBM VM/SP Operating System

The IBM VM/SP operating system is "a time sharing system control program that consists of: (a) a control program (CP) managing resources of an IBM System/370 computing system so that multiple remote terminal users have a functional simulation of a computing system (a virtual machine) at their disposal, and (b) the conversational monitor system (CMS), which provides general time sharing, program development, and problem solving facilities" [1].

VM/SP is most prevalent on small to medium sized systems as opposed to large systems which are dominated by IBM's MVS (Multiple Virtual Storage) operating system. One of VM/SP's useful features is the ability to serve as a host for one or more additional operating systems. For example, many installations that use the VSE (Virtual Storage Extended) operating system run it (or multiple copies of it) using VM/SP as the host.

VM/SP is descendant from the CP-40 operating system used in the mid-1960s. Successive generations of VM have included VM/370, VM/SP, VM/SP HPO, VM/XA, and most recently VM/ESA. See Figure 1 for a pictorial representation of VM's genealogy.

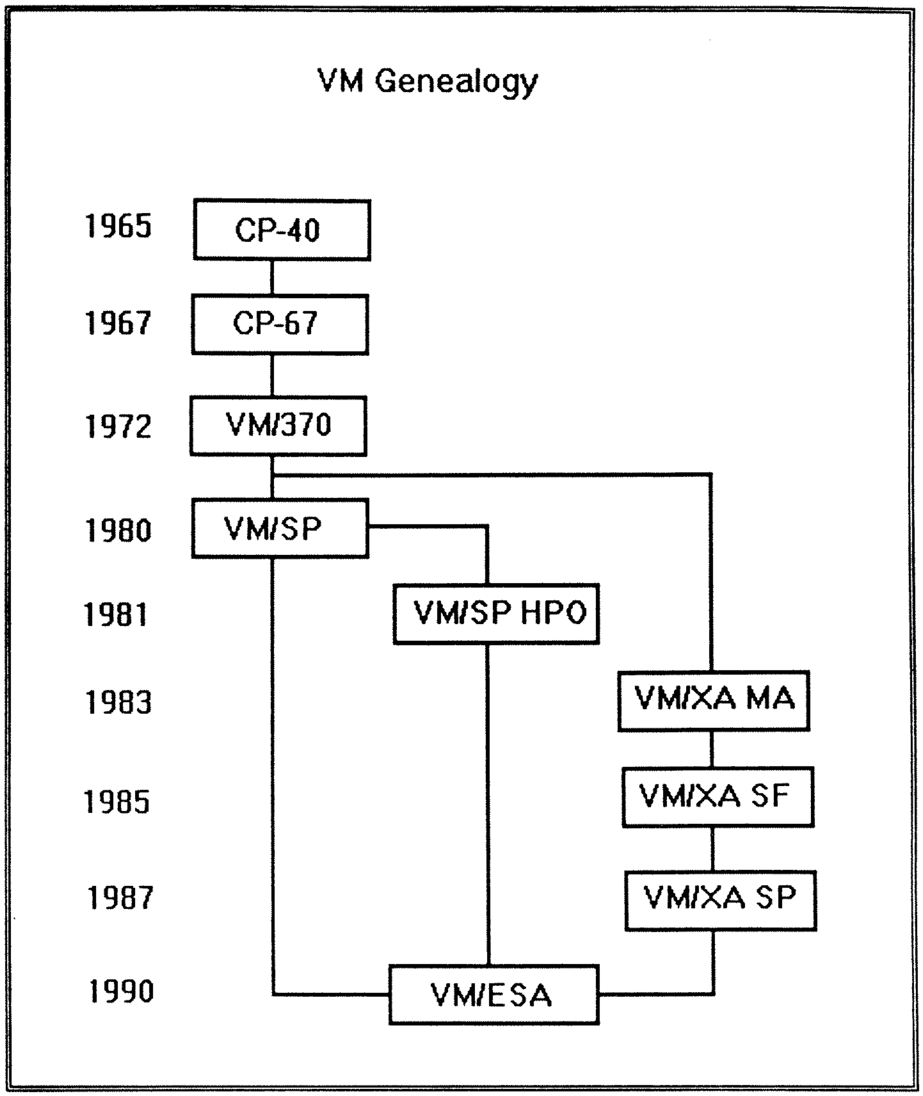

Figure 1 - VM Genealogy

Reproduced from IBM Systems Journal [4]

VM/SP has been the "work horse" version of VM for IBM. Due to its popularity and ease of availability, it was chosen for the Miami 9375 machine. Release 6 of VM/SP was received and installed on the system.

3. Implementing the System

Even though the 9375 was received on April 2, 1991, the job of implementation began several months before with planning and education. While doing this, I received and organized the system documentation. This was followed by the installation of the hardware and software.

3.1. Preparatory Training and Planning

My first task was to familiarize myself with the terminology, and learn the basic concepts that would help me understand the installation procedures. This was accomplished in several ways.

First, I read the VM/SP Planning Guide and Reference [2]. This provided me with an overview of VM/SP itself, a description of device configurations, a discussion of disk storage requirements for the system, and extensions to VM/SP.

For example, one of the main tasks described in the VM Planning Guide and Reference is to understand and modify the configuration files that have to be customized during the VM/SP installation. First was the VM/SP directory which contains a list of all potential virtual machines that may logon to the system. Second was a description of DMKRIO, which is a CP control file which consists of assembly language macros that describe real input/output devices, control units, and channels in the system. Third was information on DMKSYS, the control file consisting of information describing the CP system residence volume, system identification data, accounting parameters, dump space parameters, amount of real storage in the computer, and other miscellaneous items. Finally, this book described DMKSNT, the system name table that allows multiple users to share a single copy of a program and to IPL an already initialized shareable CMS nucleus.

The next method I used for acquiring training information was to have information sessions with the IBM systems engineer for the project, Mike Snyder. On several occasions from February through March I met with him to discuss hardware and software background, steps for a successful installation, system topology, and overviews of operating system structure and functionality.

Finally, just before beginning the system installation, I reviewed the VM/SP Installation Guide [3]. This manual would be the step-by-step guide to actually doing the VM/SP installation. A brief review of the steps involved proved to be helpful prior to beginning the work.

3.2. Documentation and Media Preparation

Foresight on the part of Mike Snyder brought the system software, documentation, and tapes to Miami a month before the hardware arrived. This gave me the opportunity to organize a nearly overwhelming amount of documentation, and made available to me some of the first manuals that I would study for background information. In total, more than thirty-four boxes of manuals and magnetic tapes were received. Specifically, nearly four hundred manuals, over one hundred tape reels, and over fifty tape cartridges were delivered for the systems. This information represented all hard copy and electronic information for both system and application software for both the 9375 and 9371 machines.

Step two of the media preparation was to organize and catalog all of these materials. First, I created a database of all manuals which included the following information: document number, version number, group name (product category), and document name. This information would be sorted once by group name and document name and again by document number. Next, all manuals were stored in lateral filing cabinets sequenced by group and document name. Third, I created a database of all tape reels and cartridges. These databases included the tape volume and the tape name. Again two cross reference listings were produced, one sorted by volume and one by tape name. The tape reels were then placed on a tape rack in the computer room in volume sequence. The tape cartridges remained in a box, also in the computer room.

In addition to manuals other documentation was received. All of this information is also filed with the manuals in the lateral filing cabinets. The following is a list of those items:

1. Technical News Letters - replacement pages for manuals reflecting enhancements or corrections.

2. "PUT" Tape Information - documents the contents of a "Program Update Tape", which contains preventative service, corrective service, and new hardware support for a product. This also provides the instructions for retrieving additional documentation from the tape.

3. Program Directories - list of product tape contents, related documentation, prerequisites, service fixes included with the product, resource requirements for the product, and additional instructions.

4. Licensed Program Information - provides a summary of product features, and a list of system requirements.

5. Microfiche Documentation - some manuals came on microfiche as well as normal hard copy.

6. Product Quick References - high level summary of product features and commands.

A copy of the document and tape cross reference listings mentioned above can be found in Appendices B, C, and D to this report. The cataloging and organization of these materials proved to be very valuable during the installation process as many of the manuals and all of the tapes were required to complete the job.

3.3. Hardware Installation

The majority of the hardware was delivered to Miami on April 2, 1991. This delivery included the mainframe racks (with the processor, all disk drives, the tape drive, and all adapter cards) and the system console. An IBM customer engineer came to Miami to do the hardware unpacking, install connections between the racks and the system console, test the hardware, and perform microcode verification. Prior to this, details on the electrical requirements had been given to Miami and the appropriate power connections had been installed.

The only remaining hardware installation required was to hook up the 3279 terminals and the 3287 printer that had also been received. Coaxial cable was run to the desired locations, connectors were attached to each end, and they were connected to the machine. Note however, that these terminals would not be recognized by the system until a tailored copy of VM/SP was IPLed. Only the system console would be used for the first of the installation procedures, but the 3279 terminals were ready when needed. Finally, an Ethernet local area network connection was made from the system to the university network. The necessary adapter card came with the machine from IBM. All that had to be added was a coaxial cable hookup, an Ethernet transceiver, and the appropriate software.

3.4. Software Installation

By April 4, 1991 the 9375 was ready for the system software installation. First, the VM/SP operating system would be built and customized. Later, system products such as compilers, database, and graphics packages would be added to the system. The information provided below represents a summary of the major steps required to install each component. Problems encountered during the process are mentioned in a later section of the report.

3.4.1. VM/SP

The major steps required for the VM/SP release 6 installation are identified below. The primary guide for these activities was the VM/SP Installation Guide [3]. The following discussion will highlight each milestone in the VM/SP installation and customization process. A step-by-step summary of my activities can be found in Appendix E. I have broken the process down into the following areas: (1) preparation, (2) using the starter system, (3) customizing CP, (4) customizing CMS, and (5) installing other VM/SP components.

The first part of the VM/SP installation can be described as preparation. Disk packs must be initialized and formatted in order to store data. Also, these packs should be labeled so that each is uniquely identified.

Next, the VM/SP starter system must be loaded, executed, and used to load additional tools. A simplified version of VM/SP is included on the installation tapes. This system must be loaded onto the disk packs and then executed by IPLing the system. It is then used to load additional tools and files for building VM/SP.

System generation tools and files are loaded using one of the VM/SP starter system utilities. Additional system userids must be defined to the system, and assembly language configuration files must be modified. These files specify hardware devices on the system, tell CP where to store paging and dump data, and identifies other named system areas. A new CP nucleus will then be built and IPLed.

Fourth, additional system disk volumes must be allocated and formatted. CMS files are loaded and customized, and the CMS system is built and executed.

Finally, other components of VM/SP can be installed. This includes problem tracking tools, network access software, remote database access, and the help system. Also, system verification procedures should be executed to ensure that all components are working properly.

3.4.2. VM/SP Products

After two months of part time work the VM/SP system had been installed and customized for the 9375. Next came the task of installing other support products such as compilers, database, and graphics packages. See Figure 2 for a list of software packages installed on the system.

Assembler H	PROFS
CICS	QMF
COBOL II	RPG II
CSP	RSCS
FORTRAN	SDF
GDDM	Sort/Merge
ISPF and IPF	SQL/DS
PASCAL	TCP/IP
PL/I	VTAM

Figure 2 - VM/SP Software products installed on IBM 9375

A fairly standard procedure is established for installing each product. However, each and every one presented unique challenges and never quite seemed to follow the installation instructions provided. Implementation problems are discussed in a later section of this report.

The typical installation procedure for a VM/SP product is as follows: (1) locate installation guide and all tapes for product, (2) define a minidisk to store the product files, (3) use the installation tool INSTFPP to load the contents of the tape to the specified minidisk, (4) customize ISPF screens and product execs so that the product can be easily invoked, and (5) test the product using sample files provided.

As shown in Figure 2, the TCP/IP networking software was installed on the system. As a result, the 9375 computer is accessible via the Ethernet network by using the TELNET command. The address for the system is "san9375.aps.muohio.edu".

3.5. Implementation Problems

During the installation of VM/SP and the VM/SP products a number of difficulties were encountered. The following is a description of the types of problems confronted and their resolution.

3.5.1. Service

The implementation of RSCS provided the first real problem with the system installation. RSCS (the Remote Spooling Communication Subsystem) is used to provide file handling across a network, special handling of devices not channel-attached to the system, and for printing spooled output.

The primary reason for installing RSCS on the 9375 was to provide control for the 3287 printer that was attached to the system. RSCS works with the GCS (Group Control System) component of VM/SP, which allows VM/SP to communicate across a network attached to the system. RSCS is thus able to perform its functions across a GCS controlled network.

After the RSCS had been installed on disk, the installation procedure stated that RSCS virtual machine should be started. RSCS is known as a service virtual machine. This means that when the VM/SP system is IPLed, this userid should be automatically logged on (autologged). It is in this state that RSCS will perform its prescribed functions.

It was during the autolog of RSCS that the problem was revealed. Immediately after being started in this fashion, RSCS would abend. A message was displayed on the operator's console indicating that this had happened. After reinstalling both GCS and RSCS, reviewing the directory entry for RSCS, and attempting other customization the problem persisted. Mike Snyder told me the only recourse was to contact the IBM Support Center for assistance.

The IBM Support Center is accessible via a toll-free telephone number 24 hours a day. Assistance is classified as "level 1" or "level 2". When calling for software support or installation service, a level 1 representative is consulted first. After describing a problem to the representative, he/she will search an IBM problem database to see if a similar issue has been recorded and solved. If so, the level 1 representative will discuss the resolution and/or send a fix on a tape. If level 1 is unable to close the problem, then the appropriate information will be conveyed to a level 2 representative who will evaluate the information and call back to discuss the issue and perhaps provide a customized fix.

The RSCS problem encountered on the 9375 turned out to be one that was previously discovered and addressed by level 1 support and could be resolved by applying several PTFs (program temporary fixes) to the system. It was recommended that the entire system be brought up to a higher service level. By doing this all system components (not just GCS and RSCS) would have service fixes applied, and the particular fixes for the GCS problem were included. This was accomplished by using the IBM PASO (Pre-

Applied Service Offering), which had been shipped along with VM/SP initially.

Installing the PASO and bringing the system up to a higher "service level" required a lengthy application of service and a complete reinstallation of the VM/SP system. Not all problems require this level of effort to fix, but this was the best way to resolve the GCS/RSCS problem being experienced.

3.5.2. EXEC Errors

Another frustrating problem encountered several times was errors in EXEC utilities to aid in software installation.

An EXEC is a simple programming language which allows someone to create a user-defined command. This facility, by providing a control structure and the ability to issue normal CP and CMS commands, allows users or software developers (or IBM itself) to perform functions that were not foreseen in CMS or to execute a collection of frequently used commands, thus hiding complicated features [5].

On several occasions the EXECs provided with the system were unable to complete the installation of the software product. In each case the EXEC had to be debugged and fixed appropriately.

4. Survey of Literature on Systems Programming

In an attempt to supplement the information that I have compiled during this project, I searched for literature that had been written on the subject of systems programming. Specifically, I hoped to find information documenting the experience of others working with VM/SP and VM/SP products. A thorough search of Miami's library revealed that little information on systems programming was available. Additionally, I found no articles at all discussing VM/SP installation.

I discussed this finding with Mike Snyder and have come to the following conclusions. First, literature discussing systems programming is probably confined primarily to documents which are not well indexed and thus hard to locate. For example, vendor training manuals or course materials. Second, information on systems programming might be contained in less formal publications or not be published at all. For instance, user group newsletters and meeting minutes might be a more common medium, and a personal network of individuals communicating among themselves could be a better way of getting more current and specific information. Third, due to the vast array of hardware configurations, different software products, numerous software versions, and customized variations, it is very difficult to produce meaningful information for a large audience. An interesting extension to this project would be to investigate these conclusions with practicing systems programmers and reveal the sources of their information.

5. Analysis of My Systems Programming Experience

I feel that my brief experience with systems programming has provided me with some insights into what elements can improve (or perhaps ensure the success of) a systems programming experience. I have organized my thoughts into two areas. First, I will present issues regarding "the human". This will investigate what skills "the human" needs to do the job of systems programming. Second, I will present issues regarding "the machine" (hardware, software, and documentation). This section will explore issues that could be addressed by the provider of the system (IBM) that might make the systems programmer's job easier.

5.1. "The Human" - Hierarchy of Skills

While waiting for the hardware and software to arrive, I was initially concerned with the following questions:

- What skills are needed to install the system software ?

- What can I read to get an overview of this process ?

- What planning can be done to make the process easier ?

- Are there any training materials for systems programmers ?

After the reading I had done and the conversations I had with Mike Snyder I began to realize that it is difficult to answer these and other similar questions. The answers I was seeking would be provided during the systems programming project rather than while preparing for it.

Thus, I began my systems programming experience with only a user's perspective on the system (from using other VM systems at Miami), and "bits and pieces" of information gathered from reference manuals and meetings with the IBM SE, Mike Snyder.

I was able to complete the system software installation after much reading of system manuals, many trial and error approaches to installation and customization issues, and with helpful guidance from the experience of Mike Snyder.

As a result of this process, I have identified six levels of skills that would help to make a systems programming project successful. These skill levels reflect the increasing difficulty in system implementation as more products are added and additional integration is required. The following sections describe these skill levels which are also shown in Figure 3.

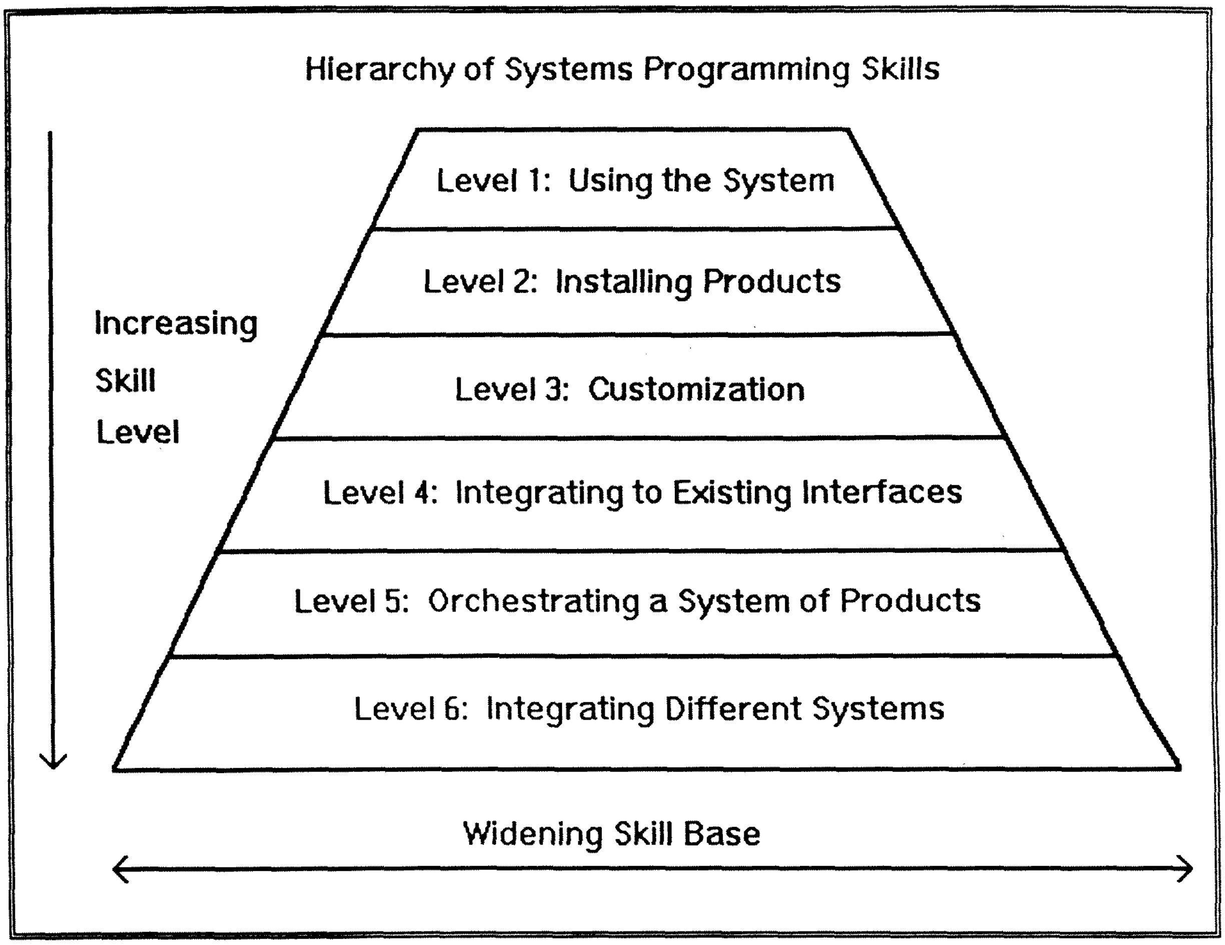
Hierarchy of Systems Programming Skills
Level 1: Using the System
Level 2: Installing Products
Level 3: Customization
Level 4: Integrating to Existing Interfaces
Level 5: Orchestrating a System of Products
Level 6: Integrating Different Systems
Increasing
Skill
Level
Widening Skill Base

Figure 3 - Hierarchy of Systems Programming Skills

5.1.1. Level 1 - Using the System

The first and most basic ability required by a VM systems programmer is knowledge of fundamental VM and CMS user skills. Having the ability to use a system similar to the one being installed proves to be an advantage during the installation. This gives the installer an appreciation for what the end result of the job will be. Also, these skills will be necessary in accomplishing the installation itself.

For this and the other skill levels, I will use an analogy of a construction worker to help describe the systems programmer's skills. The level 1 skills are perhaps comparable to the most basic of construction skills. For example, knowing that a house must have doors for entry and that the kitchen is where meals are prepared are level 1 skills for the house builder.

The skills for the systems programmer include logging on to the system; listing, editing, saving, copying, and deleting files; and knowing how to reveal the minidisk structure and utilization.

These skills can be obtained through normal training and reference materials designed to introduce someone to the VM/SP environment. However, years of experience using a VM system from a user's point of view is probably the most common and best way of obtaining this education.

Skill level 1 is assumed by all of the VM/SP installation materials I studied, but alone is insufficient to accomplish even the first step of the VM/SP installation.

5.1.2. Level 2 - Installing Products

The systems programmer also must develop skills directly pertinent to installing specific packages. This will facilitate the quick and easy addition of a new software product to a new or existing system. When a new version of an operating system or product is received, one might receive a box with a tape and a few manuals. Level 2 skills allow the systems programmer to get the package onto the system in a timely fashion.

To continue the analogy to the house builder, the construction worker knows that there are special procedures for installing a water heater or a furnace in a home. The overall house design should accommodate these items, but extra knowledge must be acquired to insure a successful installation of the item.

The skills in this level for the systems programmer include mounting tapes, allocating disk space for files, understanding and interpreting EXECs, and IPLing the system.

Once again, most of the knowledge at this level is assumed by VM/SP installation materials. There appear to be few direct sources for this information. I was most successful obtaining this knowledge from the IBM SE on an as-needed basis. Skill levels 1 and 2 will allow someone to install VM/SP or a VM/SP package onto a machine using predefined defaults and without regard for customization or integration among products.

5.1.3. Level 3 - Customizing

The next skill a systems programmer needs is the ability to change a product for site specific requirements or to take advantage of some optional aspect of a product. To accommodate as many customers as possible, software products or an operating system will install with a standard configuration and allow the installer to request specific features later.

The house builder, for example, may receive a central air conditioner to install in a home. The compressor and evaporator are delivered already assembled and operational. However, locating the evaporator, electrical wiring, and duct work are things that must be performed by the builder. These things allow the air conditioner to be customized or tailored for a specific house.

The customization skills that the systems programmer needs include updating assembly language configuration files, modifying product start-up EXECs, and adding information for hardware devices specific to a particular system.

Fortunately these skills, although more advanced than level 2, can be more easily learned. Normal installation documentation that accompanies a software package almost always includes extensive information on customization. Some aspects of customization are product specific, however, certain procedures are similar for many products. Learning these common customization skills can make the installation process much easier for future products.

5.1.4. Level 4 - Adding into Existing Interfaces

The next skill for the systems programmer is to learn how to integrate products into existing interfaces in the system. In order for a system to have a consistent "look and feel" it is helpful for each product to be invoked in a similar fashion. Level 4 skills allow the systems programmer to do this for a particular installation.

The construction worker must consider how to connect the house water, sewer, telephone, and electrical systems into existing city facilities. Extra piping or an additional pole might have to be installed to accommodate what is provided. In extreme instances a septic system or a well might have to be dug. The basic capabilities in the house, however, are unaffected by these issues.

The skills in this level for the systems programmer might include writing EXECs to link and attach the minidisks and establish the appropriate globals or filedefs. Another common interface used for products is ISPF. In this case, screens would have to be built and libraries accessed.

Knowledge for this kind of work seems to come from two sources. First, for writing EXECs, reference material exists which is useful for identifying specific capabilities and syntax. However, I feel that EXEC writing skills are primarily developed from studying existing EXECs. Perhaps while attempting to understand an EXECs functionality, or while debugging problems, anyone that works with VM very long will begin to acquire this skill. Secondly, when an interface such as ISPF is used to invoke products, skills can be obtained from the documentation on the interface itself. In my case, the ISPF manuals not only described what the product offered by default, but also directed me to areas that would have to be changed to add new products to the interface.

5.1.5. Level 5 - Orchestrating a System of Products

Skill level 5 for a systems programmer is having the ability to orchestrate a system of products under a particular operating system. Rarely will an organization want a group of application software packages and then use each of them independently of the others. The systems programmer must identify which products need to work together, and then determine how to accomplish this.

A security system is a good example of how a house builder might have to make different parts of a home work together. First, the security system must receive inputs from a variety of sources. Doors and windows must convey their status and a control panel must provide user inputs. The system must also output information to the house. If an entry is detected with the system engaged, then lights must be activated, alarms must ring, and perhaps even an automatic telephone call placed.

The skills for the systems programmer to orchestrate different products also tend to involve inputs and outputs. A CSP program developed in-house might need the ability to retrieve data from a SQL database. Similarly, determining how in-line SQL queries or CICS instructions in a COBOL program get translated into the appropriate calls is important. If a user wants to be able to take data from a COBOL program and use it as input for a GDDM-developed graph, the systems programmer must be able to accommodate this as well.

In order to develop these types of skills two things must be considered. First, the applications involved must be well understood. The systems programmer must be able to determine what data types and formats will be provided and perhaps these will have to be changed. The software on the receiving end must be considered. Data requirements are again an important issue.

If SQL or CICS is involved, an analysis of a preprocessor will also be required. The second thing desired of the systems programmer is determining how to change the system so that the two products involved can communicate. For example, one might evaluate changes so that a preprocessor will be automatically invoked to translate the SQL or CICS commands. Documentation on the products involved will assist the systems programmer, but it is his/her own experience and creativity that will determine the success of the project.

5.1.6. Level 6 - Integrating Different Systems

The last and most difficult skills for a systems programmer involve integrating systems of different types. Whether it involves two operating systems running on the same machine (perhaps using VM as a host) or systems running on different machines, this type of integration is sure to provide a challenge.

My own systems programming project provided me with minimal experience at this level when I installed VSE/SP using VM/SP as a host. This required modification of the VM/SP user directory to establish a virtual machine for the guest operating system (VSE/SP). I worked with this configuration only briefly having concentrated the majority of my project on VM/SP.

My expectation is that, similar to level 5, many of the integration issues involve inputs and outputs between machines. Additional complications are sure to exist. I would expect that EBCDIC to ASCII character translation, differing file structures, protocol conversions for communications over a network, and timing considerations could be factors.

Once again, experience is the best teacher. The systems programmer must be his/her own best reference. Starting with documentation for the products involved and supplementing this with common sense and the wisdom of others is probably the best way to succeed in this situation.

5.2. "The Machine"

Having explored the skill levels required (or desired) by "the human", I will now turn my attention to "the machine". Please note that my reference to "the machine" refers to any of the family of machines capable of running the VM/SP operating system. My systems programming project has revealed several things which I feel could be done to the system itself to improve the VM systems programmer's job.

My use of the term "the machine" refers primarily to the software used on the computer, the operating system itself, and the software used to install the operating system and other products. Additionally, any hardware or documentation issues will be treated as part of "the machine". There are several aspects to system design that can have a direct impact upon the ease of the systems programming experience. A discussion of these issues follows.

5.2.1. Software Status

The first major problem encountered with the VM/SP installation was the GCS/RSCS problem described above in the implementation problems section. RSCS would abend upon start-up when executed with the level of GCS initially installed. The PTFs required to fix the problem were included in the PASO (Pre-Applied Service Offering) tape included with the VM/SP installation package, but I feel that the installed version of VM/SP should have been at the current service level initially. None of the documentation included with VM/SP indicated that it was important or even recommended to immediately apply any service.

Sending a version of VM/SP with the service already applied, or at least notification that application of the service was needed would have saved much time and frustration, and a call to the IBM support center to remedy a problem that should have already been resolved.

5.2.2. Help

The next area where "the machine" could be improved is the help system. These improvements can be broken down into two areas -- help for normal installation activities, and better help in the event of abnormal conditions.

First, having more on-line help available to assist the systems programmer during installation activities would be beneficial. It is always reassuring to know that in the event of indecision or confusion that useful information is no more than a keystroke (PF1) away. Additionally, help information provided in this way could be distributed right along with the product being installed and thus probably be more current than printed documentation.

Second, even more useful but perhaps more difficult to provide, is assistance in the event of abnormal termination of a installation activity or program product. Any information that might help the systems programmer to diagnose the problem and avoid a call to the support center would save time and effort.

5.2.3. Automated Installation Tools

Next, I feel that installation tools could be improved in two ways. First, these tools should be more automated. Considering the fact that the installation procedure is similar for most program products, a single utility that comes with VM should be used to install them. Mike Snyder told me that the command INSTFPP was an attempt to do something like this, but for many of the products that I installed it would not work. Second, on several occasions I encountered installation EXECs that would not work. In each case the install EXEC had to be manually debugged before the product was installed. Quality control for these EXECs should be improved.

5.2.4. Improved Documentation

The final area where I feel "the machine" could be improved is documentation.

First, when the documentation for VM/SP and the VM/SP products arrived at Miami it arrived in 34 separate boxes with numerous packing slips. To summarize, over 400 manuals and 100 tapes were included in what appeared to be random order. Some type of comprehensive list of all materials should have been included, and, more importantly, some guide for grouping related materials was needed. Even better, related manuals should have been packed in the same boxes.

Second, after using many of the manuals on a regular basis, a comprehensive index would be helpful. Even if this were provided on a product level it would help make sense (and better use) of the massive amount of documentation accompanying the system.

Third, of all the manuals received, very little of it could be described as training materials. Reference manuals are useful, if not vital, once someone knows how to use a product. Overcoming the initial hurdle of learning to use a product could be much easier if more training materials were included.

6. Conclusion

In this paper I have recounted the activities of my graduate research project and have attempted to explore the field of systems programming. My two main goals for this work were to (1) prepare a system that could be used for a variety of tasks in the Systems Analysis department and (2) learn about systems programming to satisfy my own curiosity and interest in this area. The following sections comment on these two areas in addition to some suggestions for future work in this field.

6.1. Project Status

It was hoped that this project would be more than just an exercise that would serve the short term goals of my research work. Instead it was my hope that I could take the IBM equipment and make it into a software laboratory that could be used productively within the department. Specifically, it was desired to provide a setting where graduate and undergraduate students and faculty members could work to perform projects and improve their skills. I feel that I have accomplished this. Virtually every piece of software received with the system has been installed and is operational. All software may not represent the cutting edge of computer development, but it certainly demonstrates many of the software products and techniques commonly used in business today and could be used by students and faculty alike for many purposes.

Few systems are static and I am sure that shortly there will be requirements for the computer that can not presently be satisfied. Part time technical support will be required to install new products, upgrade existing ones, and provide administrative support just like all systems of any size.

Perhaps the biggest obstacle for using the new system is training. A plan needs to be developed to help both faculty and students become familiar with what the system offers and how to use the products available. However, I am quite satisfied that the system can benefit the Systems Analysis department with additional effort in these areas.

6.2. Project Follow-up and Further Studies

It has become evident to me that the systems programming field is large in scope and the study of it can be pursued in many areas. I have touched upon some of these areas in this paper that I was unable to cover at length due to their scale. The following is a list of some ideas for additional projects or further study that could be conducted by fellow graduate students or interested faculty members.

- Research ways to develop better on-line help, training materials, and tutorials for the systems programmer to use to learn and improve skills in systems programming.

- Study the body of literature written about the systems programming discipline.

- Develop shell applications on the 9375 system which are incomplete. Allow students to finish development concentrating on areas where different products and programming concepts (CICS, SQL, CSP, etc.) can be learned.

6.3. Systems Programming

I believe that I, an applications programmer, can now better appreciate the work required to make a mainframe system operational and what is required of the systems programmer to maintain it. Hopefully my analysis of the skill levels for a systems programmer might help others considering such a vocation. Also, perhaps vendors (especially IBM) could benefit from my suggestions regarding "the machine".

As a result of my efforts on this project I feel that, on my hierarchy of skills, I have achieved level 3 (customization) and was beginning to work at level 4 (integrating to existing interfaces) when I completed my project work. I believe that if the time and place are right, that I would enjoy a chance to exercise my abilities as a systems programmer in a "real" (business) environment. Additionally, the experience I have gained should prove to be valuable when attempting to earn such a position.

Ultimately the job of the systems programmer comes down to balancing the bad with the good. I have seen that the systems programmer's job can be a very frustrating one. He/she is always the one who, if things run smoothly, might receive little notice, but as soon as a problem arises, all attention very quickly turns his/her way. The job requires a resourceful person, able to deal with a gamut of problems on a moment's notice, fixing them with perhaps no more than a "trial and error" approach.

However, few things can compare to the rewarding feeling of building a system capable of doing so many things--venerable applications and new ones being developed every day. Indeed, to be able to sit down at nearly any terminal in the campus network and with a few keystrokes see my personalized VM logo ("VM/MK") appear is very satisfying.

REFERENCES

1. Dictionary of Computing, International Business Machines Corporation, March 1987.

2. Virtual Machine/System Product Planning Guide and Reference Release 6, International Business Machines Corporation, August 1988.

3. Virtual Machine/System Product Installation Guide Release 6, International Business Machines Corporation, August 1988.

4. Fischofer, W. T., "VM/ESA: A single system for centralized and distributed computing", IBM Systems Journal, volume 30, number 1, 1991.

5. McClain, Gary R., ed., VM Applications Handbook, Intertext Publications, McGraw-Hill Book Company, 1989.

ACKNOWLEDGMENTS

Doug Troy

First and foremost I would like to thank my advisor, Doug Troy, for all his efforts in making this project a success. For more than a year he has offered his support and guidance while I have planned, performed, and documented my project activities. He has recognized the scheduling difficulties I have had while working full time and repeatedly has shown flexibility while dealing with me.

Al Sanders and Tom Schaber

Next I would like to recognize Al Sanders and Tom Schaber, the other two members of my review committee. Their support and trust shown to me from the beginning is appreciated. I hope that the system I have put together will assist them in strengthening the programs in the Systems Analysis department.

Mike Snyder

I especially wish to acknowledge the extraordinary assistance provided by Mike Snyder. I am indebted to him for his ongoing technical support both before and during this project. On many occasions he has gone beyond the call of duty, working late into the evening, teaching me the subtleties of VM/SP and struggling side by side with me against its difficulties.

International Business Machines (IBM)

Finally, I want to thank the individuals in the IBM Corporation who were instrumental in approving the hardware and software loan to Miami University. Simply stated, this project would not have been possible without this equipment, your cooperation, and the technical assistance provided by your company. I feel that I have greatly benefitted from this experience and the efforts of everyone who contributed is appreciated.

Appendix A - Glossary of Terms

The following is a list of terms used in this report. Someone familiar with VM/SP systems would probably know them, but perhaps all readers will not. Many of the terms are abbreviations; if so, the expansion of the abbreviation is given followed by the definition.

AVS - APPC/VM VTAM - component of VM/SP that provides the capability for program to program communications across a VTAM controlled network in a host-server relationship.

CICS - Customer Information Control System - a general purpose program that controls on-line communication between terminal users and application programs.

CMS - Conversational Monitor System - a single user operating system specifically designed to run in a virtual machine created and managed by CP. Provides general interactive time sharing and program development capabilities [1].

CP - Control Program - the component of VM that manages the resources of a single computer with the result that multiple computing systems appear to exist. Each virtual machine is the functional equivalent of an IBM System/370 computer [1].

CSP - Cross System Product - a high level programming language specifically designed to be transportable across different computing platforms.

DDR - DASD Dump Restore - a CP software program that allows data to be transferred from disk to tape and vice versa [5].

DMKRIO - an assembly language program module that is part of the CP nucleus. It defines the real input/output devices and their addresses [5].

DMKSNT - an assembly language program module that is part of the CP nucleus. It defines the system name table, which identifies named shared segments and named saved systems. These components are essentially common copies of frequently used programs to save memory utilization [5].

DMKSYS - an assembly language program module that is part of the CP nucleus. It contains system definitions, which specifies the location of data which are critical to CP's operation such as system paging and dump areas [5].

DMSNGP - an assembly language program module that is part of the CMS nucleus. It defines the specific characteristics of the

CMS system such as important disk addresses and the name of the CMS saved system.

DSF - Device Support Facility - a CP software program that performs operations on disk volumes so that they can be accessed by programs. These operations include initializing a disk volume and assigning alternate tracks [1].

EREP - Environmental Recording, Editing and Printing - a program that reads and formats for analysis input/output error records written to a special DASD data set by the operating system [5].

Ethernet - local area network architecture originally developed by the Xerox Corporation. Access to such a network is governed by the TCP/IP protocol suite.

EXEC - a CMS function that allows users to create new commands by setting up frequently used sequences of CP commands, CMS commands, or both, together with conditional branching facilities, into special procedures to eliminate the repetitious retyping of those commands sequences [1].

GCS - Group Control System - a component of VM which allows it to communicate with network software.

GDDM - Graphical Data Display Manager - a group of routines that allows pictures to be defined and displayed procedurally through function routines that correspond to graphic primitives [1].

INSTFPP - VM command that can be used to load the contents of a product installation tape to a minidisk.

IPCS - Interactive Problem Control System - a component of VM that permits on-line problem management, interactive problem diagnosis, on-line debugging for disk-resident CP abend dumps, problem tracking, and problem reporting [1].

IPF - Interactive Productivity Facility - an IBM program product consisting of system function screen panels and utility programs which are useful in building service virtual machines [5].

IPL - Initial Program Load - the initialization procedure that causes an operating system to commence operation [1].

ISPF - Interactive System Productivity Facility - IBM program product that runs on multiple platforms which provides an interactive set of screen panels and utility programs to simplify program development and file management.

ITASK - VM command used to load and build the VM operating system and other operating system components.

IVP - Installation Verification Procedure - a procedure distributed with the VM system that tests the newly generated VM system to verify that the basic facilities of VM are functioning correctly [1].

Microcode - a code, representing the instructions of an instruction set, that is implemented in a part of storage that is not program addressable. Microcode represents microinstructions used in a product as an alternative to hard-wired circuitry to implement functions of a processor or other system component [1].

Minidisk - term describing a unit of allocated DASD space. A minidisk is a logical, or virtual, area of physical DASD assigned in either cylinders or blocks [5].

Mips - Millions of Instructions Per Second - a measure of the speed of a CPU.

PROFS - Professional Office System - an IBM program product that provides in interactive computer-assisted office functions such as electronic mail, document storage and retrieval, and time management [5].

PTF - Program Temporary Fix - a temporary solution or by-pass of a problem diagnosed by IBM as a result of a defect in a current unaltered release of the program [1].

PUT - Program Update Tape - tape that is periodically sent by IBM to VM installations containing recent fixes to the operating system or other application software.

QMF - Query Management Facility - IBM program product that provides an environment for building and executing database queries.

RSCS - Remote Spooling Communications Subsystem - the licensed program that transfers spool files, commands, and messages between VM users, remote stations, and remote and local batch systems through telecommunications facilities [1].

SDF - Screen Definition Facility - IBM program product that provides an environment for designing screens to be used by CICS, ISPF, etc. SDF will then build the necessary code to make the screen available to the program displaying it.

Service Virtual Machine - a virtual machine providing operating system functions such as security, automatic operation, terminal protocol conversion, accounting and error recording outside the operating system itself [5].

SQL/DS - Structured Query Language / Data System - program product implementing a standard query language (originally defined by IBM) that is used with relational database management systems [5].

TCP/IP - Transmission Control Protocol / Internet Protocol - IBM program product used in interfacing with an Ethernet network.

Token-ring - a ring network that allows unidirectional data transmission between data stations by a token passing procedure over one transmission medium so that the transmitted data returns to the transmitting station. A network that uses a ring topology, in which tokens are passed in a circuit from node to node. A node that is ready to send can capture the token and insert data for transmission [1].

TSAF - Transparent Service Access Facility - a CP service that provides networking of up to eight processor complexes for the exchange of data between applications (especially database applications) [5].

Virtual Machine - a functional simulation of a computer and its associated devices. Each virtual machine is controlled by a suitable operating system (for example, CMS). VM controls concurrent execution of multiple virtual machines on a single system/370 computer [1].

VMFPLC2 - VM command that can be used to load and unload the contents of tapes.

VM/SP - Virtual Machine / System Program - a time sharing system control program that consists of: (a) a control program (CP) managing resources of an IBM System/370 computing system so that multiple remote terminal users have a functional simulation of a computing system (a virtual machine) at their disposal, and (b) the conversational monitor system (CMS), which provides general time sharing, program development, and problem solving facilities [1].

VTAM - Virtual Telecommunications Access Method - generic name for a series of program products for each of the major IBM operating systems (MVS, VM, and VSE). VTAM both manages the connectivity of a processor complex and provides an interface for applications to communicate with devices and other applications in a teleprocessing environment [5].

Appendix B - Documentation

The following is a list of all IBM 9371/9375 software documentation received with the systems:

Seq	Doc. number	Ver	Group	Document name
1	GC26-4035-1	1	ASSEMBLER	ASSEMBLER H VERSION 2 GENERAL INFORMATION
2	SC26-4030-2	1	ASSEMBLER	ASSEMBLER H VERSION 2 INSTALLATION
3	GC26-4037-1	1	ASSEMBLER	ASSEMBLER H VERSION 2 LANGUAGE REFERENCE
4	LY26-3908-0	1	ASSEMBLER	ASSEMBLER H VERSION 2 LOGIC
5	SC26-4036-2	1	ASSEMBLER	ASSEMBLER H VERSION 2 PROGRAMMING GUIDE
6	LY27-8030-0		BTAM	BTAM EXTENDED SUPPORT (BTAM-ES) LOGIC
7	SC33-0077-5	1.7	CICS	CICS APPLICATION PROGRAMMER'S REFERENCE MANUAL (COMMAND LEVEL)
8	SC33-0079-5	1.7	CICS	CICS APPLICATION PROGRAMMER'S REFERENCE MANUAL (MACRO LEVEL)
9	SC33-0080-4	1.7	CICS	CICS CICS-SUPPLIED TRANSACTIONS
10	SC33-0139-0	1.6	CICS	CICS COBOL SOURCE CODE FOR THE EXAMPLE APPLICATION
11	SC33-0131-2	1.7	CICS	CICS CUSTOMIZATION GUIDE
12	LY33-6033-3		CICS	CICS DATA AREAS
13	LC33-0105-2		CICS	CICS DIAGNOSIS REFERENCE
14	SC33-0228-0	1.7	CICS	CICS FACILITIES AND PLANNING GUIDE
15	SC33-0096-2	1.7	CICS	CICS IBM 3270 DATA STREAM DEVICE GUIDE
16	SC33-0073-3	1.7	CICS	CICS IBM 3650/3680 GUIDE
17	SC33-0074-03	1.7	CICS	CICS IBM 3767/3770/6670 GUIDE
18	SC33-0075-5	1.7	CICS	CICS IBM 3790/3730/8100 GUIDE
19	SC33-0072-3	1.7	CICS	CICS IBM 4700/3600/3630 GUIDE
20	SC33-0070-6	1.7	CICS	CICS INSTALLATION AND OPERATION GUIDE
21	SC33-0133-2	1.7	CICS	CICS INTER-COMMUNICATION FACILITIES GUIDE
22	GC33-0356-2		CICS	CICS LIBRARY GUIDE
23	SC33-0139-0	1.6	CICS	CICS MAIN INSTRUCTIONAL TEXT
24	SC33-0219-01	1.7	CICS	CICS PERFORMANCE DATA
25	SC33-0134-2	1.7	CICS	CICS PERFORMANCE GUIDE
26	SC33-0089-3	1.7	CICS	CICS PROBLEM DETERMINATION GUIDE
27	SX33-6010-05		CICS	CICS PROGRAM DEBUGGING REFERENCE SUMMARY
28	SC33-0135-2	1.7	CICS	CICS RECOVERY AND RESTART GUIDE
29	GC33-0130-02	1.7	CICS	CICS RELEASE GUIDE
30	LC33-0438-0		CICS	CICS REMOTE SERVER DIAGNOSIS
31	SC33-0382-00	1.7	CICS	CICS REPORT CONTROLLER USER'S GUIDE
32	SC33-0149-3	1.7	CICS	CICS RESOURCE DEFINITION (MACRO)
33	SC33-0238-0	1.7	CICS	CICS RESOURCE DEFINITION (ON-LINE)
34	SH19-6629-0		CICS	CICS VSAM RECOVERY/VSE GUIDE
35	SC33-0695-0	1	CICS	CICS/DDM RELEASE 1.0 USER'S GUIDE
36	SC33-0085-2	1.6	CICS	CICS/DOS/VS APPLICATION PROGRAMMER'S REFERENCE MANUAL (RPGII)
37	SC33-0512-01	2.1	CICS	CICS/MVS APPLICATION PROGRAMMER'S REFERENCE
38	SC33-0570-1	2	CICS	CICS/VM APPLICATION PROGRAMMING
39	SC33-0593-0	2	CICS	CICS/VM PROBLEM DETERMINATION

40	SH20-6810-00		CICS	CICS/VS PERFORMANCE ANALYSIS REPORTING SYSTEM GUIDE TO REPORTS
41	SH20-6839-01		CICS	CICS/VS PERFORMANCE ANALYSIS REPORTING SYSTEM PROBLEM DET GUIDE
42	SH20-6806-00		CICS	CICS/VS PERFORMANCE ANALYSIS REPORTING SYSTEM/VSE GUIDE TO ON-LINE
43	SH20-6808-00		CICS	CICS/VS PERFORMANCE ANALYSIS REPORTING SYSTEM/VSE INSTALLATION & ADMIN
44	SC26-4049-4	3.2	COBOL II	VS COBOL II APPLICATION PROGRAMMING DEBUGGING
45	SC26-4045-4	3.2	COBOL II	VS COBOL II APPLICATION PROGRAMMING GUIDE FOR MVS AND CMS
46	SC26-4697-0	3.2	COBOL II	VS COBOL II APPLICATION PROGRAMMING GUIDE FOR VSE
47	GC26-4047-6	3.2	COBOL II	VS COBOL II APPLICATION PROGRAMMING LANGUAGE REFERENCE
48	SX26-3721-4	3.2	COBOL II	VS COBOL II APPLICATION PROGRAMMING REFERENCE SUMMARY
49	LY27-9523-3	3.2	COBOL II	VS COBOL II DIAGNOSIS GUIDE
50	LY27-9522-3	3.2	COBOL II	VS COBOL II DIAGNOSIS REFERENCE
51	GC26-4042-7	3.2	COBOL II	VS COBOL II GENERAL INFORMATION
52	SC26-4213-4	3.2	COBOL II	VS COBOL II INSTALLATION AND CUSTOMIZATION FOR CMS
53	SC26-4696-0	3.2	COBOL II	VS COBOL II INSTALLATION AND CUSTOMIZATION FOR VSE
54	GH23-0500-4	3.3	CSP	CSP GENERAL INFORMATION
55	SH20-6769-0	3.3	CSP	CSP/AD CSP/AE ADMINISTERING CSP/AD AND CSP/AE ON OS/2 AND IBM DOS
56	SH20-6766-0	3.3	CSP	CSP/AD CSP/AE ADMINISTERING CSP/AD AND CSP/AE ON VM
57	SH20-6767-0	3.3	CSP	CSP/AD CSP/AE ADMINISTERING CSP/AD AND CSP/AE ON VSE
58	SK2T-6600-0	3.3	CSP	CSP/AD CSP/AE BASIC CSP INSTALLATION COURSEWARE INSTALLATION
59	SH23-0505-4	3.3	CSP	CSP/AD CSP/AE MESSAGES, CODES, AND PROBLEM DETERMINATION
60	SH20-6770-0	3.3	CSP	CSP/AD CSP/AE PLANNING
61	SH20-6434-0	3.3	CSP	CSP/AD DEFINING APPLICATIONS ON THE SYSTEM/370
62	SH20-6435-0	3.3	CSP	CSP/AD DEVELOPING APPLICATIONS
63	SH20-6433-1	3.3	CSP	CSP/AD EXTERNAL SOURCE FORMAT REFERENCE
64	SH20-0520-0	3.3	CSP	CSP/AD EXTERNAL SOURCE FORMAT REFERENCE SUMMARY
65	GH23-0515-3	3.3	CSP	CSP/AD REFERENCE
66	SH23-0519-1	3.3	CSP	CSP/AD REFERENCE SUMMARY
67	GH19-8122-01	3	DFT	DATA FILE TRANSFER, TESTING, AND OPERATIONS FOR VSE & VM GENERAL INFO
68	SH19-8123-1	3.2	DFT	DATA FILE TRANSFER, TESTING, AND OPERATIONS FOR VSE & VM USER'S GUIDE
69	GC35-0033-15	12	DSF	DEVICE SUPPORT FACILITIES USER'S GUIDE AND REFERENCE
70	GG66-3168-00		ENT SYS	ENTERPRISE SYSTEMS SUPPORT
71	GC28-1378-3	3.4	EREP	EREP USER'S GUIDE AND REFERENCE
72	LY27-9516-02	4	FORTRAN	VS FORTRAN VERSION 2 DIAGNOSIS GUIDE
73	SC26-4339-02	4	FORTRAN	VS FORTRAN VERSION 2 INSTALLATION AND CUSTOMIZATION FOR CMS
74	SC26-4223-2	3	FORTRAN	VS FORTRAN VERSION 2 INTERACTIVE DEBUG GUIDE AND REFERENCE
75	SC26-4221-04	4	FORTRAN	VS FORTRAN VERSION 2 LANGUAGE AND LIBRARY REFERENCE
76	SC26-4603-00	4	FORTRAN	VS FORTRAN VERSION 2 MASTER INDEX AND GLOSSARY
77	SC26-4222-04	4	FORTRAN	VS FORTRAN VERSION 2 PROGRAMMING GUIDE
78	SC33-0326-2	2.2	GDDM	GDDM DIAGNOSIS AND PROBLEM DETERMINATION GUIDE
79	SC33-0327-1	2.2	GDDM	GDDM GUIDE FOR USERS
80	SC33-0329-0	2.1	GDDM	GDDM IMAGE SYMBOL EDITOR
81	SC33-0323-2	2.2	GDDM	GDDM INSTALLATION AND SYSTEM MANAGEMENT FOR VM
82	SC33-0322-02	2.2	GDDM	GDDM INSTALLATION AND SYSTEM MANAGEMENT FOR VSE

83	GC33-0595-0	2.2	GDDM	GDDM LIBRARY GUIDE AND MASTER INDEX
84	SC33-0325-2	2.2	GDDM	GDDM MESSAGES
85	SC33-0324-0	2.1	GDDM	GDDM PERFORMANCE GUIDE
86	SC33-0328-0	2.1	GDDM	GDDM PGF INTERACTIVE CHART UTILITY
87	SC33-0330-0	2.1	GDDM	GDDM PGF VECTOR SYMBOL EDITOR
88	GC33-0320-1	2.2	GDDM	GDDM RELEASE GUIDE
89	GC34-4129-0	3	ISPF	ISPF AND ISPF/PDF DIRECTORY OF PROGRAMMING INTERFACES FOR CUSTOMERS
90	GC34-4186-0	2.2	ISPF	ISPF AND ISPF/PDF DIRECTORY OF PROGRAMMING INTERFACES FOR CUSTOMERS
91	GC34-4036-0	2.2	ISPF	ISPF AND ISPF/PDF GENERAL INFORMATION
92	GC34-4133-0	3	ISPF	ISPF AND ISPF/PDF GENERAL INFORMATION
93	SC34-2080-1	3	ISPF	ISPF AND ISPF/PDF INSTALLATION AND CUSTOMIZATION
94	SC34-4070-00	2.2	ISPF	ISPF AND ISPF/PDF INSTALLATION AND CUSTOMIZATION
95	SC34-4220-0	3	ISPF	ISPF AND ISPF/PDF PLANNING AND CUSTOMIZING
96	SC34-4017-0	2.2	ISPF	ISPF AND ISPF/PDF PRIMER
97	SC34-4219-0	3	ISPF	ISPF AND ISPF/PDF PRIMER
98	GC34-2172-4	3	ISPF	ISPF AND ISPF/PDF WHAT'S NEW IN ISPF AND ISPF/PDF ?
99	SC34-4009-0	2.2	ISPF	ISPF DIALOG MANAGEMENT GUIDE
100	SC34-4010-0	2.2	ISPF	ISPF DIALOG MANAGEMENT SERVICES AND EXAMPLES
101	SC34-4014-0	2.2	ISPF	ISPF/PDF EDIT MACROS
102	SC34-4243-0	2.2	ISPF	ISPF/PDF GUIDE
103	SC34-4013-0	2.2	ISPF	ISPF/PDF LIBRARY MANAGEMENT
104	SC34-4012-0	2.2	ISPF	ISPF/PDF PROGRAM DEVELOPMENT FACILITY SERVICES
105	SH21-0833-0		MISC	INVENTORY FORECASTING AND REPLENISHMENT MODULES III IMP. GUIDE
106	SC26-4319-1	2	PASCAL	VS PASCAL APPLICATION PROGRAMMING GUIDE
107	LY27-9525-1	2	PASCAL	VS PASCAL DIAGNOSIS GUIDE AND REFERENCE
108	SC26-4342-1	2	PASCAL	VS PASCAL INSTALLATION AND CUSTOMIZATION FOR VM
109	SC26-4320-1	2	PASCAL	VS PASCAL LANGUAGE REFERENCE
110	SX26-3760-1	2	PASCAL	VS PASCAL REFERENCE SUMMARY
111	PASO-9004-00		PASO	PASO LEVEL 9004 LOADING, INSTALLING AND USING
112	SC33-0051-0		PL/I	DOS PL/I OPTIMIZING COMPILER: CMS USER'S GUIDE
113	SC33-0019-2		PL/I	DOS PL/I OPTIMIZING COMPILER: EXECUTION LOGIC
114	SC33-0020-7	6	PL/I	DOS PL/I OPTIMIZING COMPILER: INSTALLATION
115	SC33-0021-4		PL/I	DOS PL/I OPTIMIZING COMPILER: MESSAGES
116	LY33-6010-2		PL/I	DOS PL/I OPTIMIZING COMPILER: PROGRAM LOGIC
117	SC33-0008-7	6	PL/I	DOS PL/I OPTIMIZING COMPILER: PROGRAMMER'S GUIDE
118	LY33-6011-2		PL/I	DOS PL/I RESIDENT LIBRARY: PROGRAM LOGIC
119	SC33-0035-2		PL/I	DOS PL/I TRANSIENT LIBRARY: MESSAGES
120	LY33-6012-1		PL/I	DOS PL/I TRANSIENT LIBRARY: PROGRAM LOGIC
121	GC26-3977-0		PL/I	OS AND DOS PL/I LANGUAGE REFERENCE MANUAL
122	SC26-3971-1		PL/I	OS AND DOS PL/I OPTIMIZING COMPILERS: EXTENDED GRAPHIC CHARACTER SET
123	GC26-4313-4	3	PL/I	OS PL/I VERSION 2 GENERAL INFORMATION
124	SC26-4312-2	3	PL/I	OS PL/I VERSION 2 INSTALLATION AND CUSTOMIZATION UNDER CMS
125	LY27-9528-2	3	PL/I	OS PL/I VERSION 2 PROBLEM DETERMINATION
126	SC26-4307-2	3	PL/I	OS PL/I VERSION 2 PROGRAMMING GUIDE
127	SC26-4308-2	3	PL/I	OS PL/I VERSION 2 PROGRAMMING: LANGUAGE REFERENCE
128	SC26-4309-2	3	PL/I	OS PL/I VERSION 2 PROGRAMMING: MESSAGES AND CODES

129	SX26-3759-2	3	PL/I	OS PL/I VERSION 2 PROGRAMMING: REFERENCE SUMMARY
130	SC26-4310-2	3	PL/I	OS PL/I VERSION 2 PROGRAMMING: USING PLITEST
131	SH20-6796-01	2.2	PROFS	GETTING STARTED WITH THE PROFESSIONAL OFFICE SYSTEM
132	SH20-7251-02		PROFS	INSTALLING AND MANAGING PROFESSIONAL OFFICE SYSTEM APPLICATIONS
133	GH20-6795-0	2	PROFS	INTRODUCING THE PROFESSIONAL OFFICE SYSTEM
134	SH20-6799-01	2.2	PROFS	MANAGING THE PROFESSIONAL OFFICE SYSTEM
135	SH20-6800-01	2.2	PROFS	PLANNING AND INSTALLING THE PROFESSIONAL SYSTEM
136	SH20-7250-02		PROFS	PROFESSIONAL OFFICE SYSTEM APPLICATION SUPPORT GETTING STARTED
137	SH20-6801-01	2.2	PROFS	PROFESSIONAL OFFICE SYSTEM PROGRAMMER'S GUIDE
138	SH21-0044-1		PROFS	PROFS EXTENDED MAIL INSTALLATION AND USER'S GUIDE
139	SH20-6802-01	2.2	PROFS	USING LINE-MODE SUPPORT WITH THE PROFESSIONAL OFFICE SYSTEM
140	SH20-6797-01	2.2	PROFS	USING THE PROFESSIONAL OFFICE SYSTEM
141	SH20-6344-00		PROFS	USING THE PROFESSIONAL OFFICE SYSTEM APPLICATIONS SUPPORT
142	SC26-4715-00	3.1	QMF	QUERY MANAGEMENT FACILITY ADVANCED USER'S GUIDE
143	SC26-4722-00	3.1	QMF	QUERY MANAGEMENT FACILITY APPLICATION DEVELOPMENT GUIDE
144	LY27-9590-00	3.1	QMF	QUERY MANAGEMENT FACILITY DIAGNOSIS GUIDE FOR VM
145	GC26-4713-01	3.1	QMF	QUERY MANAGEMENT FACILITY GENERAL INFORMATION
146	SC26-4718-00	3.1	QMF	QUERY MANAGEMENT FACILITY INSTALLATION GUIDE FOR VM
147	SC26-4714-00	3.1	QMF	QUERY MANAGEMENT FACILITY LEARNER'S GUIDE
148	SC26-4720-00	3.1	QMF	QUERY MANAGEMENT FACILITY PLANNING AND ADMINISTRATION GUIDE FOR VM
149	SC26-4717-00	3	QMF	QUERY MANAGEMENT FACILITY QUERY-BY-EXAMPLE GUIDE AND REFERENCE
150	SC26-4716-00	3.1	QMF	QUERY MANAGEMENT FACILITY REFERENCE
151	SX26-3783-00	3.1	QMF	QUERY MANAGEMENT FACILITY REFERENCE SUMMARY
152	SH24-5035-0	1	QMF	QUERY MANAGEMENT FACILITY/VSE INSTALLATION
153	SH24-5034-0	1	QMF	QUERY MANAGEMENT FACILITY/VSE PLANNING AND ADMINISTRATION
154	SC33-6034-1		RPG II	DOS/VS RPGII AUTO REPORT
155	SC33-6032-2		RPG II	DOS/VS RPGII INSTALLATION REFERENCE
156	SC33-6031-3	3	RPG II	DOS/VS RPGII LANGUAGE
157	LY33-9062-2		RPG II	DOS/VS RPGII LOGIC
158	SC33-6033-1		RPG II	DOS/VS RPGII MESSAGES
159	SC33-6074-0	3	RPG II	DOS/VS RPGII USER'S GUIDE
160	SC26-4399-1		SAA	SAA COMMON PROGRAMMING INTERFACE COMMUNICATIONS REFERENCE
161	LH19-6251-1		SDF	SCREEN DEFINITION FACILITY II DIAGNOSIS GUIDE
162	GH19-6114-1		SDF	SCREEN DEFINITION FACILITY II GENERAL INFORMATION
163	SH19-8128-0		SDF	SCREEN DEFINITION FACILITY II GENERAL INTRODUCTION - PART 1
164	SH19-8129-0		SDF	SCREEN DEFINITION FACILITY II GENERAL INTRODUCTION - PART 2
165	SH19-6458-1		SDF	SCREEN DEFINITION FACILITY II PREPARING A PROTOTYPE
166	SH19-6118-1		SDF	SCREEN DEFINITION FACILITY II PRIMER FOR CICS/BMS
167	SH19-6459-1		SDF	SCREEN DEFINITION FACILITY II PRIMER FOR GDDM-IMD & CSP/AD
168	SH19-6453-1		SDF	SCREEN DEFINITION FACILITY II PRIMER FOR IMS/MFS
169	SH19-6119-1		SDF	SCREEN DEFINITION FACILITY II PRIMER FOR ISPF
170	LY19-6065-1	5	SDF	SCREEN DEFINITION FACILITY/CICS DIAGNOSIS REFERENCE
171	SH19-8107-1		SDF	SCREEN DEFINITION FACILITY/CICS MESSAGES AND CODES
172	SH19-6094-02	5	SDF	SCREEN DEFINITION FACILITY/CICS OPERATIONS GUIDE
173	SH19-6102-0		SDF	SCREEN DEFINITION FACILITY/CICS PRIMER
174	LY19-6060-3		SDF	SCREEN DEFINITION FACILITY/CICS PROGRAM LOGIC

175	LY19-6066-1	5	SDF	SCREEN DEFINITION FACILITY/CICS PROGRAM LOGIC MANUAL
176	SH19-8105-1		SDF	SCREEN DEFINITION FACILITY/CICS PROGRAM REFERENCE
177	LY43-0081-1		SNA	SYSTEMS NETWORK ARCHITECTURE NETWORK PRODUCT FORMATS
178	LY33-8044-2		SORT/MERGE	DOS/VS SORT/MERGE VERSION 2 PROGRAM LOGIC
179	SC33-4045-5	2.5	SORT/MERGE	DOS/VS-VM/SP SORT/MERGE VERSION 2 INSTALLATION
180	SC33-4044-3	2.5	SORT/MERGE	DOS/VS-VM/SP SORT/MERGE VERSION 2 PROGRAMMING GUIDE
181	SH09-8086-00	3.1	SQL/DS	SQL/DATA SYSTEM APPLICATION PROGRAMMING FOR IBM VM SYSTEMS
182	SH09-8098-00	3.1	SQL/DS	SQL/DATA SYSTEM APPLICATION PROGRAMMING FOR VSE
183	GH09-8083-00	3.1	SQL/DS	SQL/DATA SYSTEM DATABASE ADMINISTRATION FOR IBM VM SYSTEMS
184	GH09-8095-00	3.1	SQL/DS	SQL/DATA SYSTEM DATABASE ADMINISTRATION FOR VSE
185	SH09-8088-00	3.1	SQL/DS	SQL/DATA SYSTEM DATABASE SERVICES UTILITY FOR IBM VM SYSTEMS
186	SH09-8100-00	3.1	SQL/DS	SQL/DATA SYSTEM DATABASE SERVICES UTILITY FOR VSE
187	LH09-8081-00	3.1	SQL/DS	SQL/DATA SYSTEM DIAGNOSIS GUIDE AND REFERENCE FOR IBM VM SYSTEMS
188	LH09-8093-00	3.1	SQL/DS	SQL/DATA SYSTEM DIAGNOSIS GUIDE AND REFERENCE FOR VSE
189	GH09-8074-00	3.1	SQL/DS	SQL/DATA SYSTEM GENERAL INFORMATION FOR IBM VM SYSTEMS
190	GH09-8075-00	3.1	SQL/DS	SQL/DATA SYSTEM GENERAL INFORMATION FOR VSE
191	GH09-8078-00	3.1	SQL/DS	SQL/DATA SYSTEM INSTALLATION FOR IBM VM SYSTEMS
192	GH09-8090-00	3.1	SQL/DS	SQL/DATA SYSTEM INSTALLATION FOR VSE
193	SH09-8085-00	3.1	SQL/DS	SQL/DATA SYSTEM INTERACTIVE SQL GUIDE AND REFERENCE FOR IBM VM SYSTEMS
194	SH09-8097-00	3.1	SQL/DS	SQL/DATA SYSTEM INTERACTIVE SQL GUIDE AND REFERENCE FOR VSE
195	SH09-8077-00	3.1	SQL/DS	SQL/DATA SYSTEM MANAGING SQL/DATA SYSTEM
196	SH09-8089-00	3.1	SQL/DS	SQL/DATA SYSTEM MASTER INDEX FOR IBM VM SYSTEMS
197	SH09-8101-00	3.1	SQL/DS	SQL/DATA SYSTEM MASTER INDEX FOR VSE
198	SH09-8079-00	3.1	SQL/DS	SQL/DATA SYSTEM MESSAGES AND CODES FOR IBM VM SYSTEMS
199	SH09-8091-00	3.1	SQL/DS	SQL/DATA SYSTEM MESSAGES AND CODES FOR VSE
200	SH09-8080-00	3.1	SQL/DS	SQL/DATA SYSTEM OPERATION FOR IBM VM SYSTEMS
201	SH09-8092-00	3.1	SQL/DS	SQL/DATA SYSTEM OPERATION FOR VSE
202	SH09-8103-00	3.1	SQL/DS	SQL/DATA SYSTEM PROCEDURES LANGUAGE INTERFACE
203	SX09-1173-00	3.1	SQL/DS	SQL/DATA SYSTEM REFERENCE SUMMARY FOR IBM VM SYSTEMS
204	SX09-1174-00	3.1	SQL/DS	SQL/DATA SYSTEM REFERENCE SUMMARY FOR VSE
205	SH09-8087-00	3.1	SQL/DS	SQL/DATA SYSTEM SQL REFERENCE FOR IBM VM SYSTEMS AND VSE
206	GH09-8084-00	3.1	SQL/DS	SQL/DATA SYSTEM SYSTEM ADMINISTRATION FOR IBM VM SYSTEMS
207	GH09-8096-00	3.1	SQL/DS	SQL/DATA SYSTEM SYSTEM ADMINISTRATION FOR VSE
208	SC31-6082-0		TCP/IP	TCP/IP VERSION 2 FOR VM: INSTALLATION AND MAINTENANCE
209	SC31-6084-0		TCP/IP	TCP/IP VERSION 2 FOR VM: PROGRAMMER'S REFERENCE
210	SC31-6081-0		TCP/IP	TCP/IP VERSION 2 FOR VM: USER'S GUIDE
211	SC34-2166-3	1.1	VM	VM MONITOR ANALYSIS PROGRAM USER'S GUIDE AND REFERENCE
212	SH20-2337-6	1.1	VM	VM REAL-TIME MONITOR PROGRAM DESCRIPTION/OPERATIONS
213	GC19-6212-5	5	VM	VM RUNNING GUEST OPERATING SYSTEMS
214	LY20-0890-04	6	VM	VM SERVICE ROUTINES PROGRAM LOGIC
215	SC24-5288-01	6	VM	VM SYSTEM FACILITIES FOR PROGRAMMING
216	SC24-5374-00	4	VM	VM/PASS-THROUGH FACILITY MANAGING AND USING
217	GC24-5502-00	4	VM	VM/PASS-THROUGH FACILITY PROGRAM UPDATE INFORMATION
218	LY20-0889-6	4	VM DIRECT	VM/DIRECTORY MAINTENANCE LICENSED PROGRAM DIAGNOSIS REFERENCE
219	GC20-1836-5	4	VM DIRECT	VM/DIRECTORY MAINTENANCE LICENSED PROGRAM GENERAL INFORMATION
220	SC23-0437-1	4	VM DIRECT	VM/DIRECTORY MAINTENANCE LICENSED PROGRAM OPERATION AND USE
221	SC23-0533-0	4	VM DIRECT	VM/DIRECTORY MAINTENANCE LICENSED PROGRAM SECURITY ENHANCEMENTS

222	SC23-0567-0	4	VM DIRECT	VM/DIRECTORY MAINTENANCE LICENSED PROGRAM VM/ESA SUPPORT
223	SC23-0563-0	4	VM DIRECT	VM/DIRECTORY MAINTENANCE PROGRAM DFSMS/VM AND 3390 DASD SUPPORT
224	SC24-5320-02	2.3	VM IPF	VM/INTERACTIVE PRODUCTIVITY FACILITY ADMINISTRATION
225	GC24-5472-00	2.3	VM IPF	VM/INTERACTIVE PRODUCTIVITY FACILITY BETWEEN-RELEASE SUPPORT INFO
226	SC24-5325-02	2.3	VM IPF	VM/INTERACTIVE PRODUCTIVITY FACILITY GENERAL USE GUIDE
227	SC24-5362-01	2.3	VM IPF	VM/INTERACTIVE PRODUCTIVITY FACILITY INTRODUCTION
228	SC24-5331-02	2.3	VM IPF	VM/INTERACTIVE PRODUCTIVITY FACILITY MAINTENANCE
229	SC24-5324-02	2.3	VM IPF	VM/INTERACTIVE PRODUCTIVITY FACILITY MESSAGES
230	SC24-5319-02	2.3	VM IPF	VM/INTERACTIVE PRODUCTIVITY FACILITY OPERATION
231	SC24-5323-02	2.3	VM IPF	VM/INTERACTIVE PRODUCTIVITY FACILITY PROBLEM CONTROL
232	SC24-5321-02	2.3	VM IPF	VM/INTERACTIVE PRODUCTIVITY FACILITY SYSTEM REFERENCE
233	GC24-5489-00	6	VM PGM UPD	VM PROGRAM UPDATE INFO ALTERNATE FORMAT EXECS AND EXEC LANGUAGE
234	GC24-5482-0	6	VM PGM UPD	VM PROGRAM UPDATE INFO CMS SERVICES COMMAND RESTRUCTURE
235	GC24-5488-00	6	VM PGM UPD	VM PROGRAM UPDATE INFO ECF/PASF COEXISTENCE WITH IBM VM/SP
236	GC24-5533-00	6	VM PGM UPD	VM PROGRAM UPDATE INFO ENHANCEMENTS FOR RECEIVING ELECTRONIC SERVICE
237	GC24-5418-00	6	VM PGM UPD	VM PROGRAM UPDATE INFO NATIONAL LANGUAGE SUPPORT FILE NAMING
238	GC24-5515-01	6	VM PGM UPD	VM PROGRAM UPDATE INFO SECURE SPOOL FILE ORIGIN DATA
239	GC24-5507-00	6	VM PGM UPD	VM PROGRAM UPDATE INFO SERVICE ENHANCEMENTS
240	GC24-5499-00	6	VM PGM UPD	VM PROGRAM UPDATE INFO SUPPORT FOR 3480 TAPE WITH IMPROVED RECORDING
241	LY24-5228-02	2.3	VM RSCS	VM RSCS NETWORKING DIAGNOSIS REFERENCE
242	SH24-5197-00	2.3	VM RSCS	VM RSCS NETWORKING EXIT CUSTOMIZATION
243	SH24-5196-00	2.3	VM RSCS	VM RSCS NETWORKING MESSAGES AND CODES
244	SH24-5058-03	2.3	VM RSCS	VM RSCS NETWORKING OPERATION AND USE
245	SH24-5057-03	2.3	VM RSCS	VM RSCS NETWORKING PLANNING AND INSTALLATION
246	SC24-5285-01	6	VM SP	VM/SP ADMINISTRATION
247	SC24-5247-03	6	VM SP	VM/SP APPLICATION DEVELOPMENT FOR FORTRAN AND COBOL
248	SC24-5286-01	6	VM SP	VM/SP APPLICATION DEVELOPMENT GUIDE FOR CMS
249	SC24-5284-01	6	VM SP	VM/SP APPLICATION DEVELOPMENT REFERENCE FOR CMS
250	SC24-5366-00	6	VM SP	VM/SP APPLICATION MIGRATION GUIDE FOR CMS
251	SC19-6209-05	6	VM SP	VM/SP CMS COMMAND REFERENCE
252	LY24-5221-03	6	VM SP	VM/SP CMS DATA AREA AND CONTROL BLOCKS
253	LY20-0893-05	6	VM SP	VM/SP CMS DIAGNOSIS REFERENCE
254	SC24-5236-04	6	VM SP	VM/SP CMS PRIMER
255	SC24-5367-00	6	VM SP	VM/SP CMS SHARED FILE SYSTEM ADMINISTRATION
256	SC19-6210-05	6	VM SP	VM/SP CMS USER'S GUIDE
257	SC24-5378-00	6	VM SP	VM/SP CONNECTIVITY PLANNING, ADMINISTRATION, AND OPERATION
258	SC24-5377-00	6	VM SP	VM/SP CONNECTIVITY PROGRAMMING GUIDE AND REFERENCE
259	LY24-5220-04	6	VM SP	VM/SP CP DATA AREA AND CONTROL BLOCKS
260	LY20-0892-05	6	VM SP	VM/SP CP DIAGNOSIS REFERENCE
261	SC19-6211-05	6	VM SP	VM/SP CP GENERAL USER COMMAND REFERENCE
262	SC24-5402-00	6	VM SP	VM/SP CP SYSTEM COMMAND REFERENCE
263	LY24-5241-01	6	VM SP	VM/SP DIAGNOSIS GUIDE
264	GC24-5417-00	6	VM SP	VM/SP DIRECTORY OF PROGRAMMING INTERFACES FOR CUSTOMERS
265	SC24-5219-04	6	VM SP	VM/SP EXEC 2 REFERENCE
266	SC24-5250-02	6	VM SP	VM/SP GROUP CONTROL SYSTEM COMMAND AND MACRO REFERENCE

267	SC24-5237-04	6	VM SP	VM/SP INSTALLATION GUIDE
268	SC24-5260-01	6	VM SP	VM/SP INTERACTIVE PROBLEM CONTROL SYSTEM GUIDE AND REFERENCE
269	GC19-6200-05	6	VM SP	VM/SP INTRODUCTION
270	GC19-6207-05	6	VM SP	VM/SP LIBRARY GUIDE AND MASTER INDEX
271	SC19-6202-05	6	VM SP	VM/SP OPERATOR'S GUIDE
272	SC19-6201-06	6	VM SP	VM/SP PLANNING GUIDE AND REFERENCE
273	SC24-5291-02	6	VM SP	VM/SP PROGRAMMER'S GUIDE TO THE SERVER-REQUESTOR INTERFACE FOR VM/SP
274	SC24-5368-00	6	VM SP	VM/SP RELEASE 6 GUIDE
275	SC24-5389-00	6	VM SP	VM/SP SERVICE GUIDE
276	SC19-6204-05	6	VM SP	VM/SP SYSTEM MESSAGES AND CODES
277	SC24-5264-02	6	VM SP	VM/SP SYSTEM MESSAGES CROSS-REFERENCE
278	SC24-5221-05	6	VM SP	VM/SP SYSTEM PRODUCT EDITOR COMMAND AND MACRO REFERENCE
279	SC24-5220-04	6	VM SP	VM/SP SYSTEM PRODUCT EDITOR USER'S GUIDE
280	SC24-5239-03	6	VM SP	VM/SP SYSTEM PRODUCT INTERPRETER REFERENCE
281	SC24-5238-04	6	VM SP	VM/SP SYSTEM PRODUCT INTERPRETER USER'S GUIDE
282	GC19-6206-05	6	VM SP	VM/SP TERMINAL REFERENCE
283	GC24-5399-00	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO ALTERNATE-VSAM
284	GC24-5540-00	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO ANCHOR MACRO SUPPORT
285	GC24-5497-00	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO CMS ACCESS COMMAND ENHANCEMENT
286	GC24-5484-00	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO COMMIT WITHOUT CLOSE ENHANCEMENT
287	GC24-5548-00	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO DIAGNOSE CODE X'84'
288	GC24-5412-00	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO DOUBLE-BYTE CHARACTER SET ENHANCEMENTS
289	GC24-5313-00	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO ENHANCEMENTS TO LINERD AND LINEWRT MACROS
290	GC24-5406-00	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO EXTENDED EXECUTOR LANGUAGE ENHANCEMENTS
291	GC23-0565-0	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO MONITOR CHANGES
292	GC24-5493-00	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO QUERY WORKUNITID CSL ROUTINE
293	GC24-5312-00	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO SECURITY AND INTEGRITY ENHANCEMENTS
294	GC24-5495-00	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO SHARED FILE SYSTEM ENHANCEMENTS
295	GC24-5545-00	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO SUPPORT FOR THE 9348 MOD 12 TAPE DRIVE
296	GC24-5490-00	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO SUPPORT FOR THE COBOL-85
297	GC23-0459-0	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO SUPPORT FOR VM/DIRECTORY MAINT REL 4
298	GC24-5542-00	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO SUPPORT OF SYSTEM/3270 MODELS 10,12,14
299	GC24-5392-00	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO TRANSPARENT SERVICES ACCESS VTAM LINE DRIVER
300	GC24-5422-00	6	VM SP PGM UPD	VM/SP PROGRAM UPDATE INFO TRANSPARENT SERVICES FULL BUFFER TRACE
301	GC24-5539-00	6	VM SP PGM	VM/SP PROGRAM UPDATE INFO VM WORKSTATION FACILITY PROGRAMMING
302	LY33-9090-1	2	VSE	VSE FAST COPY DATA SET DIAGNOSIS REFERENCE
303	SC33-6082-0		VSE	VSE FAST COPY DATA SET INSTALLATION REFERENCE
304	GC33-6156-1		VSE	VSE/ON-LINE TEST EXECUTIVE PROGRAM INSTALLATION AND OPERATION
305	SC33-6356-01	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS APPLICATION PROGRAMMING MACRO REFERENCE

306	SC33-6355-01	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS APPLICATION PROGRAMMING MACRO USER'S GUIDE
307	LY33-9148-00	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS DIAGNOSIS REF ERROR RECOVERY AND TRANSIENTS
308	LY33-9151-00	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS DIAGNOSIS REF LIBRARIAN
309	LY33-9152-00	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS DIAGNOSIS REF LINKAGE EDITOR
310	LY33-9116-0		VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS DIAGNOSIS REF LIOCS VOL 1 GEN INFO AND MACROS
311	LY33-9157-00	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS DIAGNOSIS REF LIOCS VOL 2 SAM
312	LY33-9118-0		VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS DIAGNOSIS REF LIOCS VOL 3 DAM AND ISAM
313	LY33-9159-00	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS DIAGNOSIS REF LIOCS VOL 4 SAM FOR DASD
314	LY33-9153-00	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS DIAGNOSIS REF MAINTAIN SYSTEM HISTORY PROGRAM
315	LY33-9155-00	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS DIAGNOSIS REF SERVICEABILITY AIDS
316	LY33-9150-00	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS DIAGNOSIS REF SUPERVISOR
317	LY33-9114-0		VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS DIAGNOSIS REF SYSTEM UTILITIES
318	LY33-9149-00	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS DIAGNOSIS REF TRANSIENTS AND $IJBSXXX PHASES
319	LY33-9160-00	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS INITIAL PROGRAM LOAD AND JOB CONTROL
320	SC33-6357-01	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS MAINTAIN SYSTEM HISTORY PROGRAM REFERENCE
321	SC33-6353-00	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS OPERATION
322	SC33-6351-01	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS PLANNING AND INSTALLATION
323	SC33-6358-00	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS SERVICE AIDS
324	SC33-6354-01	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS SYSTEM CONTROL STATEMENTS
325	SC33-6352-00	4.1	VSE ADV FUNCT	VSE/ADVANCED FUNCTIONS SYSTEM MANAGEMENT GUIDE
326	LY33-9120-0		VSE ICCF	VSE/INTERACTIVE COMPUTING AND CONTROL FACILITY
327	SC33-6203-02		VSE ICCF	VSE/INTERACTIVE COMPUTING AND CONTROL FACILITY INSTALLATION & OP REF
328	SC33-6204-02		VSE ICCF	VSE/INTERACTIVE COMPUTING AND CONTROL FACILITY TERMINAL USER'S GUIDE
329	SC33-6374-01	4.1	VSE POWER	VSE/POWER APPLICATION PROGRAMMING
330	LY33-9140-01	4.1	VSE POWER	VSE/POWER DIAGNOSIS REFERENCE MANUAL
331	SC33-6371-01	4.1	VSE POWER	VSE/POWER INSTALLATION AND OPERATIONS GUIDE
332	SC33-6373-01	4.1	VSE POWER	VSE/POWER NETWORKING USER'S GUIDE
333	SC33-6372-01	4.1	VSE POWER	VSE/POWER REMOTE JOB ENTRY USER'S GUIDE
334	SC33-6405-01	4.1	VSE SP	VSE/SP ADMINISTRATION
335	GC33-6401-01	4.1	VSE SP	VSE/SP GENERAL INFORMATION
336	SC33-6410-01	4.1	VSE SP	VSE/SP GUIDE FOR SOLVING PROBLEMS
337	SC33-6404-01	4.1	VSE SP	VSE/SP INSTALLATION
338	SC33-6402-01	4.1	VSE SP	VSE/SP INSTALLATION PLANNING
339	SC33-6420-01	4.1	VSE SP	VSE/SP MASTER INDEX
340	SC33-6407-01	4.1	VSE SP	VSE/SP MESSAGES AND CODES VOLUME 1 AND 2
341	SC33-6408-01	4.1	VSE SP	VSE/SP NETWORKING SUPPORT
342	SC33-6406-01	4.1	VSE SP	VSE/SP OPERATION
343	GX33-9022-01	4.1	VSE SP	VSE/SP QUICK REFERENCE
344	SC33-6411-00	4.1	VSE SP	VSE/SP RELEASE INFORMATION GUIDE
345	SC33-6403-01	4.1	VSE SP	VSE/SP SYSTEM PLANNING
346	SC33-6412-00	4.1	VSE SP	VSE/SP UNATTENDED NODE SUPPORT
347	SC33-6409-01	4.1	VSE SP	VSE/SP USING IBM 3270 DISPLAY STATIONS AND PCS

348	LY24-5195-1	2	VSE VSAM	VSE/VSAM ACCESS METHOD SERVICES LOGIC
349	LY33-9130-00	1.4	VSE VSAM	VSE/VSAM LOGIC
350	SC33-6435-00	1.4	VSE VSAM	VSE/VSAM PROGRAMMER'S REFERENCE
351	LY24-5204-0	2	VSE VSAM	VSE/VSAM SPACE MANAGEMENT FOR SAM FEATURE LOGIC
352	SC33-6432-00	1.4	VSE VSAM	VSE/VSAM USING COMMANDS AND MACROS
353	LY24-5191-2	3	VSE VSAM	VSE/VSAM VSAM LOGIC, VOLUME 1: CATALOG, OPEN/CLOSE, DADSM, CTL BLOCK
354	LY24-5192-02	3	VSE VSAM	VSE/VSAM VSAM LOGIC, VOLUME 2: RECORD MANAGEMENT
355	SC30-3351-2		VTAM	NETWORK PROGRAM PRODUCTS (NCP, NETVIEW, VTAM) PLANNING
356	SC31-6801-0		VTAM	NETWORK PROGRAM PRODUCTS (NCP, NETVIEW, VTAM) SAMPLES
357	SC30-3403-6		VTAM	NETWORK PROGRAM PRODUCTS (NCP, NETVIEW, VTAM) STORAGE ESTIMATES
358	LY43-0046-1	3.3	VTAM	VTAM CUSTOMIZATION
359	LY43-0045-0	3.3	VTAM	VTAM DATA AREAS FOR VM
360	LY30-5594-0	3.2	VTAM	VTAM DATA AREAS FOR VSE
361	LY43-0042-1	3.3	VTAM	VTAM DIAGNOSIS
362	GC31-6429-1	3.3	VTAM	VTAM DIRECTORY OF MIGRATION INFORMATION
363	GC31-6403-0	3.2	VTAM	VTAM DIRECTORY OF PROGRAMMING INTERFACES FOR CUSTOMERS
364	SC23-0111-5	3.2	VTAM	VTAM INSTALLATION AND RESOURCE DEFINITION
365	SC31-6405-1	3.3	VTAM	VTAM MESSAGES AND CODES
366	SC31-6404-1	3.3	VTAM	VTAM NETWORK IMPLEMENTATION GUIDE
367	SC23-0113-5	3.2	VTAM	VTAM OPERATION
368	SC31-6408-1	3.3	VTAM	VTAM OPERATION
369	SC31-6811-0		VTAM	VTAM PLANNING AND REFERENCE FOR NETVIEW NETWORK CONTROL PROGRAM VTAM
370	SC23-0115-4	3.2	VTAM	VTAM PROGRAMMING
371	SC31-6409-1	3.3	VTAM	VTAM PROGRAMMING
372	SC30-3400-1	3.2	VTAM	VTAM PROGRAMMING FOR LU6.2
373	SC31-6410-1	3.3	VTAM	VTAM PROGRAMMING FOR LU6.2
374	LY43-0047-1	3.3	VTAM	VTAM REFERENCE SUMMARY
375	SC31-6412-1	3.3	VTAM	VTAM RESOURCE DEFINITION REFERENCE

Appendix C - Tapes

The following is a list of all IBM 9371/9375 software tapes received with the systems:

Seq	Volume	Tape name
1	A003T	CSP AE V3 VM/SP V3 BASIC
2	A005D	VSE/AF V4
3	A005F	VSE/AF V4
4	A005G	VSE/AF V4
5	A01D2	ISPF/PDF V2 FOR VM/SP BASIC
6	A01GF	SCREEN DEFINITION FACILITY II VM
7	A01GG	SCREEN DEFINITION FACILITY II VM
8	A01GH	CICS/VM
9	A01R0	ISPF/PDF VER 2 FOR VM/SP BASIC
10	A01R1	PROFS V2R2 MOD # 3
11	A01R2	PROFS-223F-5002
12	A01R3	PROFS PASF V2R2 MOD # 2
13	A01R4	VM/IPF V2 VM/IPF BASIC
14	A01R5	VS PASCAL COMPILER AND LIBRARY BASIC CMS
15	A01R6	VS PASCAL COMPILER AND LIBRARY OPT SOURCE
16	A01R7	VS FORTRAN V2 C/L/I BASIC MATERIAL
17	A01R8	VM FORTRAN V2 C/L/I 5668-801 RESTRICTED
18	A01R9	GDDM-PGF V2 PGF.BASIC.VM
19	A01RG	EREP V3
20	A01RH	VM/SP 3370/9335 START SYSTEM
21	A01RJ	VM/SP OBJECT VOL 1
22	A01RK	VM/SP OBJECT VOL 5
23	A01RL	VM/SP OBJECT VOL 4
24	A01RM	VM/SP OBJECT VOL 2
25	A01RN	VM/SP OBJECT VOL 3
26	A01RP	VM/SP CMS/IPCS SOURCE
27	A01RQ	VM/SP CP SOURCE
28	A01RR	VM/SP PASO 9004 VOL 3
29	A01RS	VM/SP PASO 9004 VOL 2
30	A01RT	VM/SP PASO 9004 VOL 1
31	A01RV	RSCS VER 2 RSCS BASE
32	A01RW	VM MAP BASIC
33	A01RX	GDDM/VM GDDM.BASIC.VM
34	A01RY	GDDM/VM GDDM.PCLKF.VM
35	A01RZ	ISPF/PDF FOR VM/SP VER 2 BASIC VM
36	A01S0	PROFS EXTENDED MAIL
37	A01S1	PROFS EXTENDED MAIL
38	A01S2	VSE/ON-LINE TEST EXECUTIVE
39	A01S3	VSE/ICCF V2
40	A01S4	VSE/POWER V4
41	A01S5	VSE/SP V1 UNIQUE CODE
42	A01S6	DOS PL/I OPT COMP COMPOSITE OPT 1

43	A01S7	DOS PL/I OPT COMP COMPOSITE OPT 2
44	A01S8	VSE/FAST COPY DATA SET PROG
45	A01S9	BTAM-ES FOR DOS/VSE
46	A01SB	CSP/AD V3 VM/SP V3 BASIC
47	A01SC	CSP/AD V3 VM/SP V3 PWS
48	A01SD	CSP/AE V3 VM/SP V3 BASIC
49	A01SF	OS/PL/I V2 COMP, LIB, ITF COMMON-LIB-CMS
50	A01SG	OS/PL/I V2 COMP, LIB, ITF BASIC-CMS
51	A01SH	VS COBOL II COMP, LIB, DEBUG
52	A01SJ	ASSEMBLER H V2
53	A01SK	ASSEMBLER H V2
54	A01SL	ACF VTAM V3 FOR VM/9370
55	A01SM	DITTO VSE/VM
56	A01SN	SQL/DS 3.1 VSE BASIC
57	A01SP	SQL/DS 3.1 VM BASIC
58	A01SQ	SQL/DS 3.1 RXSQL FEATURE
59	A01SR	QMF VM V3R1
60	A01SS	VSE/VSAM
61	A01ST	VSE/VSAM
62	A01SV	VSE/VSAM
63	A01SW	ICKDSF
64	A01SX	PVM
65	A01SY	VS/DIRMAINT LICENSED PROGRAM 1 OF 1
66	A01SZ	VM/370 REAL TIME MONITOR BASIC
67	A01TB	DOS/VS RPG II OPTIONAL REL 3
68	A01TC	SCREEN DEF FACILITY/CICS
69	A01TD	CICS/DOS/VS I69 OPTIONAL VOL 1
70	A01TF	CICS/DOS/VS I69 OPTIONAL VOL 2
71	A01TG	CICS/DOS/VS I69 OPTIONAL VOL 3
72	A01TH	VSE/SP 4.1.2 VSAM FOR VM
73	A01TJ	VSE/SP 4.1.2 VM/DITTO
74	A01TK	VSE/SP 4.1.2 CICSPARS-VSAM-DATA
75	A01TZ	ISPF/PDF V2 FOR VM/SP BASIC
76	B0006	TCP/IP V2 FOR VM 1 OF 2
77	B0007	TCP/IP V2 FOR VM 2 OF 2
78	C0034	VSE/SP 4.1.2 VSE BASE TAPE
79	C0035	VSE/SP 4.1.2 VSE BASE TAPE
80	C0036	VSE/SP 4.1.2 VSE BASE TAPE
81	C0037	VSE/SP 4.1.2 VSE OPT TAPE 1 OF 3
82	C0038	VSE/SP 4.1.2 VSE OPT TAPE 2 OF 3
83	C0039	VSE/SP 4.1.2 VSE OPT TAPE 3 OF 3
84	F0023	SERVICE FOR VM/370-9103 5664307/G 5664307/H
85	F0024	SERVICE FOR VM/370-9103 5684011/A
86	F0070	SERV FOR VM/370-9101 VMSERV 5654260/C 1 OF 1
87	F0071	SERV FOR VM/370-9101 VMSERV 5664167/E 1 OF 6
88	F0072	SERV FOR VM/370-9101 VMSERV 5664167/E 2 OF 6
89	F0073	SERV FOR VM/370-9101 VMSERV 5664167/E 3 OF 6
90	F0074	SERV FOR VM/370-9101 VMSERV 5664167/E 4 OF 6
91	F0075	SERV FOR VM/370-9101 VMSERV 5664167/E 5 OF 6
92	F0076	SERV FOR VM/370-9101 VMSERV 5664167/E 6 OF 6

93	F0077	SERV FOR VM/370-9101 5664188/B 1 OF 1
94	F0078	SERV FOR VM/370-9101 VMSERV 5664191/B 1 OF 1
95	F0079	SERV FOR VM/370-9101 5664200/E & /G
96	F0080	SERV FOR VM/370-9101 5664282/B
97	F0081	SERV FOR VM/370-9101 5664318/K 1 OF 1
98	F0082	SERV FOR VM/370-9101 5668806/D
99	F0083	SERV FOR VM/370-9101 5668962 1 OF 1
100	F0084	SERV FOR VM/370-9101 5688103
101	F0085	SERV FOR VM/370-9101 5748RC1/B
102	F0086	SERV FOR VM/370-9101 5748XE4/L
103	F0087	SERV FOR VM/370-9101 5746AM2 5688052

Appendix D - Cartridges

The following is a list of all IBM 9371/9375 software cartridges received with the systems:

Seq	Volume	Cartridge name
1	A01B8	PROFS
2	A01B9	PROFS V2V2M3
3	A01R0	VM/SP OBJECT VOL 3
4	A01R1	VM/SP OBJECT VOL 1
5	A01R2	VM/SP OBJECT VOL 2
6	A01R3	VM/SP OBJECT VOL 4
7	A01R4	VM/SP OBJECT VOL 5
8	A01R5	VM/SP PAS09004 VOL 1
9	A01R6	VM/SP PAS09004 VOL 2
10	A01R7	VM/SP PAS09004 VOL 3
11	A01R8	VM/SP 0671 STARTER SYSTEM
12	A01RY	VM/SP CP SOURCE
13	A01RZ	VM/SP CMS/IPCS SOURCE
14	A01T0	VMMAP BASIC
15	A01T1	GDDM/VM
16	A01T2	GDDM/VM
17	A01T3	ISPF FOR VM/SP V2
18	A01T4	ISPF/PDF V2 FOR VM/SP BASIC
19	A01T5	VS PASCAL COMPILER AND LIBRARY
20	A01T6	VS PASCAL COMPILER AND LIBRARY
21	A01T7	VS FORTRAN V2 C/L/I
22	A01T8	VS FORTRAN V2 C/L/I
23	A01T9	GDDM-PGF V2
24	A01TL	EREP V3
25	A01TZ	RSCS V2
26	A01V0	VSE/SP 4.1.2 VM/DITTO
27	A01V1	VSE/SP 4.1.2 PS/CICS
28	A01V2	VSE/SP 4.1.2 PS/CICS
29	A01V3	VSE/SP 4.1.2 CICSPARS
30	A01VB	CSP/AD V3
31	A01VC	CSP/AD V3
32	A01VD	CSP/AE V3
33	A01VF	OS PLI V2 COMP,LIB,ITF
34	A01VG	OS PLI V2 COMP,LIB,ITF
35	A01VH	VS COBOL II COMP,LIB,DEBUG
36	A01VJ	ASSEMBLER/H V2
37	A01VK	ACF/VTAM V3 FOR VM 9370
38	A01VL	SQL/DS V2
39	A01VM	DITTO VSE/VM
40	A01VN	QMF VM V3R1
41	A01VP	VSE/VSAM
42	A01VQ	VSE/VSAM

43	A01VR	VSE/VSAM
44	A01VS	PVM
45	A01VT	VM/DIRMAINT LICENSED PROGRAM
46	A01VV	VS COBOL II COMP,LIB,DEBUG
47	A01VW	VS COBOL II COMP,LIB,DEBUG
48	A01VX	VSE/VSAM
49	A01VY	VSE/VSAM
50	A01VZ	VSE/SP 4.1.2 VSAM FOR VM
51	B0008	TCP/IP V2 VOR VM
52	C0040	VSE/SP 4.1.2 VSE BASE TAPE
53	C0041	VSE/SP 4.1.2 VSE OPTIONAL TAPE
54	F0048	SERV FOR VM/370-9101 VM/SP
55	F0049	SERV FOR VM/370-9101 VM/SP
56	F0088	SERV FOR VM/370-9101 EREP V3
57	F0090	SERV FOR VM/370-9101 RSCS V2
58	F0091	SERV FOR VM/370-9101 VMMAP
59	F0092	SERV FOR VM/370-9101 GDDM/VM
60	F0093	SERV FOR VM/370-9101 ISPF FOR VM/SP V2
61	F0094	SERV FOR VM/370-9101 VS FORTRAN V2
62	F0095	SERV FOR VM/370-9101 ASSEMBLER/H V2
63	F0096	SERV FOR VM/370-9101 SQL/DS V2
64	F0097	SERV FOR VM/370-9101 PVM
65	F0098	SERV FOR VM/370-9101 VM/DIRMAINT LICENSED PROGRAM

Appendix E - VM/SP Installation Steps

The following is a step-by-step summary of the VM/SP installation and customization activities.

1. Initialize the DASD volumes. The VM/SP Starter System tape must be loaded into the tape drive. Then using the system console, command the system to IPL from that device. The program invoked is the Device Support Facility (DSF). This program is used to initialize disk packs and check disk surfaces for errors.

2. Format the DASD volumes. Command the system to IPL again from the same tape. This will load and execute the second program. This is the CP format/allocate program which formats the disk packs and assigns an identifying label to them.

3. Restore the starter system. IPL the third program from the starter tape. This was the DDR (DASD Dump and Restore) program which was used to load the VM/SP starter system from the tape onto the disk packs prepared above. The starter system is a "stripped down" version of the VM/SP operating system which provides just enough functionality so that the systems programmer can define the specifics of the system and build a custom version of VM/SP.

4. IPL and define the starter system. IPL the system from the disk volume where the starter system was loaded. A series of questions will then be asked by the system regarding the physical characteristics of the system. This information is stored in system configuration files and is used to build the final version of VM/SP.

5. Load installation tools from tape to disk. Use the VMFPLC2 command to copy programs from the installation tape to disk to assist in the VM/SP build process.

6. Load system generation tools and files. Use the installation tool ITASK to load additional tools which allow the installer to build a customized version of VM/SP specific to the particular hardware configuration.

7. Tailor the system generation files. This includes modifying the VM/SP directory which lists all users of the system, DMKSYS which defines where system paging and dump areas are to be located, DMKRIO which identifies all input/output devices and the addresses of those devices, DMKSNT which specifies all named systems (like CMS) and segments, and DMSNGP which identifies the specifics of the CMS configuration.

8. Build the CP nucleus. The installation tool ITASK is also used to assemble the system generation files and generate an IPLable CP nucleus.

9. IPL the new CP and format the operator minidisk. When the new CP is IPLed it will automatically logon as OPERATOR. The OPERATOR's primary minidisk must then be formatted to prepare it for data storage.

10. Allocate other system disk volumes using the ITASK procedure. Initially only the primary disk volume was formatted for storage of VM/SP. At this point all other disk volumes must be formatted. ReIPL so that the disk volumes will be known to the system.

11. CMS files must then be loaded to the new disks. The ITASK tool will do this and then invoke the system assembler to build a CMS nucleus. Then the CMS system will be saved.

12. Next load IPCS (Interactive Problem Control System) files using the ITASK installation tool. IPCS is used to assist in reporting system problems and diagnosing them by providing easy viewing of system dumps.

13. The primary minidisks for other system logons must now be formatted.

14. Build the system file pool (a shared file system) in preparation for code to be loaded later. The system file pool provides a centralized storage place for system disk files without allocating a special minidisk.

15. Build the user file pool for centralized storage of user files.

16. Use the ITASK utility to load and save system help files.

17. Load, build, and save GCS (Group Control System) which allows VM/SP to access a network such as VTAM or RSCS.

18. Load the TSAF (Transparent Services Access Facility) which allows remote access to a SQL database.

19. Load the AVS (APPC/VM VTAM Support) which allows program to program communications in a host-server environment.

20. Certain components of source code are included with the VM/SP installation materials. These allow the systems programmer even more flexibility and options when customizing. Included are CP, CMS, IPCS, GCS, and TSAF source code components.

21. If the system is to be connected to a network managed by RSCS, then a unique name must be assigned for the system and stored in a special file (the SYSTEM NETID file).

22. Install EREP (Environmental Recording, Editing and Printing). The VM/SP system will automatically record system software and hardware errors onto special areas of the disk. EREP is then used to extract this data and format it into reports which are primarily used by IBM customer engineers for problem diagnosis.

23. Perform IVP (Installation Verification Procedures) to insure that all components installed work properly. Two special virtual machines are automatically added to the system which should be logged onto and then a comprehensive series of tests can be performed to insure that the VM/SP system is ready for use.

24. Install system saved segments: CMSDOS, CMSBAS, CMSVSAM, CMSAMS, CMSINST, CMSVMLIB, and CMSFILES. The VM/SP Installation Guide describes a saved segment as an area of virtual storage that (a) has been saved in real storage, (b) has a unique name associated with it, (c) contains read-only or reentrant code, and (d) can be shared by multiple users [3]. Certain frequently used CMS components (as mentioned above) are commonly defined in this way.

25. Finally system clean up must be performed. This includes renaming disk packs, adding users, establishing system exec procedure for system setup during signon, and entering Xedit profiles.

End of Report