

Nº 40
3 EJ.



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

“ SISTEMA DE CONTROL DE RECURSOS DE
UNA MACROCOMPUTADORA I. B. M.
BAJO EL SISTEMA OPERATIVO VM/CMS ”

T E S I S
QUE PARA OBTENER EL TITULO DE
A C T U A R I O
P R E S E N T A
PATRICIA TAIANA MONTESINOS LOAYZA
Director: Fis. Adan Zepeda Gorostiza

MEXICO, D. F.

1992

FALLA EN ORIGEN



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

INDICE

INTRODUCCION	13
CAPITULO 1	
SISTEMA OPERATIVO VM/CMS	15
1.1 Historia del sistema operativo VM.	15
1.2 Definición.	18
1.3 ¿Qué es una maquina virtual?	18
1.4 Componentes de VM.	18
1.4.1 Control Program.	20
1.4.2 Conversational Monitor System.	20
1.4.3 Remote Spooling Communications Subsystem.	21
1.4.4 Group Control System.	21
1.5 Configuración.	21
1.5.1 Procesador.	21
1.5.2 Memoria.	21
1.5.3 Dispositivos de Entrada y Salida.	21
1.5.3.1 Consola.	23
1.5.3.2 Minidiscos.	23
1.5.3.3 Lectora y Perforadora de tarjetas virtuales.	23
1.5.3.4 Impresora virtual.	23
1.5.3.5 Unidades de cintas magnéticas.	23
1.6 Sistemas de comunicación del VM/CMS.	23
1.6.1. RSCS.	23
1.6.2 ACF/VTAM.	25
1.6.3 NCP/EP.	25
1.6.4 X.25 NPSI.	25
1.6.5. TCP/IP.	26
1.6.6 VM Pass-Trough Facility.	26
1.6.7 3270 PC File Transfer.	26
1.7 Algunas aplicaciones del VM/CMS.	27
1.7.1 Sistemas de automatización para oficinas.	27
1.7.1.1 DCF (Document Composition Facility).	27

1.7.1.2 Display Write / 370.	27
1.7.1.3 Office Vision y PROFS (Professional Office System).	27
1.7.2 Graficación y diseño por computadora.	27
1.7.2.1 GDDM (Graphical Data Display Manager).	27
1.7.2.2 CADAM (Computer-Graphics Augmented Design and Manufacturing).	27
1.7.2.3 CAEDS (Computer Aided Engineering Design System).	28
1.7.2.4 CIEDS/CBDS (Computer-Integrated Electrical Design Series/Circuit Board Design System)	28
1.7.3 BASES DE DATOS.	28
1.7.3.1 SQL/DS (Structured Query Language/Data System).	28
1.7.3.2 QMF (Query Management Facility).	28
1.7.4 Inteligencia artificial.	28
1.7.5 Matemáticas y estadística.	29
1.7.5.1 ACRITH (Accuracy Arithmetic Subroutines).	29
1.7.5.2 MPSX (Mathematical Programming System).	29
1.7.5.3 SAS (Statistical Analysis System).	29
1.8 Lenguajes de programación en VM/CMS.	29
1.9 El editor de textos XEDIT	30
1.10 Lenguaje intérprete de comandos REXX.	30

CAPITULO 2

LENGUAJE DE PROGRAMACION REXX	31
2.1 ¿Qué es un programa REXX?	31
2.2 Características.	31
2.2.1 Facilidad de uso.	31
2.2.2 Formato libre.	31
2.2.3 Funciones convenientemente integradas.	31
2.2.4 Fácil rastreo de errores.	31
2.2.5 Intérprete.	32
2.2.6 Capacidad extensiva de desglosamiento.	32
2.3 Estructura y sintaxis general.	32
2.3.1 Variables y expresiones.	32
2.3.2 Operadores.	33
2.4 Algunas instrucciones y funciones básicas.	34
2.5 Estructuras de control.	35
2.5.1 Selección.	35
2.5.2 Loops.	36
2.5.3 Subrutinas y funciones.	37
2.5.4 Jumps.	38
2.6 Manejo de pilas.	39

2.6.2	Uso de la pila de programas.	41
2.7	Manejo de archivos.	41
2.7.1	Uso de comandos EXECIO.	41
2.7.1.1	Lectura de archivos.	41
2.7.1.2	Escritura de archivos.	42

CAPITULO 3

DESCRIPCION DEL PROCESO DE CONTROL DE UTILIZACION DE RECURSOS DEL SISTEMA		43
3.1	Generalidades.	43
3.2	Privilegios y prioridades.	43
3.3	Directorio de usuarios.	44
3.4	Número de contabilidad.	46
3.5	Macro DMKSYS.	46
3.5	Instrucción DIAGNOSE.	48
3.6	Proceso para generar registros de contabilidad.	49

CAPITULO 4

DISEÑO GENERAL DEL SISTEMA DE CONTROL DE UTILIZACION DE RECURSOS		51
4.1	Investigaciones preliminares.	51
4.1.1	Antecedentes.	51
4.1.2	Objetivo del nuevo sistema.	51
4.1.3	Estudio de factibilidad.	51
4.1.4	Limitantes.	52
4.2	Diseño de entrada.	52
4.2.1	Archivo de entrada "ACCOUNT CARDS".	52
4.2.1.1	Registros de contabilidad para usuarios de una máquina virtual	53
4.2.1.2	Registros de contabilidad para dispositivos dedicados.	54
4.2.1.3	Registros de contabilidad para áreas temporales.	54
4.2.1.4	Registros de contabilidad para logon inválidos.	55
4.2.1.5	Registros de contabilidad para ligas a minidisks.	55
4.2.1.6	Registros de contabilidad para la máquina virtual de servicio VTAM	57
4.2.1.7	Registros de contabilidad con identificación de terminal.	57
4.2.1.8	Registros de contabilidad para la máquina virtual de servicio RSCSV2.	58
4.2.1.9	Registros de contabilidad generados por SQL/DS.	58
4.2.1.10	Creación de registros de contabilidad.	61
4.2.2	Archivo "DIRUSER DATA".	61
4.2.3	Archivo "DEPUNAM DATA".	61
4.2.4	Otros archivos.	61
4.2.5	Diseño de pantallas.	62

4.3 Diseño de salida.	62
4.3.1 Reporte general.	62
4.3.2 Reporte sobre el uso de discos temporales.	63
4.3.3 Reporte sobre logon inválidos.	63
4.3.4 Reporte de link inválidos.	64
4.3.5 Reporte sobre la utilización de terminales.	64
4.3.6 Reporte sobre utilización de VTAM.	64
4.3.7 Reporte sobre la utilización de RSCSV2.	65
CAPITULO 5	
DISEÑO DE SOFTWARE Y CONSTRUCCION DEL SISTEMA DE CONTROL DE UTILIZACION DE RECURSOS	77
CONCLUSIONES	97
APENDICE A	97
TIPOS DE DISPOSITIVOS DE ACCESO DIRECTO	97
APENDICE B	98
CLASES Y TIPOS DE DISPOSITIVOS	98
BIBLIOGRAFIA	103

INTRODUCCION

Actualmente la Dirección General de Servicios de Cómputo Académico (DGSCA), con el fin de apoyar las actividades administrativas, de investigación y docencia de la Universidad Nacional Autónoma de México (UNAM), cuenta con una variedad de equipos de cómputo. Uno de ellos es el procesador IBM-4381, el cual trabaja con el sistema operativo VM/SP, que permite realizar tareas de multiprogramación y multiprocesamiento en tiempo compartido.

Debido a esta diversidad de tareas, es necesario llevar un control y tener conocimiento sobre los recursos que maneja el sistema, para hacer una correcta asignación de estos y ofrecer un servicio cada vez más acorde a las necesidades de los usuarios.

Se pretende, mediante este trabajo, cubrir en gran medida estos requerimientos, mediante el diseño e implantación de un sistema de control de recursos, el cual genere reportes periódicos con información acerca de su disponibilidad y funcionamiento.

Para desarrollar este proyecto se cuenta, en dicha dependencia de la U.N.A.M., con el hardware y software necesario que permita acceder todo tipo de información, además de los manuales de consulta del equipo.

Inicialmente, es necesario conocer los programas que generan la contabilidad actualmente, así como familiarizarse con las tareas e información que procesa automáticamente el equipo.

Con esta bases, se determinará el número y tipo de reportes que se crearán, explotando las máximo los datos disponibles. Posteriormente, en el diseño del sistema que se propone, se describen las características del archivo maestro, los formatos de las pantallas, los formatos de los reportes, así como todos los procedimientos que lo forman, representados en diagramas estructurados, para su mejor entendimiento.

Para mejor comprensión del modo en que trabaja el sistema operativo, se hará una descripción de sus características y software que soporta.

También se considera importante detallar el lenguaje REXX, que se utiliza para la codificación de los programas, pues está diseñado específicamente para trabajar en el sistema operativo VM/CMS, por lo cual es muy poco conocido.

El presente trabajo propone nueva información que puede ser explotada dentro de la contabilidad y el monitoreo del sistema, brindando una síntesis de su funcionamiento y rendimiento.

Se espera satisfacer los requerimientos de la dependencia, en beneficio de la comunidad de usuarios.

CAPITULO 1

SISTEMA OPERATIVO VM/CMS

1.1 HISTORIA DEL SISTEMA OPERATIVO VM.

A finales de la década de 1940 y durante la de 1950, la mayoría de los computadores eran sistemas dedicados a un solo usuario, quien se sentaba frente a la consola de la máquina real con todas las capacidades de ésta visibles y disponibles para él. Uno de los problemas era el costo que representaba el tiempo que la máquina permaneciera ociosa.

Posteriormente se desarrollan los sistemas de multiprogramación por lotes para conseguir un mejor aprovechamiento de los costosos recursos en ese entonces, pero esto se hizo impráctico para los usuarios que debían controlar completamente un computador.

En 1961 se desarrolla en el Instituto Tecnológico de Massachusetts (M.I.T.), el sistema de tiempo compartido CTSS (Compatible Time-Sharing System), primero en una IBM 709 y después en una IBM 7090. CTSS ejecutaba una corriente convencional de lotes para mantener al computador ocupado mientras le daba rápidas respuestas a los usuarios interactivos que estaban editando y depurando programas.

CP/CMS comenzó como sistema experimental en 1964; estaba destinado a ser un sistema de tiempo compartido de segunda generación basado en computadores IBM/360. Fue desarrollado en principio para uso local en el Centro Científico de IBM en Cambridge, y su primera versión operacional apareció en 1966 como parte del proyecto CP-40. Este componente, diseñado para ser ejecutado en una IBM 360/40, tiene un nuevo hardware, modificado con el dispositivo de traducción dinámica de direcciones.

La primera versión del producto de CMS fue obtenida en 1968, como parte del programa de control CP-67/Cambridge Monitor System Versión 1.0 en el System/360 Modelo 67, desarrollándose un total de tres versiones de CP-67 en 1972.

CP-67/CMS disponía de hardware de traducción dinámica de direcciones y se suponía que proporcionaría la base de un dispositivo computacional de multiprogramación de tiempo compartido y propósito general llamado TSS/360, manejado independientemente del primero, pero debido a las dificultades que presentaba este nuevo proyecto, CP/CMS acaba por reemplazarlo en el System 360/67.

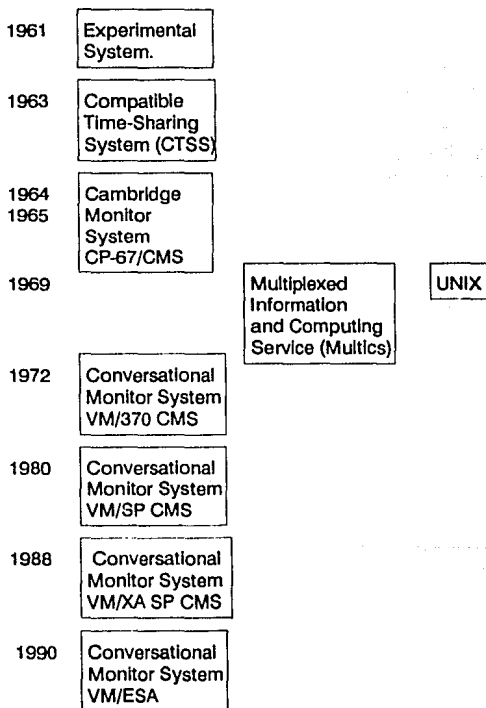
CP-67/CMS extendió las facilidades de CTSS. Se añadieron nuevos comandos y opciones a los ya existentes; los nombres y clases de archivos pasan a ser los nombres y tipos de archivos. El sistema es dividido en un manejador de recursos reales llamado CP y el uso de una interfase llamada CMS. Los servicios de CMS son solicitados vía SVC 202 y las funciones y comandos de CP a través de la instrucción DIAGNOSE X '8' (o una instrucción de SIO).

CTSS, usado con éxito en el M.I.T. durante 1974, influyó de manera decisiva en el diseño de CP/CMS. El grupo que desarrolló CTSS siguió adelante con el diseño del sistema Multics (Multiplexed Information and Computing Service), que nunca consiguió una gran aceptación comercial. En 1969, se empieza a conocer el sistema UNIX, desarrollado por Bell Telephone Laboratories como una alternativa para Multics y una muy moderna implementación de CTSS.

El primer release de Virtual Machine Facility/370 (VM/370) se obtiene en 1972 y soporta el System/370 Extended Architecture (XA); un año después se obtiene el primer release de Virtual Machine/System Product (VM/SP), desarrollándose después, seis versiones más hasta 1989; VM/XA SP R1.0 se conoce a principios de 1988 y en 1990 se anuncia la versión Virtual Machine / Enterprise Systems Architecture (VM/ESA) para el nuevo sistema 390.

VM/370 CMS añade, cambia y consolida los nombres de los comandos, tratando de uniformarlos. en cambio VM/SP CMS continuó introduciendo nuevos comandos y funciones (nuevos editores y ejecutores).

A continuación ofrecemos un cuadro cronológico que describe específicamente el desarrollo de estos sistemas.



VM tiene varias versiones, que se introdujeron al mercado conforme su evolución y las describimos en la siguiente tabla:

VM/XA	Virtual Machine/Extended Architecture	Nueva evolución de VM/SP.
VM/SP	Virtual Machine/System Product	Seguidos por un número de release.
VM/370	Virtual Machine/370	La versión original para mainframe VM.
VM/E	Virtual Machine/Entry	Una versión entry-level de VM para pequeñas mainframes.
VM/IS	Virtual Machine/Integrated System	Otras versiones entry-level de VM introducido con la serie 9370 de computadores.
VM/PC	Virtual Machine/Personal Computer	Una versión de VM para microcomputadores.

La versión VM/PC es una variante de VM/CMS, adaptada para ser usada en un microcomputador de la familia de IBM PC, como AT/370 y XT/370.

Por otra parte, entre los sistemas por lotes o de usuarios interactivos sencillos, podemos citar a DOS/VSE y MVS (OS/VS2), los cuales nacen independientemente de VM, pero son compatibles y en su combinación con este se logra la optimización del trabajo.

Los dos contendientes principales de IBM entre los sistemas operativos interactivos para computadores a gran escala son el TSO (la opción de tiempo compartido) de MVS y el VM. Sin duda alguna, VM es el sistema interactivo más completo, tanto en el aspecto de capacidades que proporciona como en el puro rendimiento. En un sistema de computación dado, suele suceder que VM puede soportar el doble de usuarios interactivos que MVS.

MVS ha sido durante años el principal sistema operativo de procesamiento por lotes de IBM. La flexibilidad de VM es importante, debido a que MVS puede operar en ella.

IBM también creó el DOS/VSE (en disco) para sus instalaciones más pequeñas. DOS es un sistema sencillo comparado con MVS. Contiene facultades consistentes de procesamiento por lotes, pero sus capacidades interactivas son débiles. El usuario de DOS puede cambiar a un procesador más poderoso que opere VM, generando de inmediato dos ventajas importantes: interactividad y la ejecución rápida de trabajos.

En seguida, se presenta un cuadro que muestra comparativamente la demanda que han tenido estos tres sistemas operativos en los últimos años.

	1985	1986	1987	1988	1989	1990
MVS	8,570	9,898	11,614	13,579	15,576	17,864
VSE	10,301	10,934	12,105	13,568	14,332	13,966
VM	6,888	7,940	9,995	13,196	15,928	17,571

La popularidad de VM ha ido en constante aumento y son muchas las razones para creer que seguirá haciéndolo. IBM y otros vendedores han estado sacando al mercado constantes mejoras del rendimiento. El concepto del sistema operativo de máquina virtual ha demostrado valer la pena. VM, se elaboró en un "pequeño taller" por diseñadores que vieron las ventajas de un concepto único, quienes no anticiparon el éxito que tendrían.

1.2 DEFINICION.

Un sistema operativo es un programa que controla y administra los recursos del computador, tales como procesador, memoria principal, dispositivos de almacenamiento, dispositivos de entrada y salida, así como el manejo de archivos, permitiendo el uso óptimo del computador. De aquí que se denominan también "programas de control".

Uno de los diversos sistemas operativos es el VM/CMS (Virtual Machine/Conversational Monitor System), también conocido con el nombre de VM/SP (Virtual Machine/System Product) o VM/370, actualmente instalado en la IBM-4381, de la Dirección General de Servicios de Cómputo Académico (DGSCA).

El término VM/CMS, se refiere a un sistema operativo interactivo conversacional y de múltiple acceso para computadores centrales (mainframes), minicomputadores y microcomputadores.

Interactivo conversacional, es decir, existe comunicación entre dos medios: los usuarios y VM; es además de múltiple acceso (dos o más personas pueden utilizar el sistema al mismo tiempo).

Podemos enumerar a continuación algunas de las ventajas que ofrece VM:

- El usuario final se comunica con el sistema utilizando terminales y no necesita ser especialista en computación para trabajar con él.
- Los usuarios utilizan los recursos del sistema por terminales en línea.
- Provee compatibilidad entre mainframes y microcomputadores.
- Soporta un gran número de terminales.
- Provee compatibilidad con otras familias de sistemas operativos soportados por IBM.
- Provee un ambiente donde otros sistemas operativos, como es el VM/CMS, pueden trabajar simultáneamente en un computador sencillo.

1.3 ¿QUE ES UNA MAQUINA VIRTUAL?

Una máquina virtual es el equivalente funcional de un sistema de cómputo real, el sistema operativo simula que los recursos de cómputo reales son de uso exclusivo del usuario que se encuentra en su terminal pudiendo, a través de esta, utilizar todos los recursos del sistema incluyendo los dispositivos de entrada y/o salida.

La máquina virtual está constituida por una terminal real (consola virtual), procesador, almacenamiento virtual de acceso directo, dispositivos virtuales de entrada/salida y un sistema operativo.

1.4 COMPONENTES DE VM.

El sistema operativo VM, está constituido básicamente por dos elementos:

- CONTROL PROGRAM (CP).
- CONVERSATIONAL MONITORING SYSTEM (CMS).

A su vez, cada uno de éstos es un subsistema que tiene sus propios comandos e instrucciones.

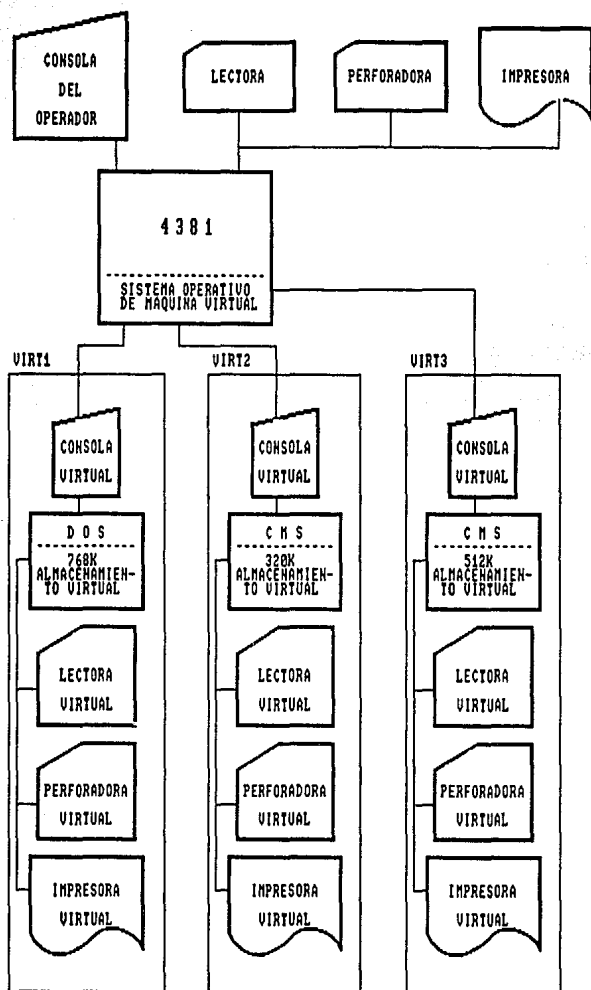


FIGURA 1. VARIAS MAQUINAS VIRTUALES CREADAS POR UNA SOLA MAQUINA REAL.

Otros subsistemas que forman parte importante de VM, son:

- REMOTE SPOOLING SUBSYSTEM (RSCS).
- GROUP CONTROL SYSTEM (GCS).

1.4.1 CONTROL PROGRAM.

El CP es el encargado de manejar los recursos reales del sistema, proporcionando a cada usuario un ambiente de trabajo individual y personalizado, el cual es el equivalente funcional de un computador real. En términos generales, se encarga de controlar los recursos físicos de la máquina durante una sesión de trabajo en VM, asignando a los usuarios de las máquinas virtuales: cierto tiempo de ejecución del procesador, segmentos de la memoria real y algunos dispositivos de entrada y salida, de tal forma que cada usuario trabaja separada e independientemente de los otros. CP multiprograma máquinas virtuales completas, en lugar de tareas y procesos individuales.

Los procesos ejecutados sobre una máquina virtual no son controlados por CP, sino por el sistema operativo que se ejecuta en esa máquina virtual. Las máquinas virtuales operan independientemente unas de otras y los conflictos los resuelve el CP.

Los sistemas operativos que se ejecutan en máquinas virtuales realizan toda una gama de funciones como: administración del almacenamiento, construcción del itinerario del procesador, control de entrada/salida, protección entre los usuarios, protección del sistema operativo de los usuarios, spool, multiprogramación, control de trabajos y procesos, manejo de errores.

1.4.2 CONVERSATIONAL MONITOR SYSTEM.

El CMS es el segundo componente principal del VM es, en sí mismo, un sistema operativo que trabaja sólo bajo CP. Nos comunicamos con CMS mediante comandos y responde mediante mensajes.

Todos los mandatos CMS disponibles para el usuario se almacenan en archivos de disco; en realidad no son parte del sistema operativo. Esto facilita a los usuarios la adaptación del conjunto de instrucciones de sus máquinas virtuales a sus propias necesidades, añadiendo, borrando o cambiando los archivos de mandatos.

CMS permite realizar varias funciones, como:

- Escribir, probar y ejecutar programas de aplicación en diversos lenguajes de programación.
- Crear y modificar archivos.
- Enviar programas para ser ejecutados en lotes (batch).
- Comunicarse con otros usuarios del sistema VM/CMS.

La instalación de lotes CMSBATCH, permite al usuario de terminal enviar trabajos más largos para su procesamiento por lotes, mientras continúa su trabajo interactivo con una máquina virtual CMS. El lote CMS corre en su propia máquina virtual, controlada por el operador del sistema. Las instrucciones de control de trabajos definen el procesamiento que va a ejecutar. Los trabajos se introducen en un formato de imagen de tarjeta. Las salidas de trabajos pueden enviarse a los dispositivos de salida reales del sistema o a los dispositivos de salida de la máquina virtual asociada al usuario que envió el trabajo. La instalación de lotes CMS notifica de su terminación al usuario que envió el trabajo.

1.4.3 REMOTE SPOOLING COMMUNICATIONS SUBSYSTEM.

El subsistema de comunicaciones y spooling remoto es el encargado de controlar la transferencia de información entre computadores y estaciones de trabajo remotas en redes de comunicaciones.

Opera en una máquina virtual creada por CP y usa enlaces de comunicación y terminales remotas para transmitir los archivos, proporciona un medio apropiado para la transferencia de archivos entre máquinas virtuales en la misma máquina real, además de una transmisión más convencional entre sistemas de computación y estaciones de trabajo remotas.

Para transmitir datos de un usuario a otro, se debe indicar a RSCS la localización e identificación del usuario receptor así como el nombre del archivo que se envía.

1.4.4 GROUP CONTROL SYSTEM.

El GCS es un supervisor de máquinas virtuales, que se emplea exclusivamente para control de subsistemas que soportan redes de arquitectura SNA (Systems Network Architecture).

GCS es una interfaz entre programas productos como VTAM, RSCS y el CP. Provee:

- Un área de almacenamiento común donde un grupo de máquinas intercambia información.
- Un espacio privado de almacenamiento para GCS que contiene los bloques de control y datos para el uso del supervisor, así como también un archivo de configuración que describe al grupo de máquinas virtuales.
- Servicio de multitarea para manejar varias operaciones activas al mismo tiempo.
- Servicios de entrada y salida (I/O) generales con programas que permiten definir dispositivos de I/O para GCS, construir programas de canal (reales o virtuales), y empezar operaciones de I/O y para rutinas que manejan interruptores para dispositivos de I/O.

1.5 CONFIGURACION.

Una máquina virtual tiene los mismos componentes de una máquina real, pero en este caso, simulados por el CP. Podemos citar los principales :

1.5.1 Procesador.

- El CP simula el procesador asignando fracciones de tiempo del procesador real. Desde el punto de vista del usuario, su procesador es idéntico a un procesador de arquitectura 370.

1.5.2 Memoria.

- Tenemos la memoria virtual, que simula una memoria real. El sistema operativo se encarga de dividir los programas en páginas de memoria, las cuales son almacenadas en áreas de disco magnético controladas por el CP y traídas a la memoria virtual con base en las demandas del programa que corre en la máquina virtual.

1.5.3 Dispositivos de Entrada y Salida.

- Una máquina virtual puede tener asignados dispositivos de entrada/salida virtuales, como:

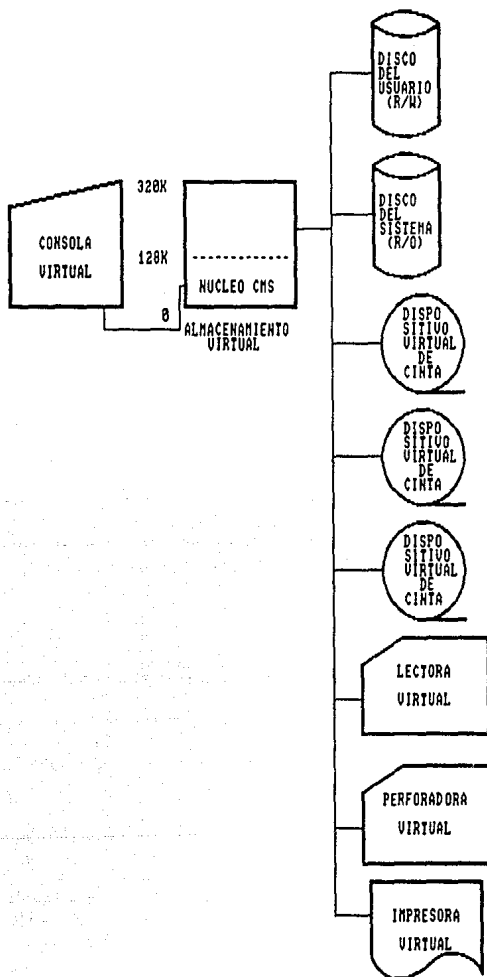


FIGURA 2. MAQUINA VIRTUAL DEDICADA VISTA POR UN USUARIO DE CMS.

1.5.3.1 Consola.

- La consola de la máquina virtual está simulada en la terminal en la que trabaja el usuario. Mediante comandos, el CP simula todas las funciones que se pueden realizar en la consola de un procesador.

1.5.3.2 Minidiscos.

- Son fracciones de discos reales de dispositivos de almacenamiento de acceso directo (DASD). Cada minidisco es un subconjunto de un disco completo, que consiste de varios cilindros.

Los minidiscos pueden ser propios, cuando están definidos en la configuración de la máquina virtual, o son minidiscos ligados cuando en realidad pertenecen a otra máquina virtual, permitiendo compartir información entre varios usuarios. Generalmente un usuario tiene uno o dos minidiscos propios, pero se puede ligar a minidiscos del sistema donde se encuentran los programas que constituyen el CMS, así como compiladores y programas de aplicación, accediéndolos sólo en modo de lectura.

1.5.3.3 Lectora y Perforadora de tarjetas virtuales.

- Estos dispositivos, son ampliamente usados en el sistema VM/CMS, como medio de comunicación entre máquinas virtuales. Cuando un usuario envía información a otro, esta es colocada primero en la perforadora virtual del usuario que envía y después transferida a la lectora virtual del usuario receptor.

Estas lectoras y perforadoras virtuales son simuladas en archivos de discos magnéticos mediante la técnica de spooling.

1.5.3.4 Impresora virtual.

- Este dispositivo se maneja también mediante la técnica de spooling. Prácticamente todas las máquinas virtuales asignadas tienen una o varias impresoras virtuales.

1.5.3.5 Unidades de cintas magnéticas.

- Las unidades de cintas magnéticas son dispositivos dedicados de uso individual, para las máquinas virtuales. Sirven en general para respaldar información contenida en minidiscos o almacenar datos y programas que no son de uso diario y restarían espacio disponible en los DASD.

Estas unidades de almacenamiento, son asignadas a una máquina virtual mediante comandos que el operador del sistema teclea desde la consola real, a petición del usuario y por tiempo limitado.

1.6 SISTEMAS DE COMUNICACION DEL VM/CMS.

1.6.1. RSCS.

RSCS es una abreviación de **Remote Spooling Communications Subsystem Networking**, nombre que a su vez lo describe.

Remote: sistema separado o dispositivo al que se tiene acceso a través de una línea de telecomunicaciones.

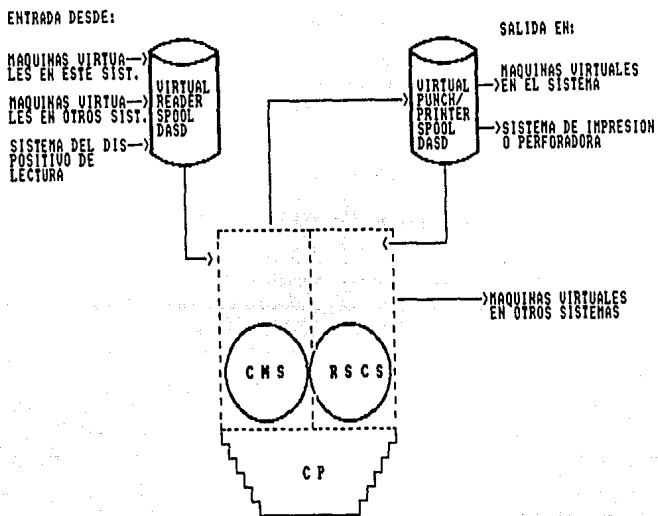


FIGURA 3. MAQUINA VIRTUAL I/O.

Spooling: almacenamiento y recuperación de datos, con ayuda de dispositivos de acceso directo (DASD).

Communications: intercambio de información entre sistemas y dispositivos.

Subsystem: un sistema que se ejecuta bajo el control de otro sistema.

Networking: sistema de procesamiento de datos y dispositivos conectados con otros.

De aquí, se concluye que RSCS, es un sistema que trabaja bajo el control de otro. Este puede almacenar y recuperar datos con el auxilio de los dispositivos DASD e intercambiarlos con otros dispositivos o sistemas, aunque sean distantes.

RSCS, se ejecuta bajo GCS (Group Control System). Usa las facilidades del spooling de VM para guardar y recuperar datos. A su vez, VM maneja la transferencia de datos dentro del sistema RSCS mediante las áreas spool. RSCS amplía las capacidades del spooling para VM, manejando la transferencia de datos entre el sistema VM y sus recursos externos.

Los datos son guardados en un área spool, después RSCS los retiene hasta poder enviarlos a su destino, usando equipo de comunicaciones para transferir los datos desde donde están hasta donde sean requeridos (entre el sistema y otros sistemas o localizaciones remotas). Para el manejo de archivos spool depende de las etiquetas, las cuales contienen información que describe el origen y destino del archivo enviado.

1.6.2 ACF/VTAM.

ACF/VTAM (Advanced Communications Function / Virtual Telecommunication Access Method), es la base de los subsistemas de comunicación de datos utilizados en el VM/CMS. VTAM, junto con NCP, son en realidad el sistema operativo de la red de telecomunicaciones.

Sus funciones son parecidas a la de los sistemas operativos del procesador central, en cuanto al control de recursos y manejo de las peticiones de los usuarios a la red.

VTAM soporta diferentes aplicaciones de telecomunicaciones concurrentes y controla cómo se comparten los recursos de la red, terminales, controladores y programas entre los diferentes usuarios, que pueden estar en uno o varios procesadores centrales.

También provee facilidades para establecer dinámicamente sesiones de trabajo entre terminales y programas de aplicación, como un mecanismo de autorización apropiado.

1.6.3 NCP/EP.

NCP (Network Control Program) reside en los controladores 3720 y 3725, proporcionan las funciones de manejo físico de la red de telecomunicaciones.

La función principal del NCP es controlar las líneas y terminales de comunicaciones conectadas al controlador, llevar a cabo la recuperación de errores y controlar el flujo de información a través de la red.

El NCP se comunica con VM/CMS, a través del método de acceso VTAM.

EP (Emulation Program) es un programa se ejecuta en los controladores 3720 o 3725 ya sea por sí mismo o conjuntamente con NCP, permitiendo la definición de protocolos BSC y asíncronos, a través del programa de control de la red, soportando terminales no-SNA.

1.6.4 X.25 NPSI.

X.25 NPSI (X.25 Network Packet Switching Interface), permite a los usuarios de una red controlada por VTAM y NCP, establecer comunicaciones a través de redes de transmisión de datos conmutado de paquetes (packet-switched).

Permite a procesadores de IBM trabajar de acuerdo a la arquitectura SNA (System Network Architecture), comunicarse con equipos SNA o no-SNA, utilizando este tipo de redes.

X.25 NPSI, causa que una red de tipo packet-switching (conmutación de paquetes) aparezca al NCP y su procesador central asociado ejecutando VTAM, como una serie de líneas de comunicación conmutadas o no-conmutadas que cumplen con arquitectura SNA.

1.6.5. TCP/IP.

TCP/IP (Transmission Control Protocol/Internet Protocol), es un programa formado por los protocolos TCP e IP.

TCP es usado por la Agencia de Proyectos de Investigación Avanzada (ARPA), Internet y una red del Departamento de Defensa de E.E.U.U.. Este protocolo estándar provee una forma de transportar servicios entre "host" en paquetes conmutados de redes de comunicación de computadores y en sistemas interconectados a estas redes.

IP, provee la interfase desde un protocolo de alto nivel a la red local de protocolos (LAN). TCP asume que IP es un protocolo de red subalterno.

TCP/IP, permite en terminales remotas, transferir archivos, manejar correo electrónico y transmitir bloques de datos o paquetes de bits, proporcionando a VM/CMS la capacidad de participar en una red INTERNET, utilizando protocolos como: Token-Ring, Ethernet, Pronet y DDN X.25.

1.6.6 VM PASS-THROUGH FACILITY.

Permite establecer una nueva sesión, es decir, accesa el sistema mediante una clave de usuario diferente, sin necesidad de interrumpir la sesión actual mediante comandos de CP.

Al invocar Pass-Through Facility, se crea, utilizando facilidades del programa de control CP, un dispositivo lógico que despliega otro logotipo de VM, en el cual es posible entrar en sesión con otra clave de usuario.

1.6.7 3270 PC FILE TRANSFER.

Este Programa Emulador 3270, permite usar el computador en una red de computadores "host". Esto significa que una IBM PC puede comunicarse con un computador "host" y posiblemente con otros computadores, incluyendo otras PC's IBM.

Específicamente, una IBM PC con el Emulador 3270 puede:

- Establecer una sesión de comunicación, emitida como una sesión 3270, con un "host".
- Solicitar acceso a los archivos del procesador central.
- Transmitir un archivo de PC al procesador central.
- Actualizar un archivo del procesador central.
- Transmitir del "host" a la PC y almacenar los cambios de un archivo.
- Enviar mensajes al procesador central.
- Imprimir archivos del procesador central.

La IBM PC también puede comunicarse directamente con otras PC's, en vez de la "host", si es parte de una LAN (Red de Area Local), lo que requiere software y hardware adicional al Emulador 3270.

De igual manera, una IBM PC puede ser una estación de trabajo única, que a diferencia de las anteriores no tiene acceso a otras PC's o terminales, simplemente se comunica con el procesador central.

1.7 ALGUNAS APLICACIONES DEL VM/CMS.

En los últimos años, se han desarrollado diversas aplicaciones, siendo algunas de ellas:

1.7.1 SISTEMAS DE AUTOMATIZACION PARA OFICINAS.

1.7.1.1 DCF (Document Composition Facility).

Es un procesador de texto muy poderoso que permite desde la creación de documentos simples hasta libros enteros. Incluye opciones avanzadas de formato e inclusión de gráficas en los documentos.

1.7.1.2 DISPLAY WRITE / 370.

Es un procesador de texto el cual, además de las facilidades de edición, permite incluir en los documentos gráficas e imágenes.

DW/370 trabaja con menús. Revisa ortografía y ayuda a corregirla dando una serie de sinónimos en español.

Se pueden visualizar dos documentos al mismo tiempo, e incluso editar uno mientras se ve el otro. Cuenta con múltiples tipos de caracteres, así como muchas otras facilidades.

1.7.1.3 OFFICE VISION Y PROFS (Professional Office System).

Son sistemas de IBM que automatizan las tareas de oficina, facilitando y optimizando el trabajo, de manera que se reduzcan los errores humanos por descuido u olvido de información.

Simplifica la creación de documentos, como cartas y memorándums, sin tener que darles formatos uno por uno. Además con el uso de directorios de personal y clientes, se pueden emitir fácilmente muchos documentos en un corto tiempo. Con el uso de RSCS, se envían y reciben documentos, con una completa información acerca de su autor, fecha y objeto de estos.

Ayuda a organizar tareas administrativas en forma automática como: correo electrónico; planificación de conferencias mediante la consulta de agendas tanto de los asistentes como de las salas disponibles, envío de avisos importantes (los cuales son notados por el usuario al entrar a su máquina virtual mediante un sonido que lo anuncia).

1.7.2 GRAFICACION Y DISEÑO POR COMPUTADORA.

1.7.2.1 GDDM (Graphical Data Display Manager).

Es un producto que permite utilizar programas de elaboración de gráficas en terminales de la familia 3270 (equipadas para gráficas), del sistema 5080, de digitalizadores (scanners) e impresoras de gráficas.

Consiste, principalmente, de un conjunto de subrutinas, que proporcionan las funciones necesarias para controlar los dispositivos de graficación, estas pueden ser invocadas por programas en Fortran o PL/I.

1.7.2.2 CADAM (Computer-Graphics Augmented Design and Manufacturing).

CADAM, es un conjunto de programas que proporcionan funciones de gráficas interactivas para CAD/CAM, Diseño y Manufactura Asistidos por Computadora, en dos y tres dimensiones de alta calidad y fáciles de aprender. Requiere para su operación de las terminales especiales 5080.

Los programas actualmente instalados en DGSCA, permiten la obtención de gráficas en dos y tres dimensiones. Posee un manejador de base de datos que permite almacenar y recuperar gráficas producidas por los demás programas, además del soporte para el graficador (plotter) 7375 que permite la impresión de las gráficas.

1.7.2.3 CAEDS (Computer Aided Engineering Design System).

CAEDS, es un paquete integrado de aplicaciones para automatizar las funciones del Análisis de Ingeniería y de Modelaje de sólidos. Requiere de terminales 5080.

Los programas de CAEDS actualmente instalados permiten elaborar y archivar las gráficas, cuenta con herramientas interactivas de modelaje de sólidos en tres dimensiones para el diseño de componentes mecánicos, ensamblaje y análisis de objetos múltiples, análisis de la conducta cinemática de los sistemas, facilidades para el análisis de elementos finitos, creación de nodos y elementos, así como la preparación del modelo para análisis.

1.7.2.4 CIEDS/CBDS (Computer-Integrated Electrical Design Series/Circuit Board Design System)

CIEDS/CBDS, es una herramienta de diseño gráfico e interactivo que facilita el diseño y la manufactura de circuitos impresos. Consiste de dos módulos principales que proporcionan las facilidades necesarias para la creación de diseños de circuitos electrónicos, además de la verificación automática de costos.

1.7.3 BASES DE DATOS.

1.7.3.1 SQL/DS (Structured Query Language/Data System).

SQL/DS, es manejador de bases de datos relacional con facilidades integradas para consultas y escritura de reportes. SQL/DS, es muy conveniente para aplicaciones en las que se requieren consultas, generación automática de informes y actualizaciones a bases de datos. Adicionalmente SQL bajo VM, proporciona la capacidad de poder acceder bases de datos que físicamente se encuentren en un procesador diferente.

El SQL puede ser utilizado desde programas escritos en una variedad de lenguajes como FORTRAN, BASIC y REXX.

1.7.3.2 QMF (Query Management Facility).

QMF, es un programa diseñado para proporcionar facilidades de acceso a bases de datos de manera interactiva a usuarios con poca o nula experiencia en computación. El usuario puede hacer consultas a bases de datos relacionales creadas mediante SQL/DS (requisito para QMF), de una manera muy sencilla y posteriormente utilizar los resultados de estas consultas para crear reportes, que pueden ser extendidos mediante la inclusión de títulos y subtítulos, cortes de control y totales.

Adicionalmente, el QMF tiene una interfase con el programa GDDM ICU, que permite la creación, despliegue e impresión de gráficas para presentar la información obtenida de las bases de datos relacionales.

1.7.4 INTELIGENCIA ARTIFICIAL.

Para el desarrollo de inteligencia artificial se cuenta con un ambiente de sistema expertos.

El Expert System Environment, está constituido por un par de programas complementarios: ESDE (Expert Systems Development Environment) y ESCE (Expert Systems Consultation Environment). El primero se utiliza para construir la aplicación, mientras que el segundo proporciona los servicios necesarios para ejecutar la aplicación de un Sistema Experto.

1.7.5 MATEMATICAS Y ESTADISTICA.

1.7.5.1 ACRITH (Accuracy Arithmetic Subroutines).

El ACRITH es una biblioteca de rutinas aritméticas de alta precisión, que permite realizar cálculos con muy alto grado de precisión, verificando a la vez la validez de sus resultados.

El programa ACRITH consiste de:

- Rutinas que simulan ciertas instrucciones de los procesadores IBM de arquitectura 370.
- Conversiones de formato hexadecimal a decimal y viceversa.
- Adición y sustracción de vectores y multiplicación y división de un vector por un escalar.
- Cálculo del producto escalar de dos vectores.
- Multiplicación de matrices.
- Evaluación de expresiones aritméticas.
- Evaluación de polinomios y expresiones aritméticas.
- Computación de ceros simples de polinomios.
- Solución de ecuaciones lineales.

1.7.5.2 MPSX (Mathematical Programming System).

El MPSX ayuda a la resolución de aplicaciones comunes de programación matemática. Permite resolver problemas de programación lineal, de enteros mixtos y separables, utilizando los métodos simplex y revisado con variables libres o restringidas.

1.7.5.3 SAS (Statistical Analysis System).

Es uno de los programas de estadística más utilizado, desarrollado originalmente en la Universidad de Carolina del Norte y distribuido por el SAS Institute en Raleigh, Carolina del Norte. Proporciona facilidades para el análisis estadístico, herramientas para recuperación y manejo de datos, programación, generación de gráficas y reportes, etcétera..

1.8 LENGUAJES DE PROGRAMACION EN VM/CMS.

Se cuenta con una variedad de lenguajes de programación instalados actualmente en DGSCA, como:

- PL/I OPTIMIZING COMPILER, apropiado para escribir aplicaciones muy grandes y complejas.
- PASCAL/VS, un compilador para un lenguaje PASCAL que soporta los estándares propuestos a la International Standards Organization, así como varias extensiones a los mismos. El compilador incluye así mismo una capacidad de depuración interactiva simbólica (Symbolic Interactive Debugger).

- VS FORTRAN Compiler and Library, soporta y proporciona compatibilidad con programas antiguos utilizando el American Standard FORTRAN X3.9-1966, así como varias extensiones de IBM al lenguaje.

- APL Versión 2, lenguaje de tipo interpretativo de alta productividad, utiliza como parte del lenguaje una serie de símbolos especiales, por lo que se requiere un teclado especial en la terminal.

- IBM BASIC, una implantación poderosa del lenguaje que tiene tanto las ventajas de intérprete como las de compilador.

- LISP/VM es un lenguaje de programación de alto nivel diseñado principalmente para desarrollar aplicaciones de Inteligencia Artificial en áreas de Sistemas Expertos, procesamiento de Lenguaje Natural, Robótica y otras tecnologías avanzadas.

- PROLOG, uno de los lenguajes más conocidos en Inteligencia Artificial, de tipo declarativo, es decir que dados los hechos y reglas necesarios, puede usar razonamiento deductivo para resolver problemas de programación.

- IBM C/370, ofrece ventajas como versatilidad, portabilidad, flexibilidad, rapidez y mantenimiento.

1.9 EL EDITOR DE TEXTOS XEDIT

VM System Productor Editor, conocido generalmente como XEDIT, es el editor de CMS; flexible, sencillo y de fácil manejo, que nos permite trabajar en la técnica "full screen", es decir, con una pantalla de datos desplegados al mismo tiempo en la terminal, a diferencia de otros editores que trabajan con una sola línea a la vez.

Es un editor muy poderoso que además de permitir la creación y modificación de archivos, editar varios archivos simultáneamente; combinar archivos o porciones de archivos, utilizar marcos para llevar a cabo funciones complejas con base en subcomandos.

1.10 LENGUAJE INTERPRETE DE COMANDOS REXX.

CMS ofrece tres diferentes lenguajes de comandos, EXEC, EXEC2 y el "System Product Interpreter, REXX (REstructured eXtended eXecutor). Los procedimientos de comandos escritos en estos tres lenguajes son introducidos en archivos de CMS de tipo EXEC.

Los lenguajes EXEC2 y EXEC están estrechamente relacionados. EXEC2 incluye varios comandos similares, así como variables predefinidas y funciones como en el lenguaje EXEC. Se puede decir que EXEC2 pasa a mejorar algunas dificultades que ofrecía EXEC.

REXX, es el lenguaje de procedimientos y comandos más reciente para VM/CMS. Los elementos que lo componen son muy diferentes que los de EXEC2 y EXEC, además tiene características más funcionales.

Este lenguaje de programación es extremadamente versátil. Características tales como la comodidad de uso y formatos libres, hacen que sea un excelente lenguaje para principiantes y usuarios en general, al igual que permite aplicaciones profesionales porque hace uso de comandos para diferentes ambientes, tiene funciones muy poderosas y capacidades matemáticas extensas.

Dedicamos el siguiente capítulo a REXX, para explicar detalladamente sus características y hacer más comprensibles los programas escritos en este lenguaje, los cuales se utilizan en el sistema que se diseña en el presente trabajo.

CAPITULO 2

LENGUAJE DE PROGRAMACION REXX

2.1 ¿QUE ES UN PROGRAMA REXX?

REXX es un lenguaje intérprete de comandos el cual permite crear programas con características especiales, que realiza de manera sencilla algún tipo de tarea.

Los comandos pueden ser tanto de CP como de CMS, incluyendo además instrucciones propias del lenguaje.

Los archivos de REXX, son creados usando XEDIT por lo cual también incluyen subcomandos de este editor de textos, considerándose generalmente de tipo EXEC.

2.2 CARACTERISTICAS.

2.2.1 FACILIDAD DE USO.

El lenguaje REXX es fácil de leer y escribir, porque muchas instrucciones son palabras significativas en Inglés.

A diferencia de otros lenguajes que usan abreviaciones, en REXX tenemos palabras completas y sencillas como: SAY, PULL y EXIT.

2.2.2 FORMATO LIBRE.

REXX tiene muy pocas reglas para sus formatos; no es necesario colocar las instrucciones en columnas particulares, se pueden tener espacios en una línea o entre líneas, acepta escribir una instrucción en varias líneas o múltiples instrucciones en una sola línea, las variables no necesitan estar pre-definidas y además las instrucciones se escriben en mayúsculas y/o minúsculas.

2.2.3 FUNCIONES CONVENIENTEMENTE INTEGRADAS.

REXX proporciona funciones integradas que ejecutan varios procesos, examinando y comparando operaciones para textos y números. Otras funciones integradas proveen capacidades de formato y cálculos aritméticos.

2.2.4 FÁCIL RASTREO DE ERRORES.

Cuando un programa REXX contiene un error, se despliegan en pantalla mensajes con explicaciones significativas. Además, tiene una instrucción (TRACE) que facilita la depuración.

2.2.5 INTÉRPRETE.

REXX en un lenguaje intérprete, aunque cuenta de manera independiente con un compilador.

2.2.6 CAPACIDAD EXTENSIVA DE DESGLOSAMIENTO.

REXX incluye una capacidad extensiva de desglosamiento para la manipulación de caracteres. Esta capacidad permite levantar un modelo para separar caracteres, números y entradas mezcladas.

2.3 ESTRUCTURA Y SINTAXIS GENERAL.

Los programas escritos en REXX deben comenzar, como regla general, con un comentario. Por esta característica se distingue de los programas en lenguajes CMS EXEC y EXEC2.

Los comentarios están formados por una secuencia de caracteres, en una o más líneas, delimitadas por /* y */. Son ignorados por el procesador del lenguaje. Por ejemplo:

```
/* Este es un ejemplo de un comentario */
```

2.3.1 VARIABLES Y EXPRESIONES.

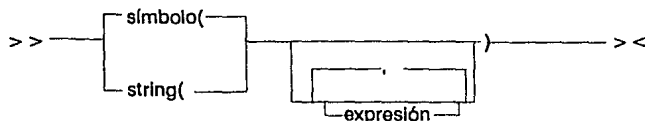
Los símbolos son grupos de caracteres ya sea alfabéticos, numéricos o alfanuméricos. Un símbolo que no empiece con un número, puede considerarse como una variable a la cual se le asignará un valor.

Un string es una cadena de caracteres que puede contener hasta 250 caracteres delimitados por " ó ', ejemplo:

```
'Sistema Operativo'
```

```
"No hay nada..."
```

Si después de un string aparece un '(', se considera como el nombre de una función, cuya sintaxis es:



O si tenemos una secuencia de uno o más dígitos entre 0-9,a-f,A-F, separados opcionalmente por espacios, delimitados por comillas e inmediatamente seguidos de una x o X se trata de un string hexadecimal, que no debe exceder de 250 bytes. Ejemplo:

```
'ABCD'x  
"1d cc f8"X
```

En VM/SP, cuando se produce un error en el proceso de un comando o en la ejecución de un programa, CMS contesta con un número de error y un breve mensaje que intenta dar idea de cual fue el error, además regresa entre paréntesis un código de retorno.

Los códigos de retorno permiten controlar algunos procesos al detectar fin de archivo, archivo no existente, etcétera.

2.3.2 OPERADORES.

Los operadores de concatenación se usan para combinar strings. Un espacio en blanco puede unir dos o más strings, si no se quieren espacios en blanco entre los strings se pueden escribir seguidos o usando el símbolo de concatenación '||'. Ejemplo:

'abc' || 'de' == > abcde

Los operadores aritméticos son los siguientes:

- + Suma
- Resta
- * Multiplicación
- / División
- % División que regresa sólo la parte entera del resultado.
- // División que regresa el residuo de la operación.
- ** Exponenciación.

Los operadores de comparación regresan un 1 si el resultado es verdadero y 0 si es falso, y son:

- = Igual que
- > <, < > Diferente que
- > Mayor que
- < Menor que
- > = Mayor e igual que
- < = Menor e igual que

Los operadores lógicos son:

- & AND
- Regresa 1 si ambos términos son verdaderos.
- | OR
- Regresa 1 si algún término es verdadero.
- && OR exclusivo
- Regresa 1 si algún término es verdadero, excepto cuando ambos son verdaderos.
- ¬ NOT lógico
- Niega, el 1 lo convierte en 0 y viceversa.

Una etiqueta es una cláusula que consiste de un símbolo simple seguido de dos puntos (:). Estas etiquetas sirven para identificar las instrucciones CALL, SIGNAL y el llamado a funciones internas.

2.4 ALGUNAS INSTRUCCIONES Y FUNCIONES BASICAS.

A continuación se enumeran algunas instrucciones básicas que se utilizan generalmente en programas REXX.

SAY

> _____ SAY _____ ; _____ > <

└───────────┬───────────┘
 expresion

El resultado de evaluar la expresión se despliega ya sea en pantalla o el dispositivo de salida al que se destine.

PULL

> _____ PULL _____ ; _____

└───────────┬───────────┘
 expresiones

Esta instrucción permite leer un string como variable de entrada a la memoria de la computadora, donde 'expresiones' es una lista de símbolos separados por blancos.

PARSE PULL

> _____ PARSE [UPPER] PULL _____ ; _____

└───────────┬───────────┘
 expresiones

PARSE PULL lee cada string del teclado tal cual se escribió, sin trasladarlo a mayúsculas. PARSE UPPER PULL traslada cada string leído del teclado a mayúsculas. Esta instrucción se abrevia usando nada más PULL.

TRACE

> _____ TRACE _____ ; _____

└───────────┬───────────┘
 opciones

Permite rastrear un programa para detectar errores. Tiene una gran variedad de opciones, pero las más usadas son:

Intermedio	Evalúa cada expresión y despliega en pantalla el resultado de cada operación.
Resultado	Cuando cada expresión ha sido evaluada, el resultado final se despliega en pantalla.
Normal	Sólo se despliega en pantalla los comandos rechazados por el intérprete.

LEFT ()

> _____LEFT(string,longitud) _____ ><

Regresa una parte del string, según la longitud que se especifique, tomando los caracteres de izquierda a derecha.

Ejemplo:

LEFT('abc',5) == > ' abc '

CENTER ()

> _____CENTER(string,longitud) _____ ><

Regresa el string centrado en la longitud que se especifica.

Ejemplo:

CENTER('abc',7) == > ' abc '

COPIES()

> _____COPIES(string,n) _____ ><

Regresa n copias concatenadas del string.

Ejemplo:

COPIES('*',5) == > '*****'

2.5 ESTRUCTURAS DE CONTROL.

Un programa es una simple lista de instrucciones o un cierto número de listas de instrucciones conectadas por otras instrucciones, con cierto orden de ejecución.

Tenemos cinco tipos de estructuras de control que a continuación se explican.

2.5.1 SELECCION.

De una lista de instrucciones se selecciona una para ejecutarla, usando las siguientes estructuras:

IF-THEN

IF expresión
THEN instrucción

El intérprete ejecuta la instrucción sólo si la expresión es verdadera.

IF-THEN-ELSE

```

IF expresión
THEN instrucción 1
ELSE instrucción 2

```

El intérprete ejecutará la instrucción 2 sólo si la expresión es falsa.

SELECT

```

SELECT
  WHEN expresión 1 THEN instrucción 1
  WHEN expresión 2 THEN instrucción 2
  WHEN expresión 3 THEN instrucción 3
  ...
  OTHERWISE
    instrucción
    instrucción
    instrucción
END

```

Si la expresión 1 es verdadera, se ejecuta la instrucción 1. El proceso continuará con la siguiente instrucción, después del END.

Pero si la expresión 1 es falsa, se checa la expresión 2, que en caso de ser verdadera se ejecuta la instrucción 2 y el proceso continúa con la instrucción después del END.

Si la expresión 1, expresión 2, expresión 3, ..., son falsas el proceso continúa con las instrucciones que siguen a OTHERWISE.

2.5.2 LOOPS.

Permiten repetir una lista de instrucciones, un número específico de veces, o si se cumple cierta condición.

```

DO
  DO nombre = expr1 [BY expr2] [TO expr3]
    instrucción 1
    instrucción 2
    instrucción 3
  ...
END

```

Donde:

nombre es una variable de control, cuyo valor cambia cada vez que se regresa al loop.

expr1 indica el valor inicial que tiene la variable de control, cuando se ingresa por primera vez al loop.

expr2 indica la cantidad con la que se incrementará la variable de control, cada vez que se regrese al loop. Si no se especifica, los incrementos serán de 1 en 1.

expr3 indica el valor máximo que alcanzará la variable de control, después de haber regresado al loop.

DO FOREVER.

```

DO FOREVER
  instrucción 1
  instrucción 2
  IF expresión THEN LEAVE
    instrucción 3
    instrucción 4
END
instrucción
instrucción
....

```

Al ingresar a este loop, se ejecutan las instrucciones 1 y 2, se evalúa una expresión, en caso de ser verdadera salimos del loop y se ejecutan a continuación las instrucciones después del END. Pero mientras la expresión resulte ser falsa se ejecutan las instrucciones 4, 5 y 6 y continuamos dentro del loop.

DO WHILE

```

DO WHILE expresión
  instrucción 1
  instrucción 2
  instrucción 3
END

```

Mientras la expresión sea verdadera se ejecutan las instrucciones 1, 2 y 3, en caso contrario salimos del loop.

DO UNTIL

```

DO UNTIL expresión
  instrucción 1
  instrucción 2
  instrucción 3
END

```

Si la expresión es falsa se ejecutarán las instrucciones 1, 2 y 3 hasta que sea verdadera.

2.5.3 SUBROUTINAS Y FUNCIONES.

Una subrutina es un programa en REXX de menor rango, que puede ser invocada en varios puntos del programa principal. Para llamar una subrutina se usa la instrucción CALL

```
CALL nom_rutina [argumento 1, ...]
```

Una función es también un pequeño programa en REXX, que realiza cierto proceso, que puede ser repetitivo. REXX tiene sus propias funciones definidas, pero también se pueden construir otras en base a las necesidades del usuario. Para utilizar una función, basta invocarla por su nombre e indicar los parámetros, es decir:

nom_rutina([argumento 1, ...])

Las subrutinas pueden ser internas y externas. Una subrutina es interna si va incluida dentro del programa principal, como indica la Figura 4.

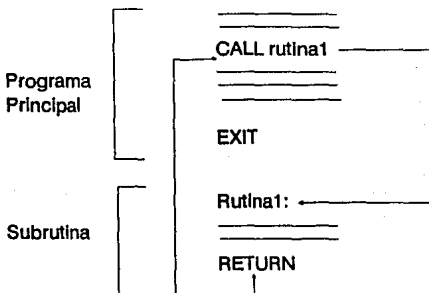


Figura 4.

Una subrutina externa será un archivo EXEC, con todas las características de un programa REXX, donde las variables no se especifican al invocarla, como se ilustra en la Figura 5.

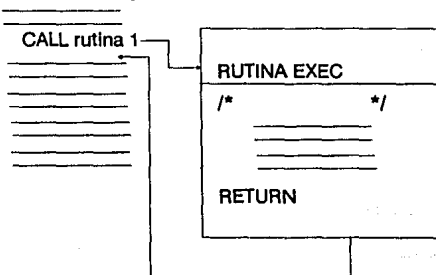


Figura 5.

2.5.4 JUMPS.

Los jumps son instrucciones que permiten transferir el control a un punto diferente del programa, no es aconsejable usarlo en loops.

SIGNAL es una instrucción que causa un cambio anormal en el flujo de control del programa.

En caso de que el intérprete lo encuentre en medio del programa, ignorará todas las construcciones de **SELECT** y **loops** que encuentre después. Para otros propósitos, es mejor construir jumps usando **IF**, **SELECT** o **DO**.

SIGNAL etiqueta

Donde, "etiqueta" es un simbolo seguido por dos puntos que se coloca en algún lugar del programa, después del cual se encuentran otras instrucciones desde las cuales continúa el proceso. Por ejemplo:

```
SIGNAL interrumpe
...
EXIT
/*-----*/
/* Código para un final anormal */
/*-----*/
ABEND:
say "Fin anormal. No puede continuar"
exit
```

2.6 MANEJO DE PILAS.

En REXX, se pueden usar pilas de programas como una especie de buzones, pueden ser accedidos usando generalmente instrucciones de REXX y comandos de CMS. Los comandos de CMS, por ejemplo, pueden introducir datos y con una instrucción del intérprete se pueden recuperar, o al contrario una instrucción de REXX puede introducir datos y un comando de CMS los recupera. Tenemos en REXX tres instrucciones QUEUE, PUSH y PULL (ver Figura 6).

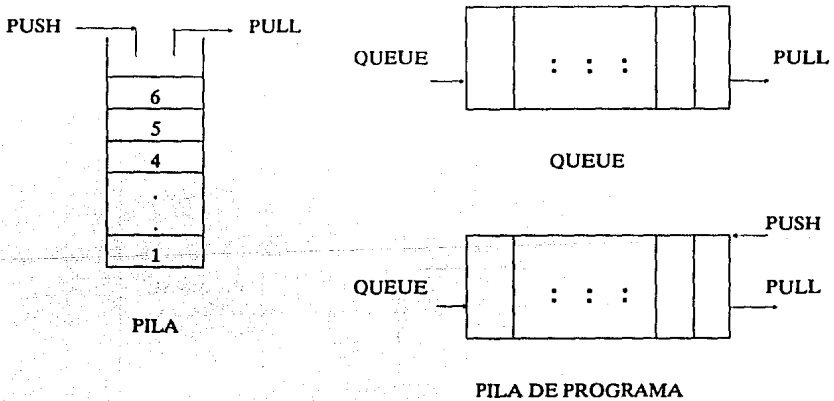


Figura 6.

PILA DE PROGRAMAS.

Un buffer es en general un término que se refiere a una parte de la memoria RAM, se usa para la entrada y salida de información.

Se pueden construir extensiones para la pila de programas, llamados buffers. Generalmente existe un sólo buffer en el programa.

- Se crea un nuevo buffer usando el comando MAKEBUF.
- QUEUE y PUSH, son comandos equivalentes para ingresar datos dentro de un buffer ya creado.
- PULL se utiliza para remover datos desde un buffer creado hasta que esté vacío.
- Cuando la pila de programas está completamente vacía, el dato se toma desde el buffer de la terminal de entrada.

Esto se llama una pila de buffers. La pila completa es denominada pila de consola (ver Figura 7).

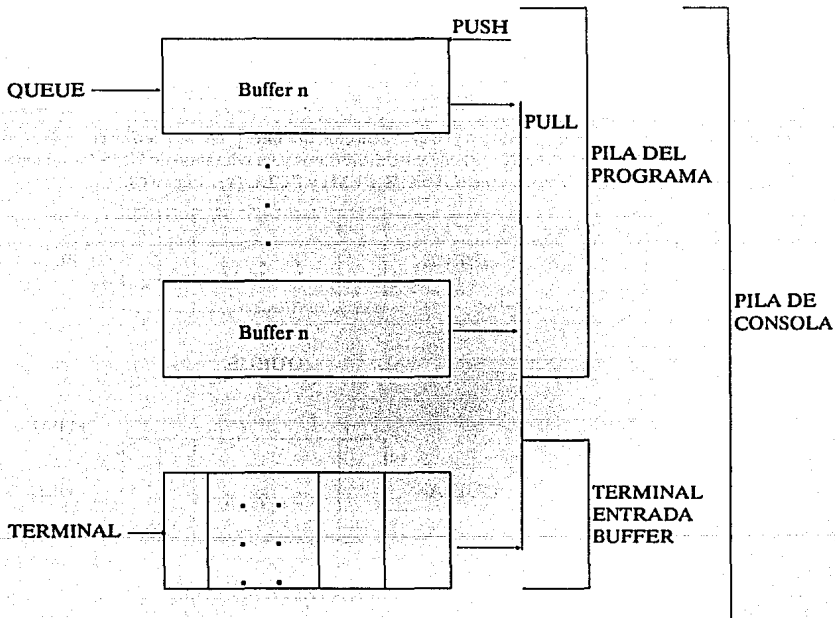


Figura 7.

2.6.2 USO DE LA PILA DE PROGRAMAS.

La forma de usar una pila de programas es la siguiente:

1. Se debe empezar una pila con MAKEBUF de CMS. Así, se tiene un buffer propio para la pila del programa.
2. La función QUEUED(), indica cuantas entradas tiene la pila.
3. La instrucción QUEUE, o un comando de CMS equivalente, permite introducir datos a la pila del programa.
4. Con la instrucción PULL, o un comando de CMS equivalente se toman datos del pila.
5. Es importante evitar el traslado de datos que no pertenezcan a la pila. Se puede checar esto utilizando la instrucción QUEUED(), preguntando que nuestras entradas no excedan al número de datos de nuestra pila del programa.
6. Con el comando de CMS DROPBUF, se trasladan todos los datos a la pila del programa antes de regresar al modo CMS.

Se pueden omitir los comandos MAKEBUF y DROPBUF sin causar errores, pero es aconsejable usarlos, para evitar problemas posteriores en caso de que se implemente como una subrutina, dentro de un programa que si los utilice.

2.7 MANEJO DE ARCHIVOS.

2.7.1 USO DE COMANDOS EXECIO.

Los comandos de CMS, EXECIO permiten manejar los movimientos de líneas entre los dispositivos virtuales y el programa pila. Leen y escriben archivos CMS. Emiten comandos de CP, colocando los resultados que deberfan aparecer en pantalla, dentro el programa pila. También permite leer de la lectora virtual y escribir en impresora y perforadora virtuales.

2.7.1.1 LECTURA DE ARCHIVOS.

Para leer una línea de un archivo desde un minidisco o un directorio, se usa:

EXECIO n DISKR fname ftype [fmode]

Donde:

EXECIO comando de CMS.

n es el número de líneas que se leen.

DISKR palabra clave de EXECIO que especifica "leer de un archivo" en un minidisco o directorio.

fname es el nombre del archivo de lectura.

ftype es el tipo del archivo de lectura.

fmode filemode que especifica el minidisco o directorio donde está el archivo de lectura.

Puede también leer una línea de un archivo y asignarla a una variable.

EXECIO 1 DISKR fname ftype [fmode] (VAR nvar

Donde:

nvar es el nombre de la variable.

No existe ninguna manera especial para leer el último registro, después de trabajar con los anteriores. Sin embargo, cuando el comando EXECIO llega al fin del archivo regresa un código de error igual a 2.

2.7.1.2 ESCRITURA DE ARCHIVOS.

Para escribir un número de líneas desde una pila en un archivo, se usan las siguientes instrucciones:

EXECIO n DISKW fname ftype [fmode] (STRING exp

Donde:

EXECIO comando de CMS.

n número de líneas a escribir.

DISKW palabra clave de EXECIO que especifica "escribir en un archivo" de un minidisco o directorio.

fname nombre del archivo en el que se escribe.

ftype tipo del archivo en el que se escribe.

fmode filemode que especifica el minidisco o directorio donde está el archivo de escritura.

STRING opción de EXECIO, que especifica que el resto del comando se escribirá como una línea en el archivo. Cuando se usa en un programa REXX, va seguido de una expresión.

exp expresión REXX. El resultado de ésta será escrito en un archivo.

CAPITULO 3

DESCRIPCION DEL PROCESO DE CONTROL DE UTILIZACION DE RECURSOS DEL SISTEMA

3.1 GENERALIDADES.

La contabilidad de los datos reunidos por el VM/CMS, puede ayudar en el análisis de todas las operaciones del sistema y permitir un control del uso de tiempo y otros recursos que utilice cada usuario.

Existe una máquina virtual asignada en particular para almacenar automáticamente toda la información acerca de la utilización de los recursos del sistema.

Esta máquina virtual, llamada DISKACNT, definida como máquina de servicio, tiene asignados dos minidiscos de trabajo; uno de ellos con capacidad de 32 cilindros, donde se encuentran los programas y archivos de trabajo que permiten generar los reportes de contabilidad; el segundo, de 20 cilindros, tiene entre otros archivos, el denominado "ACCOUNT CARDS", donde se almacenan todos los registros que contienen una descripción de quiénes o qué consumen recursos y una medida de estos recursos consumidos.

Antes de entrar en detalle al proceso de contabilidad, será necesario explicar algunos conceptos que se manejan posteriormente, y que se detallan a continuación.

3.2 PRIVILEGIOS Y PRIORIDADES.

Una forma de clasificar a las máquinas virtuales, es asignándoles privilegios y prioridades.

Los privilegios determinan la autorización que tiene cada usuario para utilizar todos los comandos que puedan alterar las funciones del sistema.

Las clases de privilegios son asignadas por el personal administrativo, según el tamaño y configuración del sistema y las circunstancias particulares de su instalación.

A cada versión de los comandos del CP se le asigna un código que corresponde al nivel del control del sistema que lo suministra (operaciones, recursos, programación análisis, spooling).

La siguiente tabla muestra las clases de privilegios que corresponden a las distintas máquinas virtuales.

Clases de privilegios	Usuario
A	Operador fundamental del sistema.
B	Operador de recursos del sistema.
C	Programador del sistema.
D	Operador de spool.
E	Analista del sistema.

F	Representante del servicio
G	Usuario de máquina virtual
ANY	Cualquier usuario

Las prioridades, determinan las máquinas que tienen precisamente prioridad para utilizar los recursos del sistema; se miden en una escala del 0 al 99. El sistema tiene prioridad 0, y los usuarios en general tienen 64.

3.3 DIRECTORIO DE USUARIOS.

El sistema cuenta con un directorio de usuarios que, contiene la información completa sobre la configuración de todas las máquinas virtuales del sistema, con el siguiente formato:

USER userid pass stor mstor pri clpri

En esta línea se define una máquina virtual y contiene:

- userid Clave del usuario; de uno a ocho caracteres alfanuméricos.
- pass Password, de uno a ocho caracteres que puede ser definido por el propietario de la máquina virtual, utilizando comandos de DIRMAINT.
- stor Define el tamaño de la memoria virtual para la máquina virtual; de uno a ocho dígitos decimales, en múltiplos de 4K, pudiendo especificarse en kbytes o megabytes.
- mstor Determina el tamaño máximo de memoria que puede definir el usuario; de uno a ocho dígitos decimales, en múltiplos de 4K, ya sea en kbytes o megabytes.
- pri Caracteres alfabéticos que determinan los privilegios de la máquina virtual.
- clpri Clase de prioridad.

ACCOUNT AAAAAAAAAA

Define el número de contabilidad.

OPTION XXXXXXXX

Permite seleccionar alguna de muchas opciones.

IPL sis PARM dat

Es una línea de control que contiene:

- sis Nombre del sistema, en este caso CMS, que será cargado cuando se inicie una sesión.

PARM dat Parámetros que se pasan al sistema que se carga

CONSOLE dirv tdisp

Esta línea de control especifica la consola virtual, donde:

dirv Es la dirección virtual del dispositivo; de uno a tres dígitos hexadecimales.
tdisp Es el tipo de dispositivo.

SPOOL dirv tdisp

Especifica la unidad de dispositivo que tendrá áreas spool, donde:

dirv Es la dirección virtual del dispositivo; de uno a tres dígitos hexadecimales, que serán:
 00C para lectora,
 00D para perforadora,
 00E para impresora.
tdisp Tipo de dispositivo.

LINK userid dirv1 dirv2 modo

Permite acceder la información contenida en un dispositivo de otro usuario, disponible sólo durante la sesión.

userid Clave del usuario propietario del dispositivo.
dirv1 Dirección virtual del dispositivo del usuario propietario.
dirv2 Dirección virtual del dispositivo para el usuario que hace la liga.
modo El modo de acceso formado por dos letras, que especifican:
 rr sólo de lectura,
 wr de lectura y escritura.
 mr acceso múltiple de lectura.

En general, se especificarán las ligas a los minidisks que contienen el sistema operativo, y se encuentran en los minidisks de la máquina virtual MAINT.

MKDISK dirv tdisp clys bloks

Describe las direcciones y el número de cilindros de todos los minidisks de trabajo de que dispone la máquina virtual.

dirv Dirección virtual del minidisk; de uno a tres dígitos hexadecimales.
tdisp Tipo de dispositivo.
clys Número de cilindros del minidisk; de uno a cuatro dígitos decimales.
bloks Número de bloques de arquitectura FBA; de uno a seis dígitos decimales.

DISKACNT, tiene actualmente la siguiente definición dentro del directorio de usuarios:

- USER DISKACNT XXXXXXXXX 512K 16M G 64
- ACCOUNT S0000031 S.OPER
- OPTION ECMODE
- IPL CMS PARM AUTOOCR
- CONSOLE 009 3215
- SPOOL 00C 2540 READER *
- SPOOL 00D 2540 PUNCH A

SYSACNT **USERID** = usuario,
OUTPUT = reader/punch,
CLASS = clase,
LIMIT = nnnnn

Donde:

Userid Es la identificación de la máquina virtual a la cual llegan los archivos de spool, generalmente DISKACNT.
Output Es el tipo de archivo spool que será creado.
Class Es la clase de archivo spool.
Limit Es el número de registros que serán acumulados antes de cerrar el archivo spool y enviarlo a la máquina virtual, encargada de la contabilidad. Este número tendrá un valor adecuado

Otra macro de DMKSYS, es SYSJRL, controla el monitoreo de los intentos inválidos que se hacen para iniciar una sesión o para tratar de ligarse a un minidisco protegido.

Tiene el siguiente formato:

SYSJRL **JOURNAL** = no/yes,
STQUERY = no/yes,
LOGUID = usuario,
LOGLOC = (n,m),
OGLMT = (x,y,z),
LNKUID = usuario,
LNKLMT = (x,y,z),
PSUPRS = no/yes

Donde:

Journal si se incluye esta función de monitoreo.
Stquery indica si se tiene o no capacidad para usar los comandos SET y QUERY JOURNAL como parte de la inicialización del sistema.
Loguid determina el usuario que recibirá la indicación de que se dio un LOGON inválido. En caso de no especificarlo, lo recibe el operador.
Logloc n indica el número máximo de veces para ingresar al sistema con un password inválido; y m el tiempo que transcurrirá, en minutos, hasta poder volver a ingresar nuevamente.
Loglmt x, y, z son los valores límites de intentos, para cuando se dé un LOGON/AUTOLOG con un password inválido, generando registros de contabilidad de tipo 04.
Lnkuid determina el usuario que recibirá la indicación de que se intentó un LINK, con password inválido. En caso de no especificarlo lo recibe el operador.
Lnlmt x, y, z son los valores límites para cuando se intente dar LINK con un password inválido, generando registros de contabilidad tipo 06.
Psuprs para formatos de LINK y LOGON se puede especificar o no el password en los comandos.

Actualmente, se tiene definida la macro como sigue:

SYSJRL	JOURNAL	= yes,
	STQUERY	= yes,
	LOGLOC	= (10,20),
	LOGLMT	= (2,10,10),
	LNKLMT	= (2,10,10),
	PSUPRS	= no

En otro capítulo se verán en detalle, los tipos de registros que genera SYSJRL.

3.5 INSTRUCCION DIAGNOSE.

La instrucción de DIAGNOSE, no puede ser utilizada en una máquina virtual como una función normal, porque sólo causa una interrupción en la sesión y regresa el control a CP. El formato contiene un código hexadecimal de dos bytes, valor que el CP utiliza para determinar la función a ejecutarse.

En particular la instrucción DIAGNOSE Code X'4C', genera los registros de contabilidad para usuarios virtuales de cualquier clase de privilegio, excepto ANY. Puede ser usado por cualquier máquina virtual, que incluya en su directorio la instrucción OPTION ACCT.

Dentro su formato el DIAGNOSE Code X'4C', contiene una dirección virtual, basada en subcódigos hexadecimales, que señalan algún tipo de información, y son:

- '0000' lista de parámetros sólo con la clave del usuario.
- '0004' lista de parámetros con la clave de usuario y el número de contabilidad.
- '0008' lista de parámetros con la clave de usuario y el número de distribución.
- '000C' lista de parámetros con la clave del usuario el número de contabilidad y el número de distribución.
- '0010' área de datos que contiene más de 70 bytes con información del usuario, que será transferida al archivo de contabilidad, empezando en la columna nueve, después de la clave.

Cuando un usuario quiere generar un registro de contabilidad, al introducir la instrucción DIAGNOSE Code X'4C' se verifica la bandera VMACCOUN en VMPSTAT, para comprobar que el usuario tiene la opción ACCT, si no, regresa el control al usuario que lo solicitó.

En caso contrario, pasa el control a la macro DMKCPV para generar el registro, y a su vez esta macro pasa el control a DMKACO para completar el cargado de información, ya sea desde el ACCTBLOK (Bloque de Contabilidad del Usuario) si tiene lista de parámetros, o del VMBLOK del usuario.

DMKCPV regresa el control a DIAGNOSE Code X'4C' para liberar el espacio en memoria para ACCTBLOK si existe la lista de parámetros. La instrucción de DIAGNOSE verifica luego esta lista de parámetros que le proporcionan los subcódigos '0000', '0004', '0008' y '000C'. Si la dirección de la lista de parámetros es igual a cero, el control regresa al usuario con un código de condición igual a cero.

Para el subcódigo '0010', si tanto la dirección virtual, como el tamaño son válidos, se llama a DMKFREE para obtener espacio para un buffer de contabilidad (ACNTBLOK), el cual se inicializa después con espacios en blanco.

La clave del usuario de la máquina virtual que utiliza la instrucción **DIAGNOSE**, queda en las columnas 1 a la 8, y el código de identificación del registro de contabilidad, se localiza en las columnas 79 y 80, y es igual a 'C0'.

La instrucción **DIAGNOSE Code X'40'**, durante una sesión virtual, limpia todo, restaurando la página del usuario y liberando la página real, siempre que no se trate de una máquina real, que actúa como virtual.

3.6 PROCESO PARA GENERAR REGISTROS DE CONTABILIDAD.

Los registros de contabilidad, en general, son de 80 bytes, área que es inicialmente borrada por el módulo **DMKACO**, con la instrucción **DIAGNOSE Code X'40'**, antes de ser llenada.

Cuando existe un número determinado de registros especificados en la macro de nombre **SYSACNT**, el archivo se cierra y queda disponible para ser procesado. También se puede producir el cierre de este archivo, mediante la ejecución del comando **ACNT CLOSE**.

Actualmente, la macro **SYSACNT** tiene la siguiente definición:

```
SYSACNT    USERID = DISKACNT,  
           OUTPUT  = READER,  
           CLASS   = C,  
           LIMIT   = 100
```

Por lo tanto, cada vez que se acumulan 100 registros, el archivo spool se cierra y se envía a la lectora de la máquina **DISKACNT**, para ser procesados periódicamente evitando que se acumulen.

DISKACNT trabaja desconectada con los registros de contabilidad. Un archivo ejecutable, llamado **"DTR\$ACNT FMTEXEC"**, se encarga de agregar la información del archivo de contabilidad de spool, a un archivo llamado **"ACCOUNT CARDS"**, que se encuentra en un minidisco de trabajo de esta máquina virtual, además de controlar los posibles errores que ocurran en este proceso, como es que el minidisco donde está el archivo **ACCOUNT CARDS** esté lleno y no pueda recibir más registros, caso en que dejará el archivo spool en la lectora.

Con este archivo de datos, se procederá a clasificar la información y generar los reportes.

CAPITULO 4

DISEÑO GENERAL DEL SISTEMA DE CONTROL DE UTILIZACION DE RECURSOS

4.1 INVESTIGACIONES PRELIMINARES.

4.1.1 ANTECEDENTES.

Los programas que actualmente generan la contabilidad del equipo I.B.M.-4381, producen dos tipos de reportes.

El primero describe por usuario la siguiente información: número de sesiones, tiempo de conexión al sistema, tiempo de uso del procesador real y virtual, páginas de memoria leídas, número de páginas de memoria escritas, operaciones de entrada y salida, líneas impresas, e incluyen el cálculo del costo monetario de los campos por dependencia.

Estos reportes van acompañados de cartas dirigidas a los encargados de todas las dependencias, informando sobre los costos comerciales de los recursos utilizados.

El segundo reporte contiene el total de cada uno de los campos antes mencionados, acumulados por mes, además del costo global para estas cantidades.

Ambos reportes se generan a partir de un sólo tipo de registros (Tipo 01), que contiene el archivo de entrada ACCOUNT CARDS, limitando la información que se puede obtener a partir de este archivo.

4.1.2 OBJETIVO DEL NUEVO SISTEMA.

Se propone un nuevo sistema que genere varios reportes, utilice todos los tipos de registros del archivo ACCOUNT CARDS, y brinde información más completa sobre la utilización de los recursos del equipo.

Además, es importante que estos reportes puedan generarse en cualquier tipo de período, según los requerimientos del usuario, pudiendo ser diarios, semanales, mensuales, etcétera, inclusive permitir el análisis en determinadas horas del día.

También, se pretende que sea un sistema amigable, de tal manera que el usuario no necesite conocer a fondo su funcionamiento y simplemente se limite a elegir las opciones y el período deseado.

4.1.3 ESTUDIO DE FACTIBILIDAD.

El nuevo sistema puede desarrollarse en las condiciones en que se encuentra el equipo actualmente. Se utilizará el Lenguaje Intérprete REXX en la codificación de los programas, aunque en caso de que los procesos sean muy complicados y necesiten mucho tiempo de ejecución, se requerirá del uso de un compilador para este lenguaje, software aún no disponible en la dependencia; para la generación de pantallas, se tiene el producto ISPF (Interactive System Product Facility).

Por otra parte, la emisión de varios reportes sobre la utilización de los recursos del sistema beneficiará a los usuarios, brindándoles información detallada que puede ser explotada en la op-

timización del sistema del equipo I.B.M.-4381, siendo mínimos los costos del desarrollo del nuevo sistema.

Por la facilidad de operación de este, no es necesaria una capacitación especial para el personal encargado de ejecutarlo.

4.1.4 LIMITANTES.

No se considera dentro del nuevo sistema el cálculo del valor comercial de cada recurso, dejando este detalle en consideración de los usuarios, al igual que la generación de cartas y oficios que acompañan los reportes dirigidos a otras dependencias.

En el caso de reportes que requieren mucho tiempo de procesamiento, los períodos se restringen a diarios o semanales, considerando que en un mes es excesiva la información contenida en el archivo ACCOUNT CARDS (promedio de 90 000 registros mensuales); sin embargo el espacio en disco es suficiente para almacenar esta información.

No se generan reportes sobre las ligas que los usuarios realizan con éxito a minidisquetes de otras máquinas virtuales (registros tipo 05), porque no se considera una información importante.

4.2 DISEÑO DE ENTRADA.

4.2.1 ARCHIVO DE ENTRADA "ACCOUNT CARDS".

Inicialmente, como se mencionó antes se cuenta con el archivo ACCOUNT CARDS generado automáticamente dentro de la máquina virtual DISKACNT. Este contiene registros de 80 bytes, que suministran información acerca del uso de los recursos del sistema y está disponible para ser procesado en cualquier momento.

En general, los registros de contabilidad del archivo ACCOUNT CARDS, se pueden clasificar en cuatro categorías:

- 1o. Registros de máquinas virtuales.
- 2o. Registros de dispositivos dedicados y espacio en discos temporales, asignados a usuarios de máquinas virtuales.
- 3o. Registros para procesos de LOGON, AUTOLOG y LINK.
- 4o. Registros generados por una instrucción DIAGNOSE X'4C'.

En el caso de proceso por lotes, que realiza la máquina CMSBATCH, se crea un registro de contabilidad que contiene la clave y número de contabilidad del usuario que envía un trabajo a esta máquina.

Existen varios tipos de registros para las cuatro categorías generales, antes mencionadas, en la contabilidad del sistema. A continuación se muestra una tabla con cada uno de estos registros, incluyendo su formato.

COMPARACION DE LOS TIPOS DE REGISTROS DE CONTABILIDAD

CP	DIAGNOSE X' 4C'	CONTENIDO
	Tipo C0	Generado por una máquina virtual de servicio, como la RSCSV2.
Tipo 01	Tipo C1	Uso de recursos de una máquina virtual.
Tipo 02	Tipo C2	Uso de dispositivos dedicados.
Tipo 03	Tipo C3	Uso de discos temporales.
Tipo 04		LOGON/AUTOLOG, password inválidos.
Tipo 05		Protección para liga de minidiscos.
Tipo 06		Password inválido para liga de minidiscos.
Tipo 07		Generado por la máquina virtual de servicio VTAM.
Tipo 08	Tipo C8	Identificación de la terminal.

4.2.1.1 REGISTROS DE CONTABILIDAD PARA USUARIOS DE UNA MAQUINA VIRTUAL

Cuando un usuario finaliza una sesión de terminal o se invoca el comando ACNT, se graba un registro de Tipo 01. Este registro contiene un resumen de todos los recursos que consumió el usuario durante la sesión, en detalle tiene la siguiente información:

TIPO 01

Columna Contenido

- 1-8 Clave del usuario (USERID).
- 9-16 Número de contabilidad.
- 17-28 Fecha y hora en que se produce el registro (mmddyyhhmmss).
- 29-32 Número de segundos conectados al sistema.
- 33-36 Milisegundos de proceso usado, incluyendo el tiempo de supervisión de VM/CMS.
- 37-40 Milisegundos de tiempo de uso de proceso virtual (sin incluir el tiempo de supervisión de VM/CMS).
- 41-44 Número de páginas leídas a la memoria virtual.
- 45-48 Número de páginas escritas desde la memoria virtual.
- 49-52 Número de operaciones de entrada y salida (sin incluir spooling).
- 53-56 Número de "tarjetas" virtuales perforadas.
- 57-60 Número de líneas para impresora virtual.
- 61-64 Número total de "tarjetas" virtuales leídas.
- 65-78 Reservado.
- 79-80 Código de identificación del registro de contabilidad (01).

4.2.1.2 REGISTROS DE CONTABILIDAD PARA DISPOSITIVOS DEDICADOS.

Se entiende por dispositivo dedicado, las unidades de cinta y dispositivos gráficos.

Un dispositivo dedicado, se asigna exclusivamente a una máquina virtual particular, mediante el comando ATTACH, asignados por la entrada del directorio o conectado mediante el comando DIAL, con un sistema que tiene líneas virtuales.

Estos registros son de Tipo 02, y se generan cuando el usuario los libera mediante el comando DETTACH, o si termina su sesión con LOGOFF o vía DIAL. Tienen el siguiente formato:

TIPO 02

Columna	Contenido
1-8	Clave del usuario (USERID).
9-16	Número de contabilidad.
17-28	Fecha y hora en que se produce el registro (mmddyymmss).
29-32	Número de segundos conectados al sistema.
33	Clase de dispositivo (1).
34	Tipo de dispositivo.
35	Modelo.
36	Características.
37-64	Sin usar.
65-72	Identificación de la terminal.
73-78	Sin usar.
79-80	Código de identificación del registro de contabilidad (02).

4.2.1.3 REGISTROS DE CONTABILIDAD PARA AREAS TEMPORALES.

Las áreas temporales son minidisks que cualquier usuario puede definir para su máquina virtual, especificando una dirección virtual, el número de cilindros y el modo de acceso. Son temporales, porque están disponibles solamente durante la sesión en que fueron creados o hasta que se inicia nuevamente el sistema en general, mediante el comando IPL.

Cuando un usuario libera un disco temporal, vía DETACH, LOGOFF o al terminar con DIAL, se genera un registro de contabilidad de Tipo 03, con el siguiente formato:

TIPO 03

Columna	Contenido
1-8	Clave del usuario (USERID).
9-16	Número de contabilidad.
17-28	Fecha y hora en que se produce el registro (mmddyymmss).
29-32	Número de segundos conectados al sistema.
33	Clase de dispositivo (1).
34	Tipo de dispositivo.
35	Modelo.
36	Características.
37-38	Número de cilindros usados en el disco temporal.
37-40	Número de bloques usados para dispositivos de arquitectura de bloques fijos (FBA)
39-78 & 41-78	Sin usar.
79-80	Código de identificación del registro de contabilidad (03).

NOTA:

(1) Para la clase de dispositivos, en la posición 33 identifica un dispositivo de bloque no fijos y fijos con:

- X'40', para dispositivos de arquitectura CKD.
- X'01', para dispositivos de arquitectura FBA.

4.2.1.4 REGISTROS DE CONTABILIDAD PARA LOGON INVALIDOS.

Cuando un usuario ha alcanzado o excedido el valor permitido de intentos para poder conectarse al sistema, es decir:

- Si el VM/CMS detecta que el usuario ha intentado darse LOGON o AUTOLOG, varias veces, con un password incorrecto.
- Si digita el USERID o PASSWORD incorrecto para una función APPC/VM CONNECT.

Entonces la macro SYSJRL, genera un registro de contabilidad de Tipo 04, con el siguiente formato:

TIPO 04

Columna	Contenido
1-8	Clave del usuario (USERID).
9-16	Reservado para uso de IBM.
17-28	Fecha y hora en que se produce el registro (mmddyyhhmss).
29-32	Dirección de la terminal (2).
33-40	Password inválido (3).
41-48	Userid usado en el comando AUTOLOG o userid para APPC/VM CONNECT..
49-51	Reservado para uso de IBM.
52-53	Número real de intentos del password inválido.
54-55	Límite máximo de intentos pre-establecido (JPSLOGAR).
56-70	Reservado para uso de IBM.
71-78	LUNAME para terminales SNA.
79-80	Código de identificación del registro de contabilidad (04).

4.2.1.5 REGISTROS DE CONTABILIDAD PARA LIGAS A MINIDISCOS.

Un usuario puede acceder minidiscos de otras máquinas virtuales, mediante el comando LINK, si conoce los password de liga para estos discos.

En general, al iniciar una sesión, todas las máquinas virtuales se ligan a minidiscos del sistema, para tener una copia del sistema operativo.

Cuando VM/CMS, mediante la macro SYSJRL, detecta que un usuario realizó un LINK a un minidisco protegido de otro usuario, se genera un registro de contabilidad Tipo 05, que se detalla a continuación:

TIPO 05

Columna	Contenido
1-8	Clave del usuario (USERID).
9-16	Número de contabilidad.
17-28	Fecha y hora en que se produce el registro (mmdyyhhmss).
29-32	Dirección de la terminal (2).
33-40	Reservado para uso de IBM.
41-48	Clave del usuario propietario del minidisco al que se hizo LINK.
49-51	Dirección del minidisco para el cual su usó el comando LINK.
52-70	Reservado para uso de IBM.
71-78	LUNAME para terminales de SNA.
56-70	Reservado para uso de IBM.
79-80	Código de identificación del registro de contabilidad (05).

Los registros Tipo 06, se escriben cuando VM/CMS detecta que un usuario ha intentado varias veces ligarse a un minidisco con un password inválido, sin lograrlo.

TIPO 06

Columna	Contenido
1-8	Clave del usuario (USERID)
9-16	Número de contabilidad.
17-28	Fecha y hora en que se produce el registro (mmdyyhhmss).
29-32	Dirección de la terminal (2).
33-40	Password inválido (3).
41-48	Clave del usuario propietario del minidisco al que se intentó ligarse.
49-51	Dirección del minidisco para el cual se usó el comando LINK.
2-53	Número real de intentos del password inválido.
54-55	Límite máximo de intentos pre-establecido (JPSLOGAR).
56-70	Reservado para uso de IBM.
71-78	LUNAME para terminales SNA.
79-80	Código de identificación del registro de contabilidad (06).

NOTAS:

- (2) Para la dirección de la terminal, en las posiciones 29-32 tenemos:
- NONE, si no encuentra la terminal.
 - Identificación del recurso, para terminales bisncronas remotas (BSC).
 - Dirección del dispositivo real en otros casos.
- (3) Para el password inválido, las posiciones 33-40 contienen:
- Password incorrecto.
 - 'TERM/ERR', si se borra la línea durante la entrada del password.
 - 'TOO LONG', si se teclea un password de más de ocho caracteres.

4.2.1.6 REGISTROS DE CONTABILIDAD PARA LA MAQUINA VIRTUAL DE SERVICIO VTAM

Los registros Tipo 07, se generan cuando un usuario apaga un dispositivo controlado por la máquina de servicio VTAM. El registro indica los recursos que utilizó el usuario.

TIPO 07

Columna	Contenido
1-8	Clave del usuario (USERID).
9-16	Número de contabilidad ó 0000.
17-20	Overflow words.
21-24	Número de RUS de entrada.
25-28	Número total de bytes de entrada.
29-32	Número de RUS de salida.
33-36	Número total de bytes de salida.
37-40	Número total de líneas en la consola del usuario.
41-44	Número de copias a impresión del contenido de la terminal realizadas por el usuario.
45-52	Hora en la que se da LOGON a una sesión.
53-60	Hora en la que se da LOGOFF a una sesión.
61-68	Nombre lógico de la terminal de trabajo.
69-78	Sin usar.
79-80	Código de identificación del registro de contabilidad (07).

4.2.1.7 REGISTROS DE CONTABILIDAD CON IDENTIFICACION DE TERMINAL.

El registro Tipo 08, se escribe cuando el usuario da DISCONNECT, LOGOFF, o sale con los comandos SHUTDOWN y/o FORCE, además dentro del registro se especifica la identificación de la terminal de trabajo. No se genera un registro Tipo 08 para un usuario desconectado, que es sacado del sistema vía FORCE.

TIPO 08

Columna	Contenido
1-8	Clave del usuario (USERID).
9-16	Número de contabilidad.
17-28	Fecha y hora en que se produce el registro (mmddyymmss).
29-64	Reservado para uso de IBM.
65-72	Identificación de la terminal (4).
73-78	Reservado para uso de IBM.
79-80	Código de identificación del registro de contabilidad (08).

NOTA:

(4) Para la identificación de la terminal, las posiciones 65-72 contienen:

- Identificación de los recursos (4 bytes) para terminales bisncronas remotas.
- EL LUNAME (8 bytes) si el usuario está firmado a una terminal VM/VTAM (para una terminal de SNA).

- Dirección del dispositivo real (4 bytes). Si es un dispositivo lógico, entonces la dirección empieza con 'L' precedida de 3 caracteres. Si no es ninguno, entonces tiene blancos.

4.2.1.8 REGISTROS DE CONTABILIDAD PARA LA MAQUINA VIRTUAL DE SERVICIO RSCSV2.

El RSCSV2 genera el siguiente registro de contabilidad (Tipo C0) por cada archivo que recibe y/o transmite. El registro contiene información básica de los recursos que se utilizaron durante la transmisión del archivo.

TIPO C0

Columna	Contenido
1-8	Userid del RSCSV2.
9-16	Userid de la máquina virtual de origen.
17-28	Fecha y hora en que se produce el registro (mmddyymmss).
29-30	Número de archivo en el área spool que asigna RSCSV2.
31-32	Número de archivo en el área spool que asigna el sistema.
33-40	Nodo de origen.
41-48	Nodo de destino.
49	Clase del archivo que asigna el sistema.
50	Tipo de dispositivo de origen ('8N' = PUN/'4N' = PRT).
51-52	Sin usar.
53-56	Número de registros del archivo.
57-64	Userid del destinatario.
65-72	Sin usar.
73-77	Identificación del sistema (serie y modelo).
78	Código de transmisión ('01' = SEND/'02' = RECV).
79-80	Código de identificación del registro de contabilidad (C0).

4.2.1.9 REGISTROS DE CONTABILIDAD GENERADOS POR SQL/DS.

Los registros de contabilidad generados por SQL/DS contienen una descripción de los recursos consumidos por los usuarios y los procesos de SQL/DS.

Los registros se escriben cada vez que termina una sesión de SQL/DS, también si un usuario se reconecta sin liberarse explícitamente de una conexión anterior. Es decir, un usuario al usar ISQL se conecta implícitamente a SQL/DS, luego se conecta explícitamente a SQLDBA para realizar algunas tareas que requieren de la autoridad de DBA, entonces cuando se produce este cambio de usuarios se escribe un registro de contabilidad para el usuario y se abre una nueva sesión para el SQLDBA.

A veces, también se escribe un registro durante una sesión de SQL/DS. Internamente, SQL/DS contiene un límite en el uso de recursos. Cuando un usuario excede este límite, SQL/DS escribe un registro de contabilidad con el nombre de identificación del usuario, aún cuando este no termine su sesión de trabajo.

Por lo tanto, tenemos los siguientes tipos de registros de contabilidad generados por SQL/DS:

Inicialización: SQL/DS escribe un registro de contabilidad cuando termina la inicialización de SQL/DS. Este registro describe el consumo de recursos durante el proceso de inicialización.

Registro de Inicialización.

Columna	Contenido
1-8	Clave de usuario de la máquina de base de datos SQL/DS
9-16	"SQL/DS"
17-24	"INIT"
25-40	Reservado.
41-52	Fecha y hora en que se produce el registro (mmdyyhhmss).
53-60	Sin usar.
61-64	Duración del proceso de inicialización de SQL/DS, en segundos.
65-68	Tiempo del procesador usado para la inicialización de SQL/DS, en milisegundos.
69-72	Número de veces en que SQL/DS revisa dentro de la página del buffer durante la inicialización (equivalente a usar COUNTER LPAGBUFF inmediatamente después de la inicialización).
73-74	Reservado.
75-78	"xSQL" identificador que separa los registros de SQL/DS de otros registros de contabilidad de VM/SP (x = I para Inicialización).
79-80	Código de identificación del registro de contabilidad para CP (C0).

Checkpoints: Después de un checkpoint, SQL/DS escribe un registro de contabilidad, que refleja los recursos consumidos al construir el archivo y sus respectivos checkpoints.

Operación: Durante la terminación, en caso de estar en modo de usuario múltiple, se genera un registro de contabilidad para procesar lo que el agente operador hizo desde que SQL/DS empezó a funcionar.

Registro de Checkpoint y Operación.

Columna	Contenido
1-8	Clave de usuario de la máquina de base de datos SQL/DS.
9-16	"SQL/DS".
17-24	"SYSTEM".
25-40	Reservado.
41-52	Fecha y hora en que se produce el registro (mmdyyhhmss).
53-60	Sin usar.
61-64	Cero binario.
65-68	Tiempo de procesador usado por SQL/DS, en milisegundos.
69-72	Número de veces en que este agente revisa dentro de la página del buffer (equivalente a usar COUNTER LPAGBUFF sólo por el agente)
73-74	Reservado.
75-78	"xSQL" identificador que separa los registros de SQL/DS de otros registros de contabilidad de VM/SP (x = C para Checkpoint o O para Operador).
79-80	Código de identificación del registro de contabilidad para CP (C0).

Terminación: SQL/DS escribe un registro de contabilidad recopilando los recursos que se han consumido durante la ejecución completa de SQL/DS.

Registro de Terminación.

Columna	Contenido
1-8	Clave de usuario de la máquina de base de datos SQL/DS
9-16	"SQL/DS".
17-24	"TERM".
25-40	Reservado.
41-52	Fecha y hora en que se produce el registro (mmddyhhmss).
53-60	Sin usar.
61-64	Tiempo, en segundos, desde SQL/DS empieza a funcionar hasta que se da shutdown SQL/DS binario.

Lo que sigue son los totales de la ejecución completa de SQL/DS. Los totales se extraen desde los datos que se usan con el comando COUNTER.

65-68	DASDIO. Número total de DASD I/Os.
69-72	LPAGBUFF. Número de veces que SQL/DS revisa la página del buffer.
73-74	Reservado.
75-78	"xSQL" identificador que separa los registros de SQL/DS de otros registros de contabilidad de VM/SP (x = T para Terminación).
79-80	Código de identificación del registro de contabilidad para CP (C0).

Registro de Usuario.

Columna	Contenido
1-8	Clave de usuario de la máquina de base de datos SQL/DS.
9-16	Clave de usuario que accesa SQL/DS.
17-24	Clave de usuario de SQL/DS que se estableció usando el proceso de connect. A menos, que el usuario cambie la clave de usuario conectada, el userid de SQL/DS es el mismo que el userid de VM/SP.
25-40	Datos suministrados por la instalación (además de los números de contabilidad). Si no tiene un programa codificado que suministre la información a este campo, SQL/DS llena este campo en blanco.
41-52	Fecha y hora en que se produce el registro (mmddyhhmss).
53-60	Nombre del modulo de acceso se activó al final para la aplicación (también se refiere al nombre del programa o prenombre).

Lo que sigue son los totales para el agente, que reflejan los valores acumulados por el usuario.

61-64	Tiempo activo (tiempo en que el usuario estuvo conectado a un agente), en segundos.
65-68	Tiempo de uso del procesador, en milisegundos.
69-72	Número de veces que el agente revisa la página del buffer (este valor es equivalente al valor del contador LPAGBUFF, para un usuario individual).
73-74	Reservado.
75-78	"xSQL" identificador que separa los registros de SQL/DS de otros registros de contabilidad de VM/SP (x = U para Usuario).
79-80	Código de identificación del registro de contabilidad para CP (C0).

4.2.1.10 CREACION DE REGISTROS DE CONTABILIDAD.

Un usuario de una máquina virtual, puede iniciar la creación de un registro de contabilidad, que contenga más de 70 bytes de información, con sus propias modificaciones. Para esto el usuario da instrucciones al código DIAGNOSE X'4C', que se describió en el anterior capítulo, cuyos operadores son:

- La dirección del área de datos en memoria virtual que contiene la información que el usuario desee tener en las columnas 9 a la 78, de su registro.
- Una función hexadecimal de código X'10'.
- El tamaño del área de datos en bytes.

La información que contenga el registro de contabilidad será:

TIPO C0

Columna Contenido

1-8 Clave del usuario.

9-78 Formato del usuario.

79-80 Código de identificación del registro de contabilidad (C0).

Este es el caso de la máquina virtual RSCSV2, que se describió anteriormente. También tienen este formato los registros Tipo C1, C2, C3, C4 y C8.

Todos los registros contienen algunos campos en código EBCIDIC, para cuyo análisis es necesario realizar conversiones a valores decimales; estas conversiones se realizan dentro de los programas.

En general, todos los datos que se requieren para los programas los proporciona el archivo ACCOUNT CARDS. Además, mediante una pantalla se piden las fechas que se considerarán para determinar el período en el que se procesará la información.

4.2.2 ARCHIVO "DIRUSER DATA".

Solamente para reportes que se generan con registros que no incluyen el número de contabilidad, se necesita acceder el directorio general del sistema y mediante un proceso se creará un archivo que contenga todas las claves de usuario y sus respectivos números de contabilidad. Este archivo "DIRUSER DATA", es necesario actualizarlo cada vez que se produzcan altas y bajas de claves de usuario.

4.2.3 ARCHIVO "DEPUNAM DATA".

Para obtener el nombre de la dependencia, desplegado en los encabezados de todos los reportes, se cuenta con un archivo de datos que recopila estos nombre relacionados con tres dígitos numéricos que corresponde a los tres primeros dígitos del número de contabilidad de cada usuario. Este archivo "DEPUNAM DATA", deberá modificarse toda vez que se necesite dar de alta o eliminar una dependencia o en el caso de cambiar el nombre de alguna.

4.2.4 OTROS ARCHIVOS.

Durante la ejecución de los programas se crearán además otros archivos a partir del ACCOUNT CARDS.

Primero el archivo que contenga sólo los registros comprendidos en el período elegido y además que correspondan a algún código de identificación según el reporte que se quiera generar.

De este nuevo archivo, se seleccionan los campos que se requieren trabajar, convirtiéndolos de código EBCDIC a decimal, con los registros ordenados primeramente por clave de usuario y después por dependencia. Este último archivo permite generar el reporte correspondiente.

4.2.5 DISEÑO DE PANTALLAS.

Para interactuar con el usuario, se crearán pantallas que cumplan las siguientes funciones:

1. Presentar un menú de opciones, que permita elegir un tipo de reporte (ver Figura 8).
2. Pedir información a cerca del período de tiempo en el que se produjeron los registros en el archivo de entrada (ver Figura 9).
3. Desplieguen algunos mensajes de advertencia, en caso de no poder seguir el proceso (ver Figura 10).
4. Editar y mandar a impresora virtual, el archivo que contiene el reporte deseado (ver Figura 11).

4.3 DISEÑO DE SALIDA.

En base a la información contenida en el archivo ACCOUNT CARDS, se pueden generar diversos reportes. Uno de ellos incluirá información completa de todos los recursos utilizados por cada usuario y existirán otros reportes más específicos, detallando algunos campos que se indican en el reporte general.

A continuación se detallan cada uno de estos reportes.

4.3.1 REPORTE GENERAL.

Este reporte, estará clasificado por dependencia obteniendo información completa para cada usuario. Además, se incluye un reporte con los totales por dependencia.

Los campos que se describen son:

- Período.
- Nombre de la dependencia.
- Clave de usuario.
- Número de contabilidad.
- Número de sesiones durante el período especificado.
- Tiempo de conexión al sistema.
- Tiempo de utilización de CPU, con dos valores:
 - incluyendo el tiempo de supervisión de VM/CMS.
 - sin incluir el tiempo de supervisión de VM/CMS.
- Número de líneas escritas en impresora virtual.
- Número de registros en área spool: entrada y salida.
- Número de operaciones de entrada y salida.
- Número de dispositivos dedicados que se usaron.
- Tiempo de uso de dispositivos dedicados.
- Número de discos temporales utilizados.
- Tiempo de uso de discos temporales.

- Total de cilindros usados en áreas temporales.
- Número de archivos enviados por la red.
- Número de registros recibidos por la red.
- Número de bytes transmitidos mediante VTAM, especificando salidas y entradas.
- Uso de proceso en lotes con CMSBATCH, con los siguientes parámetros:
 - Tiempo de uso de CPU.
 - Número de discos temporales.
- Utilización del manejador de base de datos SQL.

4.3.2 REPORTE SOBRE EL USO DE DISCOS TEMPORALES.

Este reporte emite información más completa sobre el uso de áreas temporales, clasificada por dependencias, con los siguientes campos:

- Período.
- Dependencia.
- Clave de usuario.
- Número de contabilidad.
- Tiempo de uso de CPU.
- Clase de dispositivo.
- Tipo de dispositivo.
- Número total de cilindros usados.
- Número de bloques ocupados.

La clase y tipo de dispositivo, se especifica en un apéndice.

4.3.3 REPORTE SOBRE LOGON INVALIDOS.

Este reporte se genera en base a la información de los registros Tipo 04, cuando se intenta entrar en sesión, digitando password inválidos. Tiene el siguiente formato:

- Período.
- Dependencia.
- Clave de usuario.
- Fecha y hora.
- Dirección de la terminal.
- Password inválido.
- Número de intentos en ingresar al sistema
- Límite máximo de intentos permitidos.
- Para el caso de ser una terminal de SNA, se especifica el nombre lógico.

4.3.4 REPORTE DE LINK INVALIDOS.

Tomando en cuenta la información de los registros de Tipo 06, se emite un reporte por dependencia, para el caso de intentar ligarse a un disco protegido de otro usuario. Se detallan los siguientes campos:

- Período.
- Dependencia.
- Clave de usuario
- Fecha y hora.
- Dirección de la terminal.
- Password inválido.
- Clave del usuario propietario del minidisco.
- Dirección virtual del disco del usuario propietario.
- Número de intentos de liga.
- LUNAME para el caso de terminales de SNA.

4.3.5 REPORTE SOBRE LA UTILIZACION DE TERMINALES.

En este reporte se especifican todas las terminales, tanto locales como remotas, con información sobre el tiempo que las utilizaron, el número de usuarios y el número de sesiones por cada terminal, basándonos en los registros Tipo 01 y 08.

En el reporte figuran los siguientes campos:

- Período.
- Identificación de la terminal.
- Número de sesiones por terminal.
- Número de usuarios distintos por terminal.
- Tiempo de utilización.

4.3.6 REPORTE SOBRE UTILIZACION DE VTAM.

Todos los usuarios que trabajan con VTAM, generan registros de Tipo 07, de donde se extrae la siguiente información para el reporte:

- Período.
- Dependencia.
- Clave de usuario.
- Número de contabilidad.
- Tiempo de duración de las sesiones.
- Número total de bytes de entrada.
- Número total de bytes de salida.
- Número total de líneas desplegadas en la terminal del usuario.
- Nombre lógico de la terminal de trabajo.

4.3.7 REPORTE SOBRE LA UTILIZACION DE RSCSV2.

Todos los procesos que requieren la intervención de RSCSV2, generan mediante el uso de la instrucción de DIAGNOSE, un registro de contabilidad Tipo C0, de donde se obtiene la siguiente información:

- Período.
- Dependencia.
- Clave del usuario origen.
- Nodo de origen.
- Clave del usuario destinatario.
- Nodo destino.
- Tipo de dispositivo origen.
- Clase que el sistema le asigna al archivo enviado o recibido.
- Número de registros del archivo.
- Tipo de transmisión: enviar o recibir.

Con toda la información descrita, podemos detallar el desarrollo del sistema, expuesto en el siguiente capítulo.

SISTEMA DE CONTROL DE RECURSOS

EQUIPO I.B.M. - 4381

D.G.S.C.A.

USUARIO
HORADISKACNT
18:45

SELECCIONE LA OPCION DESEADA ==> _

- 1 REPORTE GENERAL.
- 2 REPORTE DE UTILIZACION DE DISCOS TEMPORALES.
- 3 REPORTE DE LOGON INVALIDOS.
- 4 REPORTE DE LINK INVALIDOS.
- 5 REPORTE DE UTILIZACION DE TERMINALES.
- 6 REPORTE DE UTILIZACION DE UTAM.
- 7 REPORTE DE UTILIZACION DE RSCSUZ.

PF1 = AYUDA

PF3 = SALIR

FIGURA 8. PANTALLA CON MENU DE OPCIONES

SISTEMA DE CONTROL DE RECURSOS

EQUIPO I.B.M. - 4381

D.G.S.C.A.

USUARIO DISKACNT
HORA 10:45

**A CONTINUACION INTRODUZCA EL PERIODO EN EL QUE DESEA SE
PROCESE EL REPORTE.**

**EL FORMATO PARA FECHA ES dd/mm/aa. SI DESEA, TAMBIEN PUEDE
ESPECIFICAR LA HORA, CON EL SIGUIENTE FORMATO: hh:mm:ss.**

INICIA EN: FECHA ==> / / HORA ==> : :

TERMINA EN: FECHA ==> / / HORA ==> : :

PF1 = AYUDA PF3 = SALIR PF8 = CONTINUAR

FIGURA 2. PANTALLA PARA PEDIR EL PERIODO DE TIEMPO.

SISTEMA DE CONTROL DE RECURSOS
EQUIPO I.B.M. - 4381
D.G.S.C.A.

USUARIO	DISKACNT
HORA	11:00

COMANDO ==> _

DENTRO DEL PERIODO ELEGIDO, NO SE ENCONTRARON
REGISTROS QUE PERMITAN GENERAR EL REPORTE
DESEADO.

PUEDE VOLVER A PEDIR EL MISMO REPORTE, CAMBIANDO
EL PERIODO.

PF3 = SALIR PF8 = CONTINUAR

FIGURA 10. PANTALLA DE ADVERTENCIA CUANDO NO ENCUENTRA INFORMACION.

SISTEMA DE CONTROL DE RECURSOS

EQUIPO I.B.M. - 4381

D.G.S.C.A.

**USUARIO
HORA**

**DISKACNT
11:36**

**EL PROCESO TERMINO EXITOSAMENTE. A CONTINUACION
PUEDE REALIZAR LAS SIGUIENTES OPERACIONES.**

- **VER EL REPORTE EN DISCO.
TECLEE "CMS MIRAR"
EN LA LINEA DE COMANDOS.**
- **MANDAR EL REPORTE A IMPRESORA VIRTUAL.
TECLEE "CMS IMPRIME"
EN LA LINEA DE COMANDOS**

PF3 = SALIR

PF8 = CONTINUAR

FIGURA 11. PANTALLA QUE PERMITE EDITAR E IMPRIMIR EL REPORTE.

DIRECCION GENERAL DE SERVICIOS DE COMPUTO ACADEMICO**EQUIPO I.B.M. - 4381****REPORTE DEL USO DE DISCOS TEMPORALES****PERIODO :****DEPENDENCIA :**

CLAVE DEL USUARIO	NRO. DE CONTABI- LIDAD	TIEMPO DE USO HH:MM:SS	TIPO DE DISPO- TIPO	NRO. DE CYLS. USADOS
----------------------	------------------------------	------------------------------	---------------------------	----------------------------

**FIGURA 13. FORMATO DE REPORTE DE UTILIZACION
DE DISCOS TEMPORALES.**

DIRECCION GENERAL DE SERVICIOS DE COMPUTO ACADÉMICO

EQUIPO I.B.M. - 4381

REPORTE DE LOGON INVALIDOS

PERIODO :

DEPENDENCIA :

CLAVE DEL ORDEN	FECHA	HORA	DIRECCION CENTRAL	TAMPO INVALIDO	NÚM. DE TÍTULOS	ADJAS DE TITULOS	NUMERO DE ERRORES
--------------------	-------	------	----------------------	-------------------	--------------------	---------------------	----------------------

FIGURA 14. FORMATO DEL REPORTE DE LOGON INVALIDOS.

DIRECCION GENERAL DE SERVICIOS DE COMPUTO ACADEMICO

EQUIPO I.B.M. - 4381

REPORTE DE LINK INVALIDOS

PERIODO :

DEPENDENCIA :

CLAVE DEL USUARIO	FECHA	HORA	DIRECCION CENTRAL	PASWORD INVALIDO	CLAVE DEL USUARIO DEL DISCO	DIRECCION DEL DISCO	NO. DE PASWORDS	LUNAS DE ENA ERR.
----------------------	-------	------	----------------------	---------------------	-----------------------------------	------------------------	--------------------	-------------------------

FIGURA 15. FORMATO DEL REPORTE DE LINK INVALIDOS.

DIRECCION GENERAL DE SERVICIOS DE COMPUTO ACADEMICO

EQUIPO I.B.M. - 4381

REPORTE DE LA UTILIZACION DE TERMINALES

PERIODO :

IDENTIFICACION DE LA TERMINAL	NUMERO DE SESIONES	HRO. DE USUARIOS DISTINTOS	TIEMPO DE USO HH:MM:SS
-------------------------------------	--------------------------	----------------------------------	------------------------------

FIGURA 16. FORMATO DEL REPORTE SOBRE UTILIZACION DE TERMINALES.

DIRECCION GENERAL DE SERVICIOS DE COMPUTO ACADÉMICO

EQUIPO I.B.M. - 4361

REPORTE DE LINEAS TRANSMITIDAS

PERIODO :

DEPENDENCIA :

CODIGO DEL CIRCUITO	NUMERO DE CIRCUITOS	TIEMPO DE USO	NO. TOTAL DE EXTRAS	NO. TOTAL DE EXTRAS	NO. TOTAL DE EXTRAS	NO. TOTAL DE EXTRAS
------------------------	------------------------	------------------	------------------------	------------------------	------------------------	------------------------

FIGURA 17. FORMATO DEL REPORTE DE LINEAS TRANSMITIDAS POR UTM.

DIRECCION GENERAL DE SERVICIOS DE COMPUTO ACADÉMICO

EQUIPO I.B.M. - 4381

REPORTE DEL USO DE RECURSOS

PERIODO

USUARIO	DECIMAS/ENVIOS CEROS	FECHA MM-DD-AA	HORA MM:NN:SS	NRO. DE PROCESOS	NRO. DE REQUISITOS
---------	-------------------------	-------------------	------------------	---------------------	-----------------------

FIGURA 18. FORMATO DEL REPORTE SOBRE UTILIZACION DE RECURSOS.

CAPITULO 5

DISEÑO DE SOFTWARE Y CONSTRUCCION DEL SISTEMA DE CONTROL DE UTILIZACION DE RECURSOS

Una vez establecidos los detalles del diseño del nuevo sistema, se procederá a describir la lógica de los procesos que lo forman.

Existirán distintos procedimientos para cada tipo de reporte, pero en general los que requiere son:

- Leer el archivo de entrada ACCOUNT CARDS.
- Seleccionar los registros comprendidos en el período de tiempo establecido por el usuario.
- Seleccionar los registros con determinado código de identificación de acuerdo al reporte que se quiera generar.
- Elegir de cada registro sólo los campos que se necesitan según el detalle del reporte y convertir los datos que estén en código EBCDIC a decimal.
- Ordenar la información por el campo clave de usuario y además por dependencia.
- Generar los encabezados del reporte, con el período y nombre de la dependencia correspondiente.
- Escribir en un archivo los encabezados y los respectivos datos procesados.

Para ilustrar más estas funciones, se utilizarán diagramas estructurados y modulares, exponiendo los más relevantes del sistema, que permitirán codificar el programa, simplificando su comprensión y posibles modificaciones.

La simbología utilizada (ver Figura 19) es sencilla y facilita el seguimiento de la lógica que tiene el sistema en general.

El módulo principal, es CONTIBM (ver Figura 20), despliega la pantalla de presentación del sistema, con un menú de opciones para los distintos reportes; después de realizar la elección se ingresa en uno de los programas principales correspondiente a un determinado reporte, en este caso, el Reporte General (módulo GENERAL, Figura 21), entonces se pide el período (fechas y horas) en el cual se desean generar el respectivo reporte. Estas fechas y horas ingresan en un grupo de variables globales que se define previamente, convirtiendo los datos numéricos a los nombres de los meses respectivos que se desplegarán en los reportes; todo este proceso se realiza en el programa PERIODO (ver Figura 22).

Con esta información procede a seleccionar del archivo ACCOUNT CARDS, los registros que cumplan con los dos anteriores requisitos. Este proceso se ilustra con el programa SELECOD1 (ver Figura 23), para nuestro ejemplo. Cada tipo de reporte cuenta con un programa similar que realiza esta función, creando un nuevo archivo de datos. Como el Reporte General contempla varios conceptos, genera más de un archivo de datos.

Estos archivos se clasifican por el campo clave de usuario (programa SORTUSER, Figura 24). Después se accesan, procesando cada registro, escogiendo sólo los campos de interés, convirtiendo estos datos de código EBCDIC a hexadecimal y luego a decimal, para que sean entendibles, y escribiendo esta nueva información en un nuevo archivo.

Para cada recurso, se tendrán procesos distintos, que se tratan en forma general en el reporte que ejemplificamos y de manera particular en los restantes reportes. Los programas que los ejecutan son:

DISKTEMP (ver Figura 26), toma el archivo de los registros generados por utilizar áreas temporales, con subrutinas internas de lectura, conversiones, cálculo de totales y creación de un nuevo archivo (ver Figura 27). Estos forman parte del programa principal para el Reporte de Utilización de Discos Temporales (ver Figura 33).

DISPDED (ver Figura 25), procesa de igual manera, los registros que se refieren a la utilización de dispositivos dedicados.

PVTAM (ver Figura 28), procesa los registros que se generan al utilizar VTAM, debiendo tomar en cuenta que al agrupar la información por usuarios, se detecten aquellos que no finalizaron una sesión normalmente, sino sólo se mantuvieron desconectados del sistema. También es un módulo del programa principal del Reporte de Utilización de VTAM (ver Figura 38).

PRSCS01 (ver Figura 29), para este módulo, se clasifica el archivo por el campo clave de usuario que envía información a través de RSCS. Existirá un programa similar que trabaja con los registros de los usuarios que recibieron archivos también vía RSCS. Estos programas generan la información para el Reporte sobre Utilización de RSCSV2 (ver Figura 37).

Posteriormente se clasifican los últimos archivos de datos, por dependencia (módulo **SORTDEP**, Figura 31); se escriben los encabezados para el reporte deseado, así como los registros anteriormente procesados para cada usuario, por dependencia, en módulos como **IMPRE01** (ver Figura 32).

En particular el programa principal **USOTERM** (ver Figura 34), genera el reporte sobre uso de terminales, teniendo que recurrir a dos tipos de registros 01 y 08, que se producen al terminar una sesión de trabajo; se clasifica el archivo por el campo identificación de terminal, y en un segundo proceso se contabilizan el número total de usuarios y el tiempo total de conexión al procesador, produciendo finalmente un archivo con el Reporte de Uso de Terminales.

Para los casos de los Reportes de Logon y Link Inválidos, se seleccionan los registros de tipo 04 y 06 respectivamente, se realizan las conversiones a los campos que se requieren y se crean archivos con los reportes para cada dependencia.

Si se desea conocer más detalladamente cada uno de los procesos mencionados, se puede acudir a **DGSCA**, donde se encuentra información más amplia y específica.

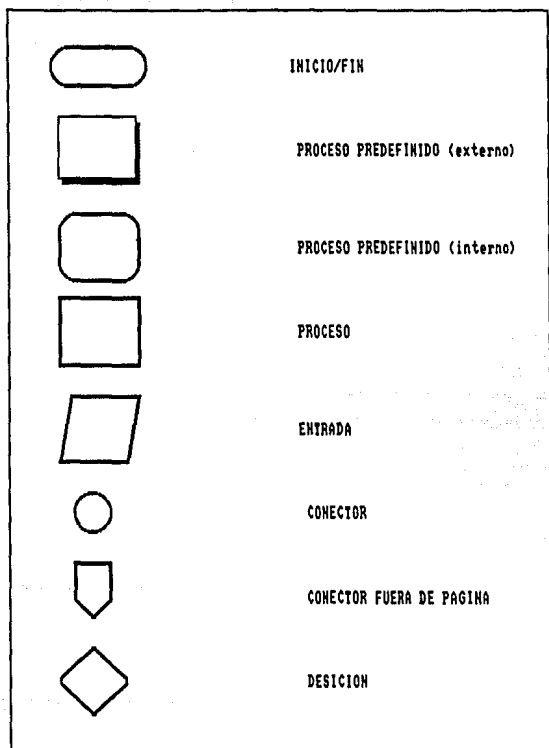


FIGURA 19. SIMBOLOS UTILIZADOS EN LOS DIAGRAMAS.

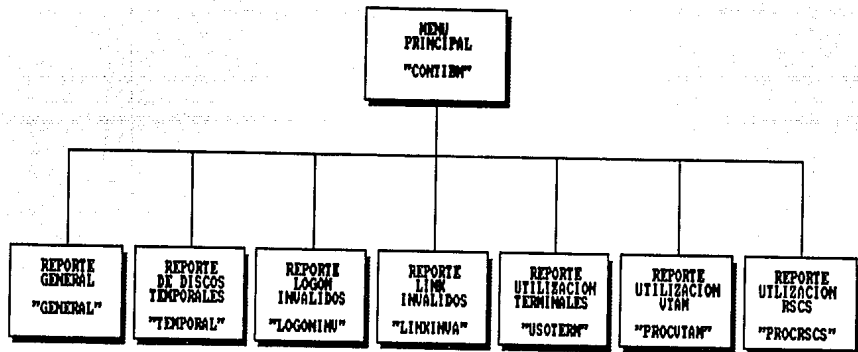


FIGURA 20. DIAGRAMA DE ESTRUCTURA QUE MUESTRA EL MAS ALTO NIVEL EN EL DISEÑO DEL SISTEMA DE CONTROL DE RECURSOS. ESTE DIAGRAMA RESUME EL SISTEMA COMPLETAMENTE.

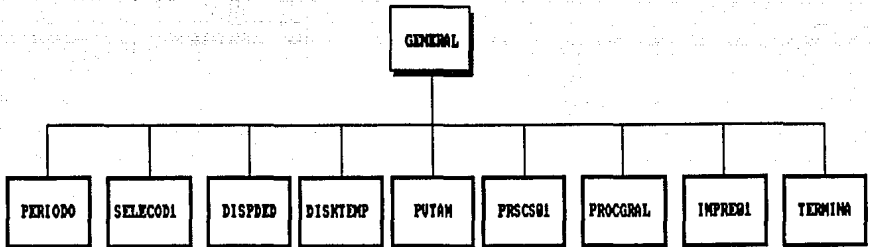


FIGURA 21. DETALLE DEL DIAGRAMA DE ESTRUCTURA QUE MUESTRA LOS PRINCIPALES PROCESOS QUE GENERAN EL REPORTE GENERAL DE UTILIZACION DE RECURSOS.

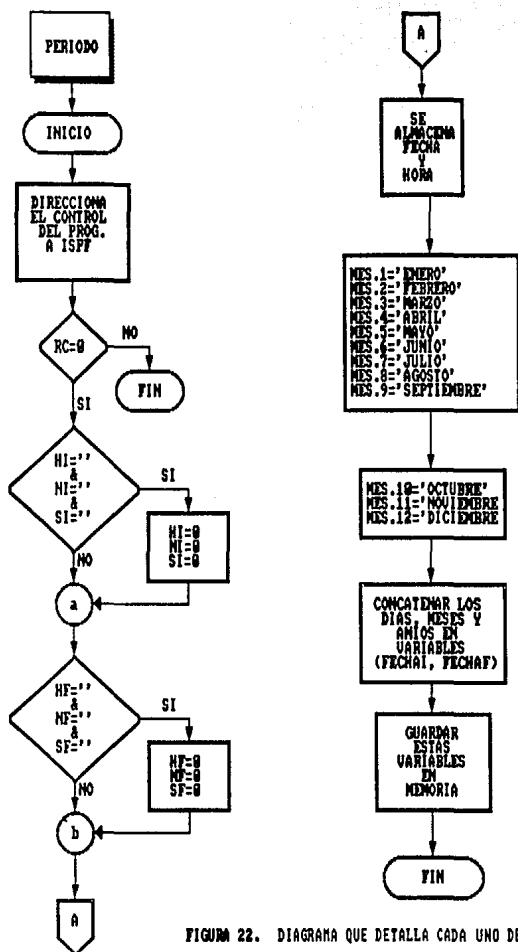


FIGURA 22. DIAGRAMA QUE DETALLA CADA UNO DE LAS INSTRUCCIONES DEL PROCESO DE SELECCION DE LOS REGISTROS EN EL RANGO ELEGIDO.

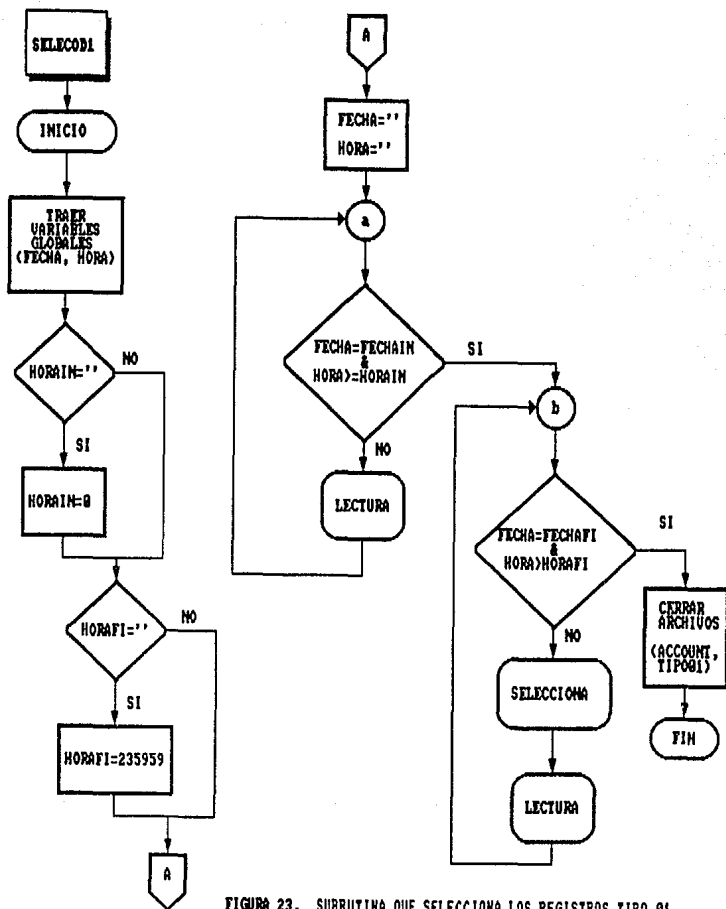


FIGURA 23. SUBRUTINA QUE SELECCIONA LOS REGISTROS TIPO 01.

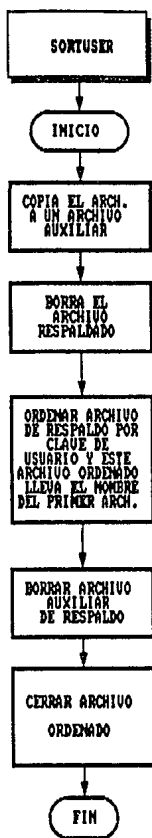


FIGURA 24. DIAGRAMA QUE DETALLA TODAS LAS INSTRUCCIONES PARA CLASIFICAR LOS ARCHIVOS POR EL CAMPO CLAVE DE USUARIO.

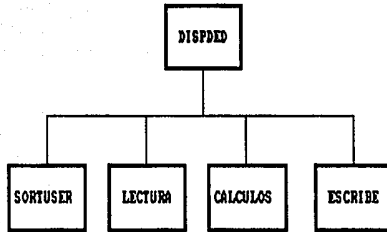


FIGURA 25. DIAGRAMA DE UN NIVEL INTERMEDIO QUE MUESTRA EL PROCESO PARA DISPOSITIVOS DEDICADOS.

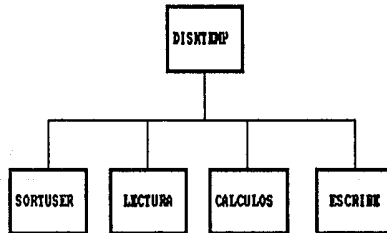


FIGURA 26. DIAGRAMA DE UN NIVEL INTERMEDIO QUE MUESTRA EL PROCESO PARA DISCOS TEMPORALES.

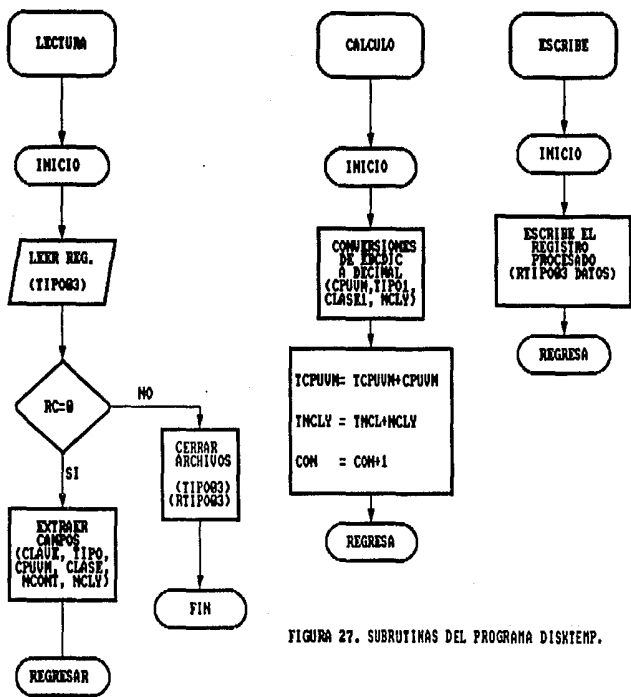


FIGURA 27. SUBROUTINAS DEL PROGRAMA DISKTEMP.

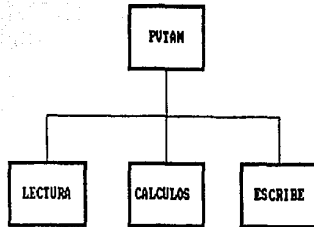


FIGURA 28. DIAGRAMA DE UN NIVEL INTERMEDIO QUE MUESTRA EL PROCESO PARA UTILIZACION DE UTAM.

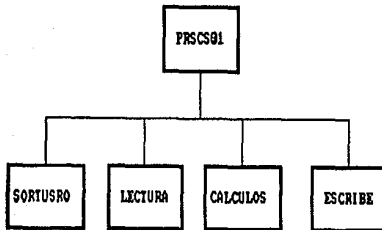


FIGURA 29. DIAGRAMA DE UN NIVEL INTERMEDIO QUE MUESTRA EL PROCESO PARA UTILIZACION DE RSCS.

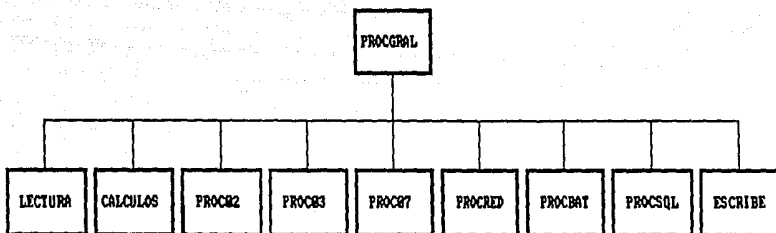


FIGURA 30. DIAGRAMA DE UN NIVEL INTERMEDIO QUE MUESTRA EL PROCESO DONDE SE RECOPILA TODA LA INFORMACION ALMACENADA EN ANTERIORES PROCESOS PARA CADA USUARIO.



FIGURA 31. DIAGRAMA QUE DETALLA LAS INSTRUCCIONES DEL PROCESO DE CLASIFICACION DE LOS REGISTROS DE UN ARCHIVO POR DEPENDENCIA.

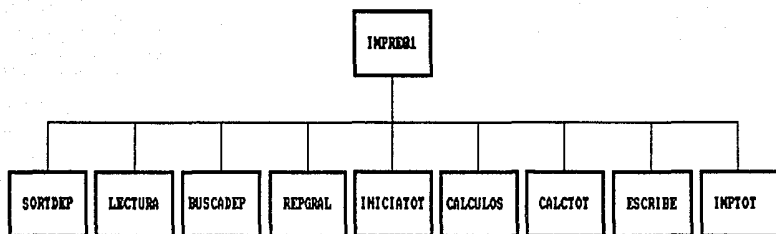


FIGURA 32. DIAGRAMA DE UN NIVEL INTERMEDIO QUE MUESTRA EL PROCESO QUE PERMITE GENERAR UN ARCHIVO CON LOS ENCAEZADOS Y DATOS PARA EL REPORTE GENERAL.

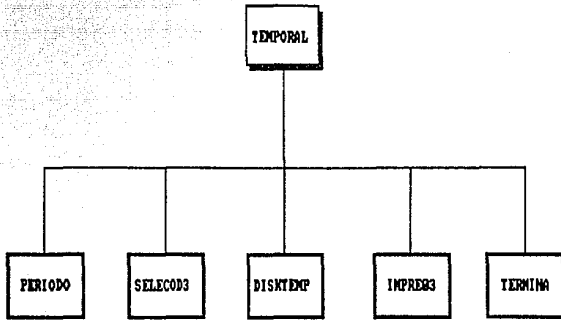


FIGURA 33. DIAGRAMA DE UN NIVEL INTERMEDIO QUE MUESTRA EL PROCESO QUE GENERA EL REPORTE SOBRE UTILIZACION DE DISCOS TEMPORALES.

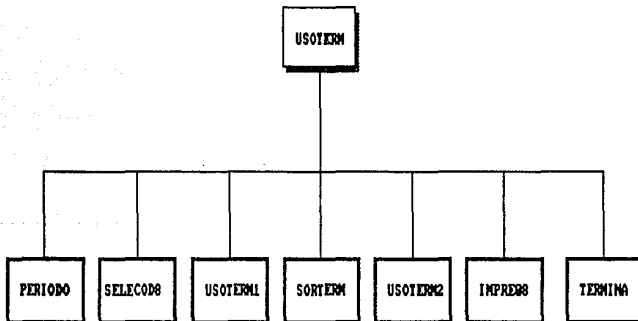


FIGURA 34. DIAGRAMA DE UN NIVEL INTERMEDIO QUE MUESTRA EL PROCESO QUE GENERA EL REPORTE SOBRE UTILIZACION DE TERMINALES.

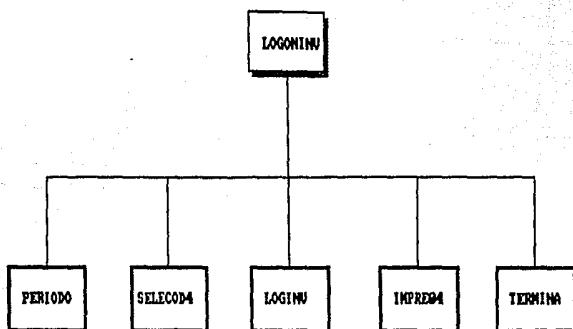


FIGURA 35. DIAGRAMA DE UN NIVEL INTERMEDIO QUE MUESTRA EL PROCESO QUE GENERA EL REPORTE SOBRE LOGON INVALIDOS.

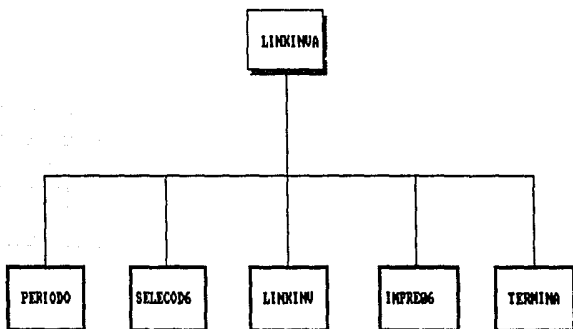


FIGURA 36. DIAGRAMA DE UN NIVEL INTERMEDIO QUE MUESTRA EL PROCESO QUE GENERA EL REPORTE SOBRE LINK INVALIDOS.

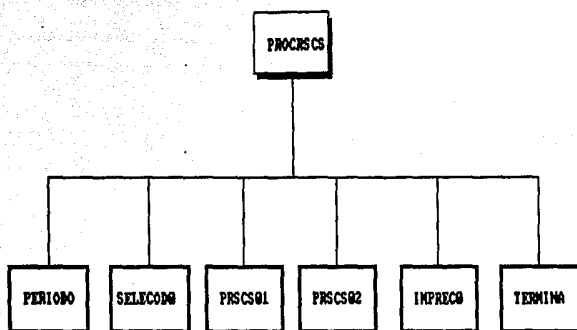


FIGURA 32. DIAGRAMA DE UN NIVEL INTERMEDIO DEL PROCESO QUE GENERA EL REPORTE SOBRE UTILIZACION DE RSCS.

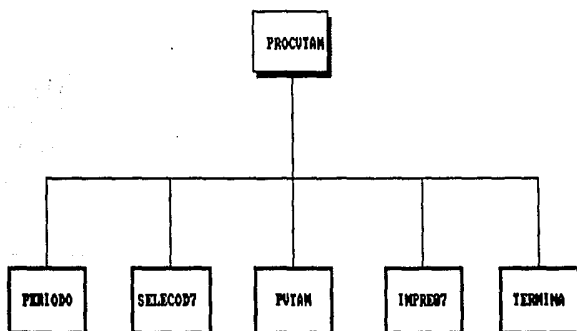


FIGURA 33. DIAGRAMA DE UN NIVEL INTERMEDIO DEL PROCESO QUE GENERA EL REPORTE SOBRE UTILIZACION DE VTAM.

CONCLUSIONES

Una vez construídos y codificados los programas, se procedió a realizar pruebas a cada uno de los módulos que forman el sistema. Primero se prueban las subrutinas externas en forma individual para comprobar su funcionamiento. En segunda instancia, se prueban los programas principales para cada tipo de reporte, con todas sus rutinas internas y externas, verificando que generen los archivos deseados. Posteriormente, se implementa el sistema con el uso de librerías que provee ISPF, para crear cada una de las pantallas que interactúan con el usuario. Cabe mencionar que dichas pantallas también se pueden generar utilizando macros del editor XEDIT, en combinación con el lenguaje REXX.

En principio se hacen pruebas con archivos de aproximadamente 30 000 registros, para detectar rápidamente algunas fallas iniciales y medir el tiempo de proceso. Es durante esta etapa que se ve la necesidad de utilizar el compilador del lenguaje REXX, porque debido a la cantidad de información del archivo ACCOUNT CARDS, los programas consumen gran parte del tiempo y recursos del procesador.

También se vió la necesidad de documentar el sistema propuesto, mediante la creación de un manual, que describe sus características en general, y en forma específica algunos requerimientos que se necesitan para funcionar óptimamente, así como todos los diagramas estructurados de todos los procedimientos y su respectiva codificación. Este manual está dirigido tanto para un usuario que no tenga mucho conocimiento sobre su funcionamiento y pueda operarlo normalmente, como a las personas que deseen realizar modificaciones y adecuaciones a los procesos que lo forman.

Este sistema de control de recursos, no puede considerarse todavía un sistema de contabilidad completo, porque carece del cálculo de los costos financieros de cada recurso, que se dejan a criterio de las autoridades competentes, y tampoco genera las cartas a los responsables de cada dependencia, informando sobre el valor monetario de la totalidad de recursos que utilizaron en determinado período. Sin embargo, se pone a consideración un reporte completo (Reporte General de la Utilización de Recursos por Usuario, ver Figura 12.), que a pesar de su complejidad, contempla aspectos importantes, como la utilización de recursos, con alto costo de mantenimiento, como son los dispositivos dedicados o el uso de áreas temporales, que restan espacio en disco y muchas veces no son aprovechados en su totalidad.

Por otra parte, los beneficios del nuevo sistema, se evaluarán, de acuerdo a la aplicación que se le de y a la manera de explotar toda la información que se facilita. Sin embargo, pienso que la información detallada en los reportes, es muy importante para la administración de este equipo de cómputo, porque actualmente existe muy poca documentación al respecto. Al describir minuciosamente el archivo ACCOUNT CARDS en el capítulo 4, se provee de una herramienta de trabajo bastante útil, permitiendo desarrollar reportes aún más específicos, de acuerdo a las necesidades que se presenten posteriormente.

También, se debe mencionar, que existe en el mercado un producto llamado VMACCOUNT, desarrollado por The Systems Center y distribuido en México por Integración de Tecnosistemas, S.A. de C.V., con un sistema completo de contabilidad para ambientes VM, que facilita:

- Información de los costos de operación.
- Definir los límites de los presupuestos.
- Identificar la utilización de recursos dirigido a pronosticar su demanda, justificar la compra de hardware nuevo y medir productividad.
- Identificar la utilización de productos mediante: recuperación de costos, distribuir costos de los paquetes de software entre los usuarios, determinar niveles de uso.

VMACCOUNT, es por lo tanto otra opción que se tiene para cubrir la necesidad de un sistema de contabilidad, pero es más interesante desarrollar un sistema propio dentro de la dependencia, más acorde a las necesidades del equipo y de las exigencias que requiere su administración y operación.

Después de esta experiencia, se puede observar la gran cantidad de procesos que se deben sistematizar para un mejor aprovechamiento y optimización de un equipo de cómputo grande como es el procesador 4381, que hoy en día permite principalmente a la comunidad universitaria, tener una herramienta que le facilite el desarrollo de proyectos de investigación, ofreciendo el soporte a una amplia gama de productos, y además facilitar la comunicación nacional e internacional a través del correo electrónico, para intercambiar experiencias e información.

APENDICE A

TIPOS DE DISPOSITIVOS DE ACCESO DIRECTO

Los diferentes tipos y modelos de almacenamiento de acceso directo tienen algunos atributos comunes relativos a los componentes físicos de las unidades y la forma en la que estos componentes funcionan almacenando datos. Nos referiremos específicamente a los tipos de dispositivos de almacenamiento de acceso directo o DASD del inglés Direct Access Storage Device.

Los discos son medios de almacenamiento de información. Pueden estar ordenados horizontal o verticalmente, dependiendo del tipo de DASD. Al girar los discos, las cabezas (que están en los brazos de acceso), escriben y leen datos (bit patterns) en la superficie del disco. Las cabezas de lectura/escritura no tienen contacto con la superficie del disco pero permanecen encima de ésta, a una distancia mínima de modo que flotan sobre un amortiguador de aire. Los brazos de acceso y las cabezas de lectura/escritura retroceden y adelantan para que ambos lados de su superficie puedan ser accedidos.

La superficie de los discos está subdividida en círculos concéntricos separados, llamados pistas o tracks, que almacenan los datos. En un tipo de DASD dado, cada pista puede tener capacidad para almacenar la misma cantidad de datos (número de bytes). El número de pistas por superficie de disco y el número de datos que puede almacenar cada pista varía según el tipo de DASD, ya sea CKD o FBA, como se explica más adelante.

La unidad de datos almacenados en disco es un registro físico, el cual contiene el dato y el control de la información, que describe el dato y permite que sea controlado para corrección cuando empieza a leer. El registro físico es la unidad de dato que se transfiere entre el disco y el procesador.

Existen dos tipos de arquitectura de discos, CKD y FBA, que a continuación se describen.

Cada pista tiene capacidad para uno o más registros, que se numeran secuencialmente. En el caso de los dispositivos de tipo CKD (Count-Key-Data), cada uno contiene un área de cuenta (con un ID del área del dato que sigue), un área llave opcional (identifica la información en el área del dato del registro) y un área de datos (con los datos que han sido organizados y ordenados por el programador), separados entre sí por un gap (o espacio sin grabación).

Para registros en dispositivos de tipo FBA (Fixed-Block Architecture), el área de datos de cualquier registro físico tiene un tamaño estándar que está predeterminado para un tipo de disco de almacenamiento. El usuario no está involucrado con la especificación del número de bytes para el área de datos, y el control de la información se escribe sobre la pista en un modelo pre-establecido durante el proceso. Los registros físicos están separados por un gap, o un espacio sin uso, en la superficie del disco.

Cabe mencionar que los discos de arquitectura CKD, tienen un mayor costo que los de FBA, y se usan generalmente en computadores grandes; sin embargo en los discos FBA, se desperdicia más espacio, en el caso de almacenar registros de tamaño variable.

APENDICE B

CLASES Y TIPOS DE DISPOSITIVOS

El código **DIAGNOSE X'24'** de CP, provee una máquina virtual con información acerca de la identificación y estado de un dispositivo virtual específico. La máquina virtual debe especificar el dispositivo virtual para el cual requiere información. CP regresa la información acerca del dispositivo virtual y asociarlo a un dispositivo real en Rx, Ry y Ry + 1 registros. Los códigos son:

CODIGOS PARA CLASES DE DISPOSITIVOS.

Código	Clase de dispositivo
X'80'	Terminal
X'40'	Dispositivos Gráficos
X'20'	Unidad de Registros de Entrada
X'10'	Unidad de Registros de Salida
X'08'	Dispositivo de Cinta Magnética
X'04'	Dispositivos de Almacenamiento de Acceso Directo
X'02'	Dispositivos Especiales
X'01'	Almacenamiento en bloques fijos

CODIGOS PARA TIPOS DE DISPOSITIVOS.

* PARA TERMINALES.

Código	Tipo de dispositivo
X'80'	Línea Binaria Sincrónica Remota
X'40'	Línea Binaria Sincrónica 2700
X'40'	Línea de Comunicación 2955
X'30'	Consola Encendida/Apagada
X'20'	Control de Terminal Telegráfica Tipo II
X'20'	Terminal Teletipo
X'1C'	Terminal Indefinida
X'18'	Terminal de Comunicación IBM 2741
X'14'	Sistema de Comunicación de Datos IBM 1050
X'10'	Terminal de Control IBM Tipo I
X'08'	Control de Liga de Datos Sincrónicos
X'00'	Consola IBM 3210
X'00'	Consola IBM 3215
X'00'	Consola IBM 2150
X'00'	Consola IBM 1052

*** PARA DISPOSITIVOS GRAFICOS.**

Código	Tipo de dispositivo
X'C0'	Dispositivos Gráficos de Alta Función
X'80'	Unidad de Despliegue IBM 2250
X'40'	Estación de Despliegue IBM 2260
X'20'	Estación de Despliegue IBM 2265
X'10'	Consola IBM 3066
X'08'	Impresora IBM 1053
X'04'	Sistema de Consola IBM 3138
X'04'	Sistema de Consola IBM 3148
X'04'	Sistema de Consola IBM 3158
X'04'	Estación de Despliegue IBM 3277
X'01'	Estación de Despliegue IBM 3278
X'01'	Estación de Despliegue IBM 3279
X'01'	Panel de Información IBM 3290
X'02'	Impresora IBM 3284
X'02'	Impresora IBM 3286
X'02'	Impresora IBM 3287
X'02'	Impresora IBM 3288
X'02'	Impresora IBM 3289-E
X'02'	Impresora IBM 4250

*** PARA UNIDADES DE REGISTROS DE ENTRADA.**

Código	Tipo de dispositivo
X'90'	Tarjeta Lectora/Perforadora IBM 2520
X'88'	Tarjeta Lectora/Perforadora IBM 1442
X'84'	Tarjeta Lectora IBM 3505
X'82'	Tarjeta Lectora IBM 2540
X'81'	Tarjeta Lectora IBM 2501
X'80'	Tarjeta Lectora
X'40'	Timer
X'24'	Lectoras de Cintas de Papel IBM 1017
X'22'	Lectoras de Cintas de Papel IBM 2671
X'21'	Lectora de Cartucho de Cinta Magnética IBM 2495
X'20'	Cinta Lectora

*** PARA UNIDADES DE REGISTROS DE SALIDA.**

Código	Tipo de dispositivo
X'90'	Tarjeta Perforadora IBM 2520
X'88'	Tarjeta Perforadora IBM 1442
X'84'	Tarjeta Perforadora IBM 3525
X'82'	Tarjeta Perforadora IBM 2540
X'80'	Tarjeta Perforadora
X'4D'	Subsistema de Impresión IBM 3800 Modelo 8
X'4B'	Impresora IBM 4248
X'4A'	Impresora IBM 4245

X'49' Subsistema de Impresión IBM 3800 Modelo 3
X'47' Impresora IBM 3262
X'46' Impresora IBM 3289
X'45' Subsistema de Impresión IBM 3800 Modelo 1
X'44' Impresora IBM 1443
X'43' Impresora IBM 3203
X'42' Impresora IBM 3211
X'41' Impresora IBM 1403
X'40' Impresora
X'24' Perforadora de Cintas de Papel IBM 1018
X'20' Cinta Perforadora

*** PARA CINTA MAGNETICA.**

Código Tipo de dispositivo
X'82' Unidad de Cinta IBM 3422
X'80' Unidad de Cinta IBM 2401
X'40' Unidad de Cinta IBM 2415
X'20' Unidad de Cinta IBM 2420
X'10' Unidad de Cinta IBM 3420
X'08' Unidad de Cinta Ibm 3410/3411
X'04' Unidad de Cinta IBM 8809
X'02' Unidad de Cinta IBM 3430
X'01' Unidad de Cinta IBM 3480

*** PARA DISPOSITIVOS DE ALMACENAMIENTO DE ACCESO DIRECTO.**

Código Tipo de dispositivo
X'80' Tambor Paralelo IBM 2301
X'80' Tambor Serial IBM 2303
X'80' Unidad de Almacenamiento de disco IBM 2311
X'80' Unidad de Celda de Datos IBM 2321
X'40' Disco de Facilidad de Almacenamiento IBM 2314
X'40' Disco de Facilidad de Almacenamiento IBM 2319
X'20' Disco de Facilidad de Almacenamiento IBM 3380
X'10' Disco de Facilidad de Almacenamiento IBM 3330
X'10' Disco de Almacenamiento y Control IBM 3333
X'08' Disco de Facilidad de Almacenamiento IBM 3350
X'04' Disco de Facilidad de Almacenamiento IBM 3375
X'02' Dispositivo de Almacenamiento de Cabeza Fija IBM 2314
X'01' Disco de Facilidad de Almacenamiento IBM 3340

*** PARA DISPOSITIVOS ESPECIALES.**

Código	Tipo de dispositivo
X'80'	Dispositivo Canal a Canal (CTCA o 3088)
X'40'	Controlador de Comunicaciones Programable 37XX
X'20'	Controlador de Almacenamiento Mass 3851
X'04'	Dispositivo SRF (7443)
X'01'	Dispositivo no soportado por VM/SP

*** PARA DISPOSITIVOS DE ALMACENAMIENTO DE BLOQUE FIJO.**

Código	Tipo de dispositivo
X'05'	3370-4
X'02'	3370, Modelo A1, A2, B1 y B2
X'01'	3310
X'00'	Bloques Fijos Genéricos (Este código se aplica a los dispositivos a los que no se ha determinado un tipo específico dentro de CP)

CODIGOS PARA CARACTERISTICAS DE DISPOSITIVOS.

*** PARA IMPRESORAS.**

Código	Tipo de dispositivo
X'80'	3800 tiene cuatro modificaciones Gráficas de Caracteres Writable (WCGM)
X'40'	Extended Sense Bytes
X'01'	UCS

*** PARA CINTAS MAGNETICAS.**

Código	Tipo de dispositivo
X'80'	Dispositivo Canal a Canal (CTCA o 3088)
X'40'	Controlador de Comunicaciones Programable 37XX.
X'20'	Controlador de Almacenamiento Mass 3851.
X'04'	Dispositivo SRF (7443)
X'01'	Dispositivo no soportado por VM/SP.

*** PARA DISPOSITIVOS DE ALMACENAMIENTO DE BLOQUE FIJO.**

Código	Tipo de dispositivo
X'05'	3370-4
X'02'	3370, Modelo A1, A2, B1 y B2
X'01'	3310
X'00'	Bloques fijos Genéricos (Este código se aplica a los dispositivos a los que no se les ha determinado un tipo específico dentro de CP)

BIBLIOGRAFIA

- Claudon, Kenneth & Claudon, Price Jane**
"Management Information Systems:
A contemporary Perspective"
Macmillan Publishing Company
U.S.A., 1988
- Deitel, Harvey M.**
"Introducción a los Sistemas Operativos"
Tr. Andrés Leyra Denaro
Addison-Wesley Iberoamérica
SITESA (Sistema Técnicos de Edición S.A. de C.V.)
México, 1987.
- Forsythe, Jason & Scott Leibs**
"Virtual no end to VM potential"
Information WEEK, 40
U.S.A., Enero 1989
- Fosdick, Howard**
"VM/CMS Handbook for Programmers, Users, and Managers"
Hayden Books
Business Information Processing Library
Second Edition
U.S.A., 1988
- International Business Machines Corporation**
Virtual Machine
Remote Spooling Communications Subsystem Networking Version 2
"Diagnosis Reference" Release 2
LY24-5228-1
Second Edition
U.S.A., 1986
- Virtual Machine/System Product**
"Administration"
Release 6
SC24-5285-01
Second Edition
U.S.A., 1988

Virtual Machine/System Product
"CMS Command Reference"
Release 6
SC19-6209-05
Sixth Edition
U.S.A., 1988

Virtual Machine/System Product
"CP General User Command Reference"
Release 6
SC19-6211-05
Sixth Edition
U.S.A., 1988

Virtual Machine/System Product
"Introduction"
Release 6
GC19-6200-05
Sixth Edition
U.S.A., 1988

Virtual Machine/System Product
"Planning Guide and Reference"
Release 5
Sc19-6201-5
Sixth Edition
U.S.A., 1986

Virtual Machine/System Product
"Systems Facilities for Programming"
Release 5
SC24-5288-0
First Edition
U.S.A., 1986

"SQL/Data System Planning and
Administration for VM/System Product"
Release 3.5
SH24-5043-2
Third Edition
U.S.A., 1985

Virtual Machine/System Product
"System Product Interpreter Reference"
Release 6
SC24-5239-03
Fourth Edition
U.S.A., 1988

Virtual Machine/System Product
"System Product Interpreter User's Guide"
Release 6
SC24-5238-04
Fifth Edition
U.S.A., 1988

VMACCOUNT
"Planning Guide"
Release 3.0
U.S.A., 1988

Márquez, Vité Juan Manuel
"Sistema de Información por Computadora:
Metodología de desarrollo"
Edit. Trillas
México D.F., 1987

Rall, Patrick R.
"Why CMS?"
Research Report
IBM Research Division
T.J. Watson Research Center, Hawthome
U.S.A., Agosto 1988

Senn, James A.
"Análisis y Diseño de Sistemas de Información"
Tr. José Lara Portal
Edit. MacGraw-Hill
México, 1988