



UNIVERSIDAD NACIONAL
AUTÓNOMA

UNIVERSIDAD NACIONAL AUTÓNOMA DE MEXICO

FACULTAD DE INGENIERIA

DESARROLLO DE UN SISTEMA GENERADOR
DE BANCOS DE DATOS VIA UN
MICROCOMPUTADOR

T E S I S

QUE PARA OBTENER EL TITULO DE:
INGENIERO EN COMPUTACION

PRESENTA:

HECTOR CAMPOS ESTRADA

DIRECTOR DE TESIS

ING. SEBASTIAN POBLANO ORDONEZ

MEXICO, D.F.

1985.



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.

TESIS CON FALLA DE ORIGEN

I N D I C E.

	Pag.
INTRODUCCION	4
1 Planteamiento	1
1.1 Antecedentes	1
1.2 Definición del Problema	3
1.3 Requerimientos	5
1.4 Metas	6
2 Análisis del Sistema	7
2.1 Sistema de Captura de Datos	8
2.1.1 Análisis de Factibilidades	8
2.1.2 Análisis del Sistema	27
2.2 Alternativas de Comunicación de los Equipos ...	36
2.3 Generación de Bancos de Información en el Equipo Central	43
3 Estudio de los Equipos	45
3.1 Desarrollo Tecnológico	45
3.2 Equipo Cromemco	51
3.2.1 Aspectos Generales del Sistema	51
3.2.2 Intrínsecos del sistema	62
3.2.3 Utilerías del sistema	66
3.2.4 Paquetes del sistema	73
3.2.5 Lenguajes de programación	75
3.2.6 Características de Hardware	82
3.3 Equipo Burroughs B-6700	92
3.3.1 Hardware del Sistema	92
3.3.2 Software del Sistema	99
3.3.3 Software de Comunicación	119
4 Diseño del Sistema	137
4.1 Diseño General del Sistema	137
4.2 Diseño del Sistema de Captura de Datos	142
4.3 Diseño del Sistema Manejador de Mensajes	149
5 Implementación del Sistema	160
5.1 Organización del proceso de captura	160
5.1.1 Descripción del Sistema	162
5.2 Operación del Sistema de Captura	166
6 Documentación del Sistema	169
6.1 Manual de Operación del Sistema de Captura	169
6.1.1 Descripción del Sistema	169
6.1.2 Operación del Sistema	171
6.1.3 Lista resumida de los comandos	190
7 Conclusiones	183
7.1 Bibliografía	184

INTRODUCCION

Dado el gran avance de la tecnología y a su creciente aplicación en las actividades del desarrollo interno de la U.N.A.M., se ha visto la necesidad de implementar un sistema generador de bancos de información que facilite la ejecución de las diversas funciones que necesitan del apoyo de la computación.

En la actualidad gran parte de las dependencias de la Universidad que utilizan la computadora como apoyo en sus procesos operativos, realizan o canalizan la captura de datos a través de equipos de perforación, los que en el presente han llegado a su estado de obsolescencia, pues su fabricación, ya no se realiza o se efectúa en menor escala.

Es por ello, que ha sido necesario buscar nuevas posibilidades para llevar a cabo la captura de información, en base a dos propósitos fundamentales, reducir los costos y aumentar la productividad. Esta búsqueda de posibilidades nos llevó a definir que el proceso de captura se pudiera realizar por medio de una microcomputadora, de tal suerte que, si se lograra en ella realizar la captura y validación en forma simultánea aumentaría la producción. En cuanto al costo, el beneficio radica en las posibilidades de almacenamiento, eliminando con ello las tarjetas (además de los costos de mantenimiento de los equipos).

Además, si se conecta el equipo microcomputador al equipo central, ofrecería rapidez en el uso de los datos captados y apoyo en la edición de programas o archivos en forma local, con lo cual se lograría un alto beneficio para los computadores grandes instalados en la U.N.A.M.

El presente trabajo propuesto como tesis describe el desarrollo y la implementación de un sistema generador de bancos de información contando con las facilidades de captura, verificación y edición de datos via una microcomputadora y con posibilidades de conexión de la microcomputadora a un equipo central ampliando con ello las facilidades del sistema.

Con el fin de mostrar ampliamente las facilidades del sistema, así como su desarrollo y aplicación se ha considerado conveniente introducir en este trabajo todo lo concerniente a las características y componentes de los equipos que intervienen, así como, las condiciones para establecer la comunicación entre ellos.

El presente trabajo esta dividido en siete capítulos en los cuales se explica el procedimiento utilizado para el logro del objetivo planteado en el primer capítulo. Por el momento, se dará un panorama general de los puntos que se tratan en cada uno de los capítulos.

En el primer capítulo se expondra lo que se quiere realizar dentro de la tesis (objetivos y metas), en el segundo capítulo se hará un análisis del sistema a realizar. En el tercer capítulo, se presentan las características de los equipos que se van a utilizar, como es el equipo CROMEMCO y BURROUGHS, tomando especial énfasis en el primero, en lo que respecta a el manejo de archivos, el tipo de lenguaje a

utilizar para la programación del sistema, el manejo de la pantalla, esto es, posicionamiento del cursor, atributos especiales de la pantalla y controles del teclado. Sobre el equipo BURROUGHS se dará un panorama general, dado que es un equipo de gran complejidad, se estudiara mas el subsistema de E/S; además, el objetivo de la tesis, no es profundizar en su funcionamiento. En el capítulo cuarto se expondrá el diseño del sistema, esto es, la estructura de datos que deberá tener y las funciones que deberá realizar para cumplir con los requerimientos del sistema planteados en el análisis. La implementación del sistema y los ajustes y puestas a punto de él, se darán en el capítulo cinco. En el capítulo número seis se dará la documentación que se realizó, para la operación del sistema y la organización propuesta para el proceso de captura de información. Finalmente el capítulo siete contendrá las conclusiones del desarrollo del sistema así como las ventajas y desventajas encontradas en la metodología que se utilizó para el desarrollo del mismo.

En este trabajo no se incluyen los listados de los programas ni la documentación entera del sistema, por ser muy voluminosa, pero se tienen a la disposición.

CAPITULO I

1 Planteamiento

1.1 Antecedentes.

Los antecedentes del presente sistema surgen, como se expuso brevemente en la introducción, debido a tres razones principales, la primera está en función de la captura de información, la segunda, con el trabajo y tiempo extra que se requiere como consecuencia del uso de un método antiguo y la tercera, a la carga que tienen los sistemas grandes en cuanto a la realización y edición de programas fuente por medio de terminales remotas conectadas en línea al equipo central. A continuación se exponen a detalle estas causas:

1) Dado que el proceso de captura de datos se realiza por medio de tarjetas perforadas, el cual es un método anticuado, considerando el gran avance de la tecnología electrónica dentro del campo de la computación, la fabricación de estos equipos ya no se realiza o se hace en forma restringida para algunas marcas y equipos de esta naturaleza, lo cual nos lleva a que el costo de la captura de información y el mantenimiento del equipo sea muy elevado, además el tiempo para reparar cualquier falla de un equipo sea muy grande teniendo estos que estar por mucho tiempo sin funcionar. Esto ha implicado que se disminuya la capacidad de producción de dichos sistemas. Debido a esto y a la producción de equipos más elaborados como son las microcomputadoras, se pensó en desarrollar el presente sistema.

ii) Para que la información captada mediante tarjetas sea confiable, se hace necesario tener programas de filtrado para cada tipo de aplicación que se requiera, como consecuencia, cada sistema que necesita de información que es captada por medio de perforadoras, se le debe desarrollar un programa que valide los datos. Si se considera el costo de tiempo-hombre y tiempo-maquina, resulta evidente que es alto, si aunamos a esto como consecuencia, el retraso en la obtención de resultados, se siente la necesidad de un cambio radical en la utilización de la metodología actual.

iii) Existe también el hecho de que la computadora central esta sobrecargada de trabajo debido al gran número de usuarios que tiene, y que utilizan las facilidades del teleproceso para diversas funciones como la actualización de archivos, la generación y edición de programas, para consultas a bancos de información. Por estas razones se pensó en un equipo fuera de línea que le ayudara en cierta forma al equipo central a realizar gran parte de las funciones antes descritas.

1.2 Definición del Problema.

Como se ha podido notar en el punto anterior, nos enfrentamos a un problema que necesita una solución rápida que satisfaga las requerimientos actuales con un menor costo pero con mayor rapidez y confiabilidad. Por tales motivos, el sistema que se pretende desarrollar debe contemplar todos los aspectos antes planteados y más, es decir, debe satisfacer las necesidades actuales en lo que a captura de información (tanto de datos como programas) se refiere, debe reducir el trabajo de los sistemas centrales y disminuir el tiempo en la obtención de resultados al desarrollar sistemas de información.

A lo largo del desarrollo de este trabajo nos enfrentaremos básicamente a los siguientes aspectos:

- 1) Se debe seleccionar un equipo de cómputo (microprocesador) que cumpla con grandes flexibilidades, no solo en lo que respecta al software, sino también, a las facilidades de su hardware para conectarlo con el sistema central.

- 2) El sistema que se desarrolle en dicho equipo debe permitir todas las facilidades que un sistema de captura pueda proporcionar por ejemplo: definición de formatos de cualquier tipo (campos de longitud variable, manejo de

información constante, validación al momento de captar) manteniendo alta velocidad en su respuesta, de tal forma que el usuario no tenga que esperar mucho tiempo, mientras el sistema verifica que cada campo que se este captando está correcto. finalmente, el sistema debe tener la facilidad de modificar la información que anteriormente haya sido captada.

3) Se requiere facilidad en la definición de bancos de información en el microcomputador, con opción a ser enviados al equipo central. El microcomputador deberá trabajar fuera de línea y se conectará en un momento dado con el sistema central solo para enviarle o recibir información en bloque.

En este trabajo se describe el sistema que resuelve los puntos antes planteados.

1.3 Requerimientos.

Para poder desarrollar el sistema que se pretende se requiere de :

- Análisis y desarrollo de un sistema de captura.
- Conocimiento de las características del Microcomputador que se va a utilizar.
- Evaluación del lenguaje a utilizar para la programación del sistema en el microcomputador.
- Evaluación de las restricciones del sistema microcomputador
- Conocimiento del equipo Central para la transmisión de archivos entre los equipos.
- Comunicación del equipo Central y Microcomputador.
- Evaluación de alternativas para la transmisión.

1.4 Metas.

La definición de las metas a alcanzar dentro de esta tesis en lo que se refiere al sistema son las siguientes:

- i) Creación y validación de un archivo en un Microcomputador
- ii) Creación y validación de un archivo en un microcomputador teniendo la opción de enviarlo al equipo central.
- iii) Creación, validación y edición de archivos que estén ya sea en el Microcomputador ó en el equipo central.
- iv) Edición de programas fuente en el microcomputador y ejecución de los mismos en el sistema central.

Para poder cumplir con estas metas, será necesario conocer el equipo microcomputador con el cual se va a trabajar, conocer el equipo BURROUGHS, (que es el sistema central al cual van a estar conectados los equipos microcomputadores) en su parte de comunicación de datos, teleproceso, configuración de la red, lenguaje NDL y lenguaje DCALSO principalmente.

CAPITULO II

2 Análisis del Sistema

Para realizar el análisis del sistema, se dividió éste en tres partes o subsistemas: la primera es la concerniente a la forma de captura de la información, la segunda es la comunicación entre los equipos y la tercera a la generación de los bancos de datos en el equipo Burroughs ya en sí por medio de una Microcomputadora. Estos tres subsistemas se verán por separado, dentro del análisis y en cada caso se comentarán los detalles que sean comunes entre los subsistemas además de que ya para fines de diseño la parte de enlace de los equipos y generación de los bancos de información será uno sólo. A continuación comenzaremos con el sistema de captura de datos.

2.1 Sistema de Captura de Datos.

Para poder realizar el análisis del sistema de captura de información se realizaron una serie de reuniones con el personal del departamento de perforación del Centro de Servicios de Computo de la UNAM, de las cuales se obtuvieron las necesidades y requerimientos operativos del sistema de captura. Además también se contó con la prueba de un sistema de captura de datos de un proveedor del cual se obtuvieron sus ventajas y desventajas, estas pruebas fueron realizadas por el personal del departamento de captura los cuales dieron sus puntos de vista y en base a estos y a otras consideraciones que se anotarán posteriormente se realizó el sistema en su parte de captura.

La primera fase del desarrollo del sistema sería hacer el análisis de factibilidad el cual se presenta en los siguientes parrafos.

2.1.1 Analisis de Factibilidades.

Dado el gran crecimiento en la industria de la electrónica, y particularmente en lo que a microprocesadores se refiere, es necesario hacer un estudio sobre la posibilidad de utilizar microcomputadoras, que mediante un sistema de captura y validación simultánea, reduzcan los costos que implican el tener equipos de perforación. Este estudio no fue realizado como parte de la presente tesis dado que este ya había sido realizado y se encuentra descrito en el documento "Estudio de Factibilidad para la Implantación de

"Microprocesadores de apoyo en la Universidad" realizado en el CSC. Por lo tanto, no se vió la necesidad de volver a realizar dicho análisis. El estudio de factibilidad comprende la selección de una microcomputadora que reúna las características más sobresalientes de todas las que se encuentran en el mercado, dentro de las consideraciones ideales requeridas de los equipos de éste tipo, son: la posibilidad de proceso local y facilidades para la comunicación con un equipo central de cómputo.

En este trabajo se darán los puntos que se consideran más relevantes en cuanto al estudio de factibilidades realizado.

Consideraciones Generales.

Para la evaluación de un equipo en particular hay que tomar en cuenta principalmente los siguientes aspectos:

i) Posibilidades locales.

- Facilidades del sistema operativo.
- Lenguajes y software existentes.
- Facilidades de expansión.
- Posibilidad de multiusuario.
- Compatibilidad con otros sistemas.
- Capacidades de almacenamiento locales.
- Facilidades de edición de archivos.

ii) Procesamiento remoto.

- Comunicación como terminal.
- Transferencia y recepción de archivos.
- Facilidad de uso.

iii) Características generales.

- Confiabilidad.
- Modularidad.
- Presentación.
- Tecnología.
- Flexibilidad.
- Tipo de compañía.

A continuación se explicarán los requerimientos antes mencionados.

- Facilidades del sistema operativo.

Un sistema operativo debe brindar al menos las facilidades de un lenguaje simple de comandos para comunicación con el equipo, el manejo de información en disco, acceso a directorios y archivos que garanticen la seguridad de la información.

La relocalización de los programas en memoria, es importante para el mejor aprovechamiento de esta, la facilidad de multiprogramación para permitir la expandibilidad a multi-usuario.

La compatibilidad con otros sistemas operativos, para los cuales hay una gran variedad de software desarrollado.

- Lenguajes y Software existente.

Debera tener por lo menos alón lenguaje de programación que permita al usuario resolver sus problemas eficientemente, así como un editor y programas de servicio tales como:

Copiado de archivos
Editor de archivos
Manejo de directorios
Rastreador de errores

De preferencia deberá contar con lenguajes tales como: Basic, Fortran, Cobol. Que son los lenguajes más comunes de programación. Además de la posibilidad de tener alón lenguaje estructurado del tipo de Pascal.

- Facilidades de expansión.

Para algunas aplicaciones, el crecimiento puede ser en número de equipos, pero en otras será necesario poder expandir un equipo en tamaño, ya sea memoria, discos o terminales, ya que si en un momento dado llegan a crecer los requerimientos locales del sistema, éste se debe poder expandir hasta un límite razonable a un bajo costo.

- Posibilidad de Multi-Usuario.

Será deseable que no solamente una persona haga uso del sistema, sino que simultáneamente varias personas puedan estar editando, corriendo, creando programas o consultando una base de datos en el mismo equipo y que el sistema sea capaz de hacer todo esto sin entrar en conflicto.

- Compatibilidad con otros sistemas.

En los últimos años ha habido un revolución, en cuanto a equipos de cómputo se refiere y se ha visto la necesidad de que exista una compatibilidad para que algo desarrollado funcione sin muchos cambios en otro sistema de las mismas características generales. Por lo que fué creado un sistema operativo llamado CP/M el cual trata de estandarizar muchos conceptos, de manera que los usuarios de este sistema puedan fácilmente aprender la utilización de algún otro, sin muchas complicaciones.

Es muy importante que el equipo a adquirir, cuente con este sistema operativo CP/M (ó al menos compatible), para tratar de establecer un estándar dentro de la Universidad, y con esto lograr una mayor flexibilidad.

- Facilidades de edición.

La edición de programas o textos es básico en cualquier sistema, por lo tanto, el equipo deberá contar con un editor poderoso, accesible al usuario y de fácil manejo que permita crear archivos y modificarlos. De preferencia deberá contar con manejo de pantalla, para una mayor rapidez en la edición.

- Procesamiento local.

Además de tener un editor, lenguajes y utilerías, el sistema debe demostrar el poder ejecutar los programas de los usuarios, haciendo uso de los puertos del equipo, el acceso a archivos sin entrar en conflicto con el sistema y brindando la posibilidad de ejecutar varios procesos simultáneos como ediciones, compilaciones y ejecuciones.

- Comunicación como terminal.

Debe ser posible que a través de este sistema se pueda establecer una comunicación con el equipo Burroughs de forma tal, que un usuario a través de la terminal del equipo se convierta en una terminal de la Burroughs, tal y como si estuviera conectada directamente, permitiéndole entrar a CANDE a ejecutar programas, lanzar jobs, (trabajos desarrollados en cadena en el computador central con una secuencia establecida) etc., con la ventaja de poder almacenar localmente en la Microcomputadora alguna información o transferir a la Burroughs algún archivo.

Para el enlace se deberá desarrollar el software necesario para lograr la comunicación entre ambas computadoras. El enlace podrá efectuarse a través de una tarjeta de hardware, o por medio de los puertos de la tarjeta del CPU (esto depende del equipo). Hay que definir en el caso de la tarjeta si esta formará parte del hardware estandar de la microcomputadora o bien se puede añadir a la arquitectura sin ningún problema.

- Transferencia y Recepción de Archivos.

El sistema nos debe brindar la posibilidad de mandar y recibir archivos a través de una línea telefónica ó línea directa conectada al equipo Burroughs.

-Facilidad de uso.

Tanto al entrar en modo de terminal; el transferir y recibir archivos, deberá de ser de una forma simple a través de comandos accesibles al usuario que le permitan fácilmente hacer lo que éste desee.

- Confiabilidad.

El equipo deberá ser confiable, o sea que no se descomponga fácilmente, lo que significa que sea un equipo de buena calidad y diseño.

- Modularidad.

Debido a que siempre existe la posibilidad de falla, en un diseño modular es posible el corregir fácilmente una falla cambiando el módulo descompuesto, lo que hace que el servicio al usuario sea casi inmediato. Estos módulos se puedan tener en un inventario de refacciones, además de que en el laboratorio puedan ser corregidos. Otra ventaja de la modularidad es la facilidad de crecimiento.

- Presentación.

Otro punto muy importante es la presentación. Por medio del cual se evitarán posibles fallas, debido al mal manejo de este. Es decir, un equipo del tipo "desarrollo" generalmente viene en una sola tarjeta sin cubierta exterior, lo que podría ocasionar desperfectos constantes.

- Tecnología.

Dados los nuevos avances tecnológicos en el área y la velocidad con que se dan las presentaciones de nuevos sistemas y equipos, es recomendable que se adquiriera un equipo que conjunte en él los últimos avances tecnológicos, que se traducen en una alta integración, un bajo consumo de energía, bajo costo, alta confiabilidad y alta capacidad de procesamiento.

Con los nuevos microprocesadores sería deseable un equipo que hiciera uso de estos tipos de circuitos especializados en cada parte dándole una mayor potencialidad.

- Flexibilidad.

Nos debe permitir el crecimiento, tanto en capacidad de almacenamiento, como en manejo de dispositivos, como Impresoras y en su caso otro tipo de dispositivos especiales según la aplicación específica de cada usuario.

- Tipo de compañía.

En la actualidad existe un gran número de compañías proveedoras de microcomputadoras; la compañía seleccionada deberá ser una que esté perfectamente establecida en el mercado para asegurar con esto la calidad del equipo y la supervivencia de ésta compañía en el mercado a través de los años, de tal manera que pueda brindar un soporte en todas las áreas que se requiera.

Equipos considerados.

A continuación se da una lista de los equipos considerados en el estudio:

Adds	Hewlet Packard
Altos	Intel
Apple II	Radio shack
Archives	Super Brain
Atari	Televideo
Cromemco	Tektronix
Codex	Texas Instruments
Heath Kit	

Datos de Software y Hardware.

A continuación se dan dos tablas comparativas de las características sobresalientes, la primera sobre datos de Hardware y la segunda sobre datos de Software.

DATOS DE HARDWARE

Fabric.	Modelo	C P U			RAM			Ficera
		\$\$	Nombre	Bits Bus	STD	Max	Ficera	
TANDY	TR880 II	3,899	Z80	8	PROPIO	64K	64K	.196M
APPLE	APLEE II	2,500	6502	8	PROPIO	16K	49K	.116M
ATARI	800	2,000	6502	8	PROPIO	16K	49K	.13ME
ALTOS	ACS-8000-E	5,990	Z80A	8	No	64K	200K	1ME
ADDS	MULTIVISION1	3,788	8085A	8	PROPIO	64K	256K	.70ME
VECTOR	VIP	3,499	Z80A	8	S-100	54K	224K	.913M
CROMEMCO	SISTEMA 0	5,500	Z80A	8	S-100	64K	54K	.70ME
CROMEMCO	SISTEMA 2M	9,995	Z80A	8	S-100	64K	512K	.70ME
INTEL	OPI/50	7,000	8084	8	PROPIO	64K	64K	1ME
INTERFACE	SUPER DRAIN	2,500	Z80	8	S-100	32K	64K	.35ME
HEATHKIT	WH89-6A	2,895	Z80	8	PROPIO	48K	64K	.1ME
ARCHIVES	ARCHIVES	4,800	Z80	8	PROPIO	64K	64K	.7ME
TELEVIDEO	SYSTEM 1	3,995	Z80	8	PROPIO	64K	64K	1ME
CODEX	CSX48/1	3,975	6800	8	PROPIO	32K	64K	1ME
TEXAS								
INSTRUMENT	DS990/1	9,995	TMS9900	14	PROPIO	32K	32K	NO
HENLET								
PACKARD	05	3,700	HP	8	GPFB	16K	32K	NO
TEKTRONIX	4051	6,900	TEK	8	GPFB	16K	64K	

nota:

*Los precios estan dados en U.S. Dollar

*Todos estos sistemas incluyen el precio de terminal.

* Datos de enero de 1981.

DATOS DE SOFTWARE

Marca	Modelo	Sist. Mono U.	Operat. Multi U.	Leng. de Progr.					
				A	B	C	F	P	
Tandy	TRS 80 Mod II	TRS2DOS	no	si	si	op	op	no	
Apple	Apple II	---	no	si	si	op	op	si	
Atari	800	---	no	si	si	no	no	no	
Altos	ACS- 8000-S	CP/M	MP/M	si	op	op	op	op	
ADDS	Multi- vision I	Muon	Muon	si	si	op	op	si	
Vector	VIP	CP/M	no	si	si	op	op	op	
Cromemco	2 ó 3	DDDS	Cromix	si	si	si	si	op	
Intel	Op 1	Dos	si	si	si	op	op		
Intertec	Super Brain	CP/M	no	si	si	si	op	op	
Heath	Wh 89	op	no	op	op	op	op	op	
Archives	archi- ves	CP/M	no	si	si	op	op	op	
Televideo	System I	CP/M	Most	si	si	si	op	op	
Codex	CDX	si	no	si	si	si	op	op	
Texas									
Instrument	DMS990-1	TMS	no	si	si	no	no	no	
Hewlett									
Packard	85	Hp	no	si	si	no	no	no	
Tektronix	4051	Tek		si	si	no	no	no	

donde:

- A = Assembler
- B = Basic
- C = Cobol
- F = Fortran
- P = Pascal
- op = opcional (no lo soporta el fabricante)
- no = no lo tiene
- si = sí lo tiene (lo soporta el fabricante)

nota:

* Datos de enero de 1981

Consideraciones Particulares.

Las consideraciones particulares que se tomaron en cuenta para el estudio tienen que ver con la aplicación de las microcomputadoras en otras áreas como son la administrativa, docencia e investigación. Las cuales tienen necesidades de cómputo un tanto distintas y de todas estas se tiene que escoger el equipo que mejor cumpla con dichas necesidades, por esta razón se incluyó a continuación las consideraciones a tomar en cuenta para la compra del equipo.

-Apoyo administrativo

Dentro de esta área tenemos :

- Substitución de equipos de perforación.
- Apoyo a proceso de control administrativo tales como:

- a)Inscripciones
- b)Control de personal
- c)Presupuesto
- d)Inventario

- Apoyo al desarrollo de nuevos sistemas.
- Procesamiento de la palabra

A continuación se explicarán cada una de las necesidades:

- Substitución de equipos de perforación.

En este caso se requiere un equipo que permita el almacenamiento de información. Con tener el equivalente a la capacidad de 1,000 tarjetas puede ser bueno. Se requiere un teclado simple para captar esta información y una vez captada, debe haber forma de que pueda ser transferida la información al equipo Burroughs. Para esto se requiere del software necesario para que simule a una perforadora de tarjetas. En algunos casos, será deseable el poder usar varias terminales (hasta 4), ya que este proceso no consume grandes recursos de procesador y el mismo equipo pudiera hacerlo.

- Apoyo a procesos de control administrativo.

Para estos casos se debe tener la posibilidad de almacenamiento local, de uno o varios archivos que puedan estar en floppy disk, y de preferencia contar con un sistema de bases de datos.

En algunos casos la cantidad de información se incrementa grandemente, creándose grandes bancos de datos con información que puedan ser materias, grupos, alumnos, contabilidad, presupuesto y almacén. En estos casos se requiere una mayor capacidad y la posibilidad de acceso simultáneo a través de más terminales.

El sistema a considerar será uno que permita el almacenamiento de un volumen suficiente de información, (10

MB) y la posibilidad de uno a cuatro usuarios simultáneos. Hay que considerar que para algunos sistemas, un equipo del tipo de sustitución de tarjetas perforadas en donde hay poco volumen de información puede ser conveniente. En todos estos casos es muy importante la comunicación con el equipo Burroughs para la integración al banco de datos generales de cada sistema.

- Apoyo al desarrollo de nuevos sistemas.

En este caso, se requiere de un equipo que permita localmente la edición de programas, el poder probar aquellas rutinas ya programadas, poder copiar programas y comunicarse a la Burroughs; por lo que un sistema con una capacidad local de almacenamiento de 500 K puede ser satisfactorio.

En un sistema de este tipo, podría ser deseable que permitiera mayor almacenamiento, además de la posibilidad de que varios usuarios lo compartan simultáneamente.

- Procesamiento de palabra.

Este caso es similar al caso anterior, pero se requiere la posibilidad de tener una impresora de preferencia del tipo Daisy-Wheel, (Sistema intercambiable de tiros de letras, para lograr calidades en la impresión como las máquinas de escribir) por lo que un sistema multiusuario en el que haya colateralmente una impresora compartida, es el más indicado.

- Investigación

En este caso requerimos un sistema del tipo de apoyo al desarrollo de nuevos sistemas, que permita la creación de programas, la compilación y prueba en forma local, además de

la protección de archivos en forma local, en unidades de floppy disk, así como el procesamiento de palabras.

También es deseable el poder contar con tarjetas de electrónica tales como:

- Convertidores Analógico/Digital y Digital/Analógico
- Controladores
- Interfases Serie/Paralelo y Paralelo/Serie

- Docencia.

Para el caso de la docencia, se considera equipo para la sustitución de perforadoras ya mencionadas y la conveniencia de equipos de un solo usuario, de tal manera que el alumno tenga a su disposición un equipo completo para aprender el uso de la máquina y realizar su programación sin la necesidad del uso de grandes volúmenes de información.

A continuación se presentan tablas comparativas para cada una de las aplicaciones antes mencionadas.

Apoyo a procesos de control Administrativo.

Sistema	1 Usuario			4 Usuarios			
	Memoria	Disco	Precio	Mem.	Disco	Cant.	Precio
Altos	308 k	10 MB	12,000	208K	10 MB	1	15,500
ADES	34 k	10 MB	8,595	252K	10 MB	1	16,085
Vector	31 k	10 MB	8,659	252K	10 MB	1	
Chromem	64 k	10 MB	11,800	252K	10 MB	1	20,040
Contel	24 k	10 MB	10,800	252K	10 MB	1	32,000
Tele video	64 k	10 MB	8,975	320K	10 MB	1	14,380

* Datos de enero de 1991

Substitución de perforación, apoyo al desarrollo de nuevos sistemas e investigación.

Sistema	1 Usuario			4 Usuarios			Precio
	Memoria	Disco	Precio	Mem.	Disco	Cant.	
Radio-							
Check	64 k	500KB	3.500	256k	2 MB	4	14.000
Apple	48 k	500KB	3.000	192k	2 MB	4	12.000
Alfas	208 k	1 MB	7.000	208k	2 MB	1	10.500
ADDS	64 k	700KB	4.100	256k	700K	1	11.200
Vector	64 k	1 MB	3.695	256k	1 MB	1	
Cro-							
menco	64 k	1 MB	3.200	256k	1 MB	1	14.700
*Ortel	64 k	1 MB	7.800	256k	1 MB	1	27.000
Surar							
Brain	64 k	700KB	2.500	256k	2.8MB	4	10.000
Heath							
Kit	48 k	400KB	2.895	192k	1.6MB	4	20.000
Archi-							
ver	64 k	700KB	4.800	256k	2.8MB	4	19.200
*Codex	64 k	1 MB	9.700	256k	2 MB	4	34.000
Tele-							
video	64 k	2 MB	3.975	520k	11 MB	1	14.390
*HP	64 k	Cinta	3.700	64k	.8MB	4	15.000
*TI	32 k	Cinta	4.795	256k	2 MB	4	20.000
*Tek-							
tronix	16 k	Cinta	6.900	64k	1 MB	4	28.000

notas:

- * No se aplica a substitución de perforación.
- * Precios en U.S. Dollar
- * Datos de enero de 1981 (Referencia 3.1)

Procesamiento de palabra
 Nota: Aproxar a cada sistema de 2,000 a 4,000 U.S. por la Impresora.

Sistema	1 Usuario			4 Usuarios		
	Memoria	Disco	Precio	Memoria	Disco	Precio
Radio Shack	64 k	446 k	3,899	NO	NO	NO
Apple	48 k	446 k	3,000	NO	NO	NO
Altos	208 k	1 MB	5,990	208 k	2 MB	10,500
ADDS	64 k	1.2 MB	3,875	256 k	700 k	11,200
Vector	64 k	315 k	3,495	256 k	1 MB	
Cromemco	64 k	1.2MB	5,500	256 k	1 MB	14,700
Ontel	64 k	1 MB	7,800	256 k	1 MB	27,000
Sleer						
Brain	64 k	700 k	2,500	NO	NO	NO
Heath-Kit	48 k	400 k	2,895	NO	NO	NO
Archives	64 k	700 k	4,300	NO	NO	NO
Codex	64 k	1.2MB	2,895	320 k	11 MB	14,380
Tele-video	64 k	2 MB	3,975	320 k	11 MB	14,380

notas:
 *Precios en U.S. Dollar
 *Datos de enero de 1981

Docencia

Sistema	Memoria	Disco	Precio
Radio Shack	64 k	446 kb	3,889
Apple	48 k	446 kb	3,000
Altos	208 k	510 kb	5,990
ADDS	64 k	1.2 mb	4,100
Vector	64 k	315 kb	3,495
Cromemco	64 k	760 kb	5,500
Sleer Brain	64 k	700 kb	2,500
Heathkit	48 k	100 kb	5,000
Archives	64 k	700 kb	4,300
Codex	64 k	1.2 mb	3,975
Televideo	64 k	1.2 mb	3,995

notas:
 * Los precios son en U.S. Dollar
 * Datos de enero de 1981

A continuación se muestra una lista con las pruebas que se ejecutaron en cada uno de los equipos bajo estudio.

1.- Funcionamiento del sistema operativo.

- a) Creación de archivos en disco.
- b) Edición de archivos.
- c) Verificación del contenido de los archivos creados.
- d) Ejecución de programas.
- e) Realizar archivos de comandos ejecutables.

2.- Uso del equipo.

- a) Como terminal.
 - i) Velocidad de transferencia.
 - ii) Bits de stop y paridad (en caso asíncrono).
 - iii) Modo de transmisión (Half duplex, Full duplex, síncrona, asíncrona).
 - iv) Número de puertos.
 - v) Compatibilidad con la interface RS-232C.
- b) Como terminal inteligente.
 - i) Transferencia de archivos de floppy a la computadora central.
 - ii) Transferencia y almacenamiento de archivos de la computadora central al floppy.
 - iii) Programa terminal.
- c) Facilidades.
 - i) Teclas programables.
 - ii) Implementación del BREAK.

Conclusiones.

Hay que considerar que las necesidades de la UNAM en esta área pueden ser para fines prácticos casi ilimitadas, por el gran volumen de aplicaciones potenciales; esto lo demuestran estadísticas de uso de los equipos con que cuenta el C.S.C. Considerando que sólo una parte muy pequeña de la comunidad Universitaria está haciendo uso de estos recursos.

Para el caso de captura de datos, la Universidad cuenta con gran cantidad de equipos de perforación, la conveniencia de cambiarlos por microcomputadoras resulta evidente, debido a la relación de los costos actuales y por la capacidad de

almacenamiento en floppy disk. Como otra característica, podemos añadir que el floppy es re-utilizable, reduciendo con esto notablemente los costos en comparación con el sistema de tarjetas tradicional.

El caso de procesamiento de palabra es una aplicación potencial muy amplia, ya que en cada dependencia universitaria podría existir un equipo para llevar a cabo dicha aplicación.

En caso de apoyo a procesos administrativos, investigación y desarrollo de nuevos sistemas, la escala es menor, ya que se reduce primordialmente al crecimiento requerido de algunos usuarios del Centro de Cómputo, como nuevas dependencias que actualmente se encuentran en el proceso de inicio de automatización.

Con estas consideraciones se requiere que el equipo que sea introducido en la UNAM, no sea un equipo experimental. Además se requiere que sea un equipo que lleve algún tiempo en el mercado, demostrando su confiabilidad, calidad y desarrollo en software, para que la introducción sea exitosa y no cause problemas en un principio. Deberá ser un equipo que no requiera de expertos para manejarlo y que se ajuste en mayor grado a las condiciones generales mencionadas en los párrafos anteriores.

De la lista de equipos considerada y de la experiencia obtenida en las pruebas de estos equipos resaltan los siguientes:

ADDS	Radio Shack Mod. II.
Altos	Intel
Cromemco	Televideo

A continuación se ven cuales son los equipos probables en cada aplicación en particular .

RESUMEN DE EQUIPOS RECOMENDADOS Y EN DONDE SE APLICAN

Sistema	Aprovo- substi- tución Invești- sación	Docen- cia	Admi- nistra- ción	Procesa- miento de Palabra	# de Aplica- ciones
ADDS	Si	No	No	No	1
Tele- video	Si	Si	Si	Si	4
Altos	Si	No	Si	Si	3
Cre- memco	Si	Si	Si	Si	4
Radio- Shack	Si	Si	No	No	2
Ontel	No	No	Si	No	1

ADDS.- Fue probado y aunque en un principio resultó atractivo, después se encontraron pequeños problemas tales como: el software no funcionaba satisfactoriamente. En el caso del multiusuario, no funcionó adecuadamente. El software con que cuenta el sistema es muy limitado, pues solo es el editor. Además de no contar con el Bus S-100.

Ontel. Resulta ser el equipo más costoso. Es un equipo orientado principalmente al proceso administrativo y no cuenta con una gran cantidad de software; Tampoco posee el Bus S-100.

Televideo.- En este caso se trata de un nuevo equipo que acaba de salir al mercado. No cuenta con el Bus S-100. Y no se tiene gran experiencia en este equipo por lo nuevo.

Por lo que se puede concluir que quedan los siguientes equipos:

Altos.- Por su flexibilidad y bajo costo en multiusuario.

Cromenco.- Por su calidad, expandibilidad, software existente y arquitectura (utiliza el Bus S-100 y compatibilidad con CP/M).

Radio Shack.- Para aplicaciones de docencia y utilización con un usuario.

De las tres opciones anteriores se pueden destacar las siguientes características deseables:

- Sistema operativo compatible con CP/M.
- Posibilidad de expansión en forma modular.
- Posibilidad de multi-usuarios.
- Bus S-100.

Debido a que la única máquina que reúne todas estas características, y es una de las más difundidas, se seleccionó a CROMEMCO, como el microprocesador que adquirirá la UNAM para todas las aplicaciones antes mencionadas.

2.1.2 Análisis del Sistema.

El método utilizado para el análisis del sistema de captura fue sencilla, como se mencionó en la introducción del capítulo, ésta consto de dos fases:

a) Primeramente se tuvieron reuniones con personal del departamento de captura del CSC, para tener una definición clara de las características con que debía contar dicho sistema, debiendo realizar al menos las funciones con sus cuenta una perforadora de tarjetas.

b) Se realizó la evaluación de un sistema de captura de datos llamado "DATA-STAR" en los siguientes dos aspectos:

i) Facilidades de la captura y validación, definidos a través del lenguaje que utiliza el sistema.

ii) tiempo en que se realiza esta captura y validación.

Habiendo realizado esta prueba con personal del Centro de Servicios de Cómputo, se observaron varios problemas, entre los cuales se pueden mencionar los siguientes:

1.- El sistema DATA-STAR no permite datos constantes

Esto implica que la información que se mantiene invariable en los documentos a captar, debe escribirse en cada uno de ellos; si consideramos los grandes volúmenes

datos que son procesados todos los días, esta desventaja implicaba un alto costo en la captura de información.

- 2.- No permite criterios de validación más allá de los numéricos y alfabéticos.

Debido a la gran cantidad de usuarios con que cuenta el CSC, se tienen muchos tipos de información a captar y cada una de ellos representa uno o varios casos específicos que las diferencian, por lo que si se optaba por este sistema, se habrían tenido que seguir realizando filtros de información, para que los archivos definitivos, almacenados en el sistema central, se mantuvieran consistentes.

- 3.- La transmisión de datos es muy lenta

Este es un de los aspectos mas importantes que se consideraron en el análisis del sistema en cuestión por lo siguiente: el personal que se encarga de realizar la captura de información, está acostumbrado a realizar la perforación de las tarjetas, tan velozmente como su experiencia y habilidad se lo permiten, se debe considerar, que la mayoría de ellos es gente que tiene muchos años de experiencia en este campo, por tanto su rapidez puede variar en promedio de doscientos a doscientos cincuenta golpes por minuto. Utilizando este sistema, se reduciría enormemente el número de caracteres capturados en el mismo tiempo.

Debido a estos factores y al elevado costo del sistema, se optó por crear uno propio, que permitiera hacer lo mismo que el DATA-STAR y además superara las deficiencias antes mencionadas (meta de esta tesis).

Presentación de Alternativas.

En base a las opiniones del departamento de captura de datos y a las propias, se obtuvieron tres alternativas, bajo las cuales podía ser realizado el sistema.

a) PROCESO POR " MASCARA "

La característica principal de esta forma, se basaba en la presentación de una cadena de 80 caracteres, definiendo al mismo tiempo el tipo de cada uno de ellos, con el objeto de realizar la captura y validación caracter por caracter.

b) PROCESO POR " CAMPO "

En esta forma, la característica principal es la definición e identificación de conjuntos de caracteres (a los que se le denominó CAMPO), dentro de un vector, proporcionándole el tipo de cada uno de los campos (alfabético, numérico, caracteres especiales, etc), así como los criterios de validación a considerar.

c) PROCESO POR FORMATO

Considerando las características del sistema "DATA-STAR" las cuales definen un campo, su identificación, los criterios de validación, así como el formato externo, entendiéndose por este último la forma en que el usuario define sus campos en la pantalla, para que sean desplegados en el momento en que se esta realizando la captura, no obstante que el formato en que queda almacenada la información en disco varía totalmente su orden.

Este tipo de proceso implica la definición de cada uno de los campos, la longitud de los mismos, el tipo de caracteres permisibles, los rangos de los valores, en caso en que el campo sea numérico, la posición en el archivo de salida, etc. Este formato identificado con un nombre quedaría almacenado en un archivo de formatos, para el momento en que se deseara utilizar nuevamente se podría hacer simplemente llamándolo.

De las tres alternativas, se escogió la tercera, puesto que permitiría la captura de cualquier tipo de formato y por su forma de presentación sería fácilmente accesibles a todos los usuarios además aumentaría la velocidad al momento de captar la información, así como al momento de verificarla.

Junto con esta alternativa, juega un papel importante la transferencia de la información captada, al equipo BURROUGHS, lo cual, permitiría agilización en el flujo de la información, desde el momento en que fuese recibida en el departamento de captura, hasta el momento en que estuviera disponible en el equipo central para el usuario.

Análisis de la Alternativa Escogida.

Como se mencionó en la sección anterior el sistema utilizaría la forma de "PROCESO POR FORMATO". De acuerdo a las necesidades detectadas, se optó por definir el análisis de este esquema en los siguientes pasos:

- a) Generación y almacenamiento del formato externo.
- b) Identificación de campos.
- c) Formato interno.
- d) Tipos de campos.
- e) Criterios de validación.
- f) Campos constantes.
- g) Verificación o validación de la información contra catálogos o directorios predefinidos.
- h) Relación entre variables.
- i) Aceptación de algoritmos para verificación de campos determinados.
- j) Acceso a cualquier registro por alguna llave o por el número de registro dentro del archivo.
- k) Definición de subcampos.
- l) Opción para verificación.

A continuación se explican cada uno de ellos:

- a) Generación y almacenamiento del formato externo.

Para este punto el sistema debe permitir la facilidad de definirle los campos distribuidos en la pantalla a su elección, con el objeto de permitir una visualización de la información sin amontonamientos o formaciones que para uno puedan ser fáciles de leer, mientras que para otros no lo son, además esto implica que las capturas fijen menos la vista, por tanto su trabajo resulta menos cansado.

Por las condiciones antes expuestas, aparecerá solo un registro a la vez en la pantalla, cada vez que haya sido terminado, el sistema eliminará la información que necesita ser eliminada, y dejará el registro como si fuese el primero.

Este cambio necesita, realizarse en el menor tiempo posible (un segundo máximo), puesto que de lo contrario resultaría contraproducente.

b) Identificación de Campos

Para este tipo de formas, se hace necesario que se reconozca a cada uno de los campos con un identificador, que será almacenado en el formato generado (interno) y corresponderá al campo definido en el formato externo. Esta identificación aparecerá en la pantalla junto a cada uno de los campos que se están captando.

c) Formato del Registro.

El formato del registro o formato interno estará determinado por la longitud máxima de cada uno de los campos, mantendrá el orden que el usuario determine y en su caso será el formato interno que mantendrá cuando sea enviado al sistema central.

d) Tipo de Campo.

Considerando la tabla de caracteres que son posibles manejar en un sistema de cómputo, se optó por el conjunto de caracteres ASCII puesto que el sistema cronemco utiliza este código, y la Burroughs identifica tanto a éste como a otros, por tanto no habría problema en conversión de códigos, entonces los tipos posibles de campo serán:

A	=	Alfabético
N	=	Numérico
AN	=	Alfanumérico
AB	=	Alfabético o blanco
NB	=	Numérico o blanco
E	=	Carácter especial
B	=	Blanco

e) Criterios de Validación.

Para que un campo pueda ser verificado, el usuario debe proporcionar las características de cada campo, a través de un lenguaje sencillo de interacción. Para ello es posible determinar criterios de validación especiales, que por las características y uso común de algún campo sea todioso generar su criterio para cada aplicación, es decir estos estarían almacenados en el sistema de tal forma que con una simple instrucción a él, actuaría automáticamente, un ejemplo sencillo de ello es el muy usado registro federal de causantes, el cual sabemos que esta formado de cuatro letras y seis dígitos, los cuales llevan siempre un orden y lógicamente no pueden exceder ciertos límites.

f) Campo Constante.

Como se mencionó en su oportunidad, este es un caso muy utilizado en la actividad de la captura de datos, que si no se considerara, llevaría mucho tiempo en estarlo captando, por tanto el sistema deberá considerar esta alternativa, de tal forma que le sea posible asignarle un valor específico a cualquier campo durante uno, dos o más registros, y que permita modificarse en cualquier momento, ya sea su valor actual o que ese campo deje de ser constante, es importante el hecho de que más de un campo pueda ser constante a la vez, desde luego con diferentes valores.

g) Validación contra un Catálogo.

La idea de llevar a cabo este tipo de validación, resulta muy sencilla, puesto que solo implica comparar uno o mas campos contra un archivo de claves almacenado tambien en el sistema, sin embargo, esta opción puede llegar a ser un poco lenta si los catálogos son muy grandes, (no olvidar que uno de los aspectos mas importantes en la buena consecución de este sistema es el tiempo de respuesta en un rango menor a

un segundo como máximo en lo que respecta a el paso del cursor de uno a otro campo así como en el almacenamiento del registro una vez que fue captado respectivamente) por tanto, se considera difícil garantizar un buen funcionamiento en línea de esta forma de validación.

h) Relación entre Variables.

Este también se ha considerado un aspecto importante en la validación de la información y se refiere a la comparación de dos o más campos , los cuales deban coincidir.

i) Acertación de Algoritmos para la Verificación de Campos Determinados.

Se refiere a la validación de la información en lo que respecta a los denominados dígitos verificadores, esto es , es posible proporcionar algoritmos tales como la verificación de números de cuenta, nombres, etc que hagan consistente la información que se pretende almacenar.

j) Acceso a Cualquier registro por alguna llave en Especial o por el número de registro dentro del archive.

Esta forma, permitiría acceder cualquier registro, al cual, por alguna causa el usuario desée regresar a él, ya sea para corregir información o para su propia consulta. Este punto es muy utilizado en el proceso de captura, sobre todo por la forma en que este se realiza. Por ello se hace necesaria su inclusión como parte del sistema.

k) Definición de Subcampos.

Esta opción debe permitir la división de un campo en varias partes, con el objeto de facilitar su captura y validación simultánea.

1) Opción para Verificación.

Este aspecto es muy importante sobre todo para aumentar la velocidad de respuesta del sistema, puesto que como es sabido existe información que NO puede ser verificada o mejor dicho no existen criterios de validación para ella, un caso muy simple es el de los textos que utilizan los sistemas de información, aquí es muy difícil establecer un criterio para determinar que cierta palabra es o no correcta. Pensemos en el caso en que se está definiendo una serie de campos para texto, y a los cuales se les quiere validar. De acuerdo a las opciones establecidas en el "TIPO DE CAMPO", estas zonas del registro serían Alfanuméricas ó Blanco, Numéricas ó Blanco y con Caracteres Especiales. Hasta aquí podría no existir problema y si se recibiera un texto como el siguiente: "DIRECCION: PASEO DE LAS PALMAS 357-23, CODIGO POSTAL 02345 NOMBRE: NUÑEZ ACUNA JAVIER", podría ser correcto, ahora, si se encontrara un campo como el siguiente: "(78799,X6E==124n:!:)", bajo que criterio se podría determinar que es o no correcto?. De acuerdo a los tipos de campo que el usuario determino para el texto, este es correcto, sin embargo, para cualquier persona que lo estuviera leyendo no tendría ningún sentido, sin embargo al sistema al estar verificando zonas como estas, en realidad solo estaría trabajando de mas. Es por esto que una opción de este tipo optimiza la utilización del sistema y aumenta su tiempo de respuesta.

2.2 Alternativas de Comunicación entre los equipos.

Dentro de esta parte se dara una exposicion de las alternativas que existen para la comunicacion entre los equipos. Para establecer la comunicacion entre el microcomputador CROMENCO y el computador central BURROUSHC es necesario que exista un protocolo de comunicacion. Este debe ser implementado tanto en el equipo Cromenco como en el equipo Burroughs, dicha implementacion en el primer equipo y en la primera alternativa fue realizada en el C.S.C. como tema de la tesis (Enlace entre una Microcomputadora Cromenco y una Computadora Burroughs) y la programacion en el equipo Burroughs fue realizada como parte de la presente tesis. El hecho de que se haya elegido esta primer alternativa para la implantacion del sistema es la facilidad de ella y el tiempo en la cual se programa dicha opcion, esto fue debido precisamente a restricciones en cuanto a tiempo.

Para el desarrollo de los sistemas de comunicacion entre los equipos se plantean las siguientes alternativas:

i) La primera alternativa de comunicacion y la mas sencilla es la de realizar la programacion tanto en el microcomputador como en el Computador Central (C.C.), que es

enlazarán mediante un protocolo muy sencillo, esto es cuando se quisiera transmitir o recibir algún archivo, se ejecutará un programa en la microcomputadora el cual a su vez ejecutará un programa en el C.C. y este se encargará de los archivos transmitidos o recibidos en el C.C. y el primero de los archivos contenidos en los diskettes del microcomputador. Las acciones que deberán ejecutar dichos programas cuando se transmitiera o recibiera un archivo son las siguientes: En la recepción de un archivo primero se tendrá que verificar que no exista un archivo con el mismo nombre en el C.C., que la clave bajo la cual se va a grabar el archivo sea válida, investigar el nombre del pack que tiene asociado la clave y si se cumple con los requisitos comenzar a grabar la información en el C.C.. Si se requiere transmitir un archivo hay que chequear que la clave del usuario sea válida y que exista el archivo en el C.C. y lo contrario en el Microcomputador.

Dentro de esta opción no existe verificación adecuada de la transmisión de mensajes puesto que el protocolo que se implementa aquí, es solo a nivel de establecer la comunicación entre los equipos, por medio de señales de reconocimiento (entre los programas de usuario), pero no se tiene la seguridad en cuanto a la integridad de cada mensaje transmitido. Cabe señalar aquí que el protocolo del que se habla es a nivel alto, esto es de comunicación de procesos o programas (la descripción de los niveles está en el capítulo IV). Este enlace y su protocolo se describe a continuación, en términos generales:

TRANSMISION

CROMEMCO

----->

BURROUGHS

Se ejecuta el programa en Cromemco para transmitir el cual a su vez ejecuta el programa en el C.C. y espera a que este le envíe con una señal de reconocimiento.

Es ejecutado el programa en el C.C. el cual como primera acción manda una señal hacia el microcomputador, indicándole que ya está listo.

El Micro recibe la señal del C.C. de que ya está listo, y le envía la extensión del archivo a transmitir, para que el C.C. decida que acción ejecutar.

Se envía una señal de reconocimiento para indicar al Micro que se recibe un mensaje (la extensión del archivo).

El Micro habilita el puerto de comunicación para que el programa en el C.C. pueda actuar interactivamente con el usuario, de acuerdo a los requerimientos de este.

Se realizan una serie de preguntas al usuario, correspondientes a la transmisión del archivo. Al finalizar estas se envía al Micro una señal para indicar a este que ya puede comenzar a enviar los registros del archivo en cuestión.

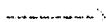
El micro recibe la señal del C.C. y envía el primer registro del archivo y espera a recibir una señal del C.C. de que recibió el registro.

Se recibe el registro y se graba en disco y se envía una señal al Micro indicándole que prosiga con el envío de registros.

Se recibe el mensaje, se envía el siguiente registro.

RECEPCION

BURROUGHS



CROMEMCO

Para la recepcion primero el microcomputador presenta cual es el nombre del archivo en el disquete de esta y a continuacion ya que se checa que no existiera un archivo con el mismo nombre, se abre el programa en Burroughs habilitandolo con la pantalla.

El programa en el equipo Burroughs presenta por el nombre del archivo que se desea transmitir, y ya que se checa que exista dicho archivo se envia una señal al microcomputador indicandole que a continuacion se le va a enviar el primer registro.

El microcomputador recibe la señal y se prepara a recibir el registro, una vez que lo ha recibido envia una señal de reconocimiento hacia el C.C. indicandole, ya que que se recibio bien el registro y se abre el siguiente registro o una señal indicandole que no se recibio correctamente el registro y que se envia una retrotransmision del mismo registro. Si ocurre que ya no hay memoria suficiente en el micro, esta puede retrasar la transmision del siguiente registro para poder traer el contenido de la memoria hacia el disquete.

Una vez que se han enviado todos los registros del archivo al C.C. envia una señal al micro indicandole que ya se termino la transmision.

El micro envia la señal al C.C. para cerrar el archivo de terminal y concluir el proceso.

En esta opción de comunicación como podrá verse lo que está ocurriendo es que la microcomputadora ejecuta un programa en su memoria y entabla una conversación con el usuario para enterarse de que es lo que este quiere realizar una vez obtenidos datos necesarios comienza a entablar la comunicación con el C.C. mandando un control por el puerto al cual está conectado, y espera a que ésta le conteste con el nombre de la estación después el programa simulara por así decirle una terminal manejada por un usuario (en este caso el usuario es el programa en el microcomputador) el cual lo que hará es dar la clave de entrada al sistema Burroughs y ejecutar un programa en este equipo ya sea para recibir o enviar archivos. En este sentido los dos programas tanto el del Microcomputador como el del Computador Central realizan las mismas funciones que son acceder el directorio para chequear que el archivo solicitado exista en caso de transmisión o que no exista en caso de recepción. Al ejecutarse el programa en el equipo Burroughs se utilizará el protocolo antes descrito. De esta forma lo que ocurre es que el programa en el Microcomputador recibe por un puerto lo que tiene en la línea de comunicación y lo pone en memoria y en la pantalla.

En este nivel de protocolo también se puede implementar un control sobre la información, es decir, el transferir caracteres de paridad, dígitos de control, pedir retransmisión en el caso de error, indicar recepción satisfactoria, etcétera.

Esta opción de comunicación como ya se expuso no es la más óptima, pero sí la más inmediata, por lo cual fué la que se implementó para que posteriormente se realizara una versión óptima del enlace de los equipos, la cual es

describiera en la tercera opción de enlace.

ii) La segunda opción es la de utilizar un protocolo de línea ya implementado en la red de comunicación del C.C., este es el protocolo RJE (B771) el cual sí tiene una verificación sobre los mensajes transmitidos en la línea. Para implementar esta opción habría que realizar los siguientes pasos:

a) Implementar el protocolo RJE (protocolo B771) en la red de comunicaciones de la E-6700. Aunque este protocolo ya existe, es decir ya está programado hay que declararlo en la red y definirle las terminales que van a funcionar con este protocolo.

b) Realizar las adiciones necesarias al programa que se ejecuta en el Microcomputador, para que pueda comunicarse con el protocolo RJE que es más sofisticado que el que se tenía en la opción anterior, esto es tiene que manejar los casos en los cuales no fue recibido bien un mensaje y hay que retransmitirlo hasta que se reciba correctamente, etcétera.

c) Desarrollar los programas en el C.C. para que estos puedan recibir y/o transmitir los archivos. Con la implementación de este protocolo en el microcomputador es posible tener varias líneas lógicas en una sola línea física.

iii) La tercera opción es la de desarrollar un "MCS" (MESSAGE CONTROL SYSTEM) dedicado a la transmisión y recepción de archivos y desarrollar en paralelo la modificación (Protocolo B771) ó desarrollo de un protocolo en la red de comunicación de datos para que exista una seguridad en cuanto a la integridad de la información transmitida, esta opción es la más optima. Puesto que se tendría un solo

trabajo "Task" ejecutándose en el computador central, y esta MCS se encargaría de la transmisión y recepción de archivos, también tendría control sobre las Microcomputadoras que se quisieran enlazar, podría llevar una estadística del uso de las líneas de comunicación, etcétera. La forma en como se desarrolla la comunicación de datos en el equipo Burroughs está descrita en el tercer capítulo, por lo tanto aquí solo se explicara lo que deberá realizar dicho sistema controlador de mensajes.

Las funciones que deberá realizar el "MCS" propuesto son como ya se dijo tener el control de las líneas de comunicación de las microcomputadoras que requieran transferir o recibir archivos, verificar que existan o que no existan los archivos especificados según sea el caso, verificar que la clave del usuario sea válida y lo más importante en esta fase es tener la lógica necesaria y la estructura de datos para manejar varias terminales a la vez (cada una de estas puede estar mandando o recibiendo datos) y manejar todas las situaciones de error que se presenten en la transmisión y la iniciación de nuevas terminales.

2.3 Generación de Bancos de Información en el equipo Central.

Como se vio en la parte anterior de las opciones de comunicación de los equipos en la tercer alternativa se explicó someramente las funciones que debería tener el MCS para controlar las terminales y recibir y enviar mensajes de y hacia el sistema, aquí se describirá lo funciones que deberá tener dicho MCS pero desde el punto de vista de manejo de archivos, para la generación y actualización de estos. Estas funciones están ligadas estrechamente con el sistema así también deberá estar ejecutandose en el microcomputador.

Una parte importante es al menos poder generar un archivo nuevo en el C.C. con la información transmitida por el microcomputador, pero también deberá de tener la opción de adresar información a archivos ya existentes chequeando compatibilidades de estos. Otra función que deberá desarrollar es la de poder transmitir o recibir porciones de un archivo dado, especificando el número de registros y desde cual registro, esto es, si nosotros queremos editar un archivo del equipo Burroughs en el microcomputador, pero esto es demasiado grande y además solo nos interesan ciertos registros, podremos solo transmitir al microcomputador dichos registros y al término de su edición ó corrección en el microcomputador podremos regresar estos a su sitio de donde fueron extraídos. Esta opción sencilla ya nos da una gran

capacidad de desahogo del sistema central y ya habremos cumplido con nuestro cometido que es el de poder generar y actualizar archivos en el C.C. con un uso limitado de sus recursos.

La siguiente función a desarrollar por el MCS es la de tener la capacidad de poder transferir archivos entre los microcomputadores utilizando a la Burroughs como canal de acceso. Esto si se ve con detenimiento resulta sencillo con el desarrollo anterior pues lo que hará el MCS es comunicarse con la micro que desea copiar un archivo para identificar de que otra microcomputadora proviene el archivo y su nombre para que a continuación estable la comunicación con la línea a la cual está la microcomputadora y ejecutar el programa que abra y envíe el archivo especificado.

En resumen las funciones que deberá realizar el MCS en esta parte son:

- Leer un archivo de el C.C. y enviarlo por la línea a el microcomputador.
- Recibir un archivo de la línea y crearlo en el C.C.
- Leer parte de un archivo del C.C. y enviarlo por la línea.
- Recibir parte de un archivo por la línea e insertarlo en el archivo especificado.
- Enviar mensajes de operador a operador de las microcomputadoras.
- Ejecutar un programa del microcomputador para copiar el archivo hacia otra microcomputadora.
- Ejecutar un programa en el C.C.
- Ejecutar un Job (almacenado en el Micro) en el C.C.

CAPITULO III

3 Estudio de los Equipos

Dentro de este capítulo se expondrán las características de los equipos utilizados para la elaboración de la tesis, pero antes de comenzar con este estudio quisiera dar un breve esbozo de la evolución de las computadoras para enmarcar la relación de un microcomputador y un computador; a continuación se presenta esta breve síntesis de la evolución.

3.1 Desarrollo Tecnológico.

Uno de los avances más significativos, dentro del siglo XX es el desarrollo tecnológico que han tenido los circuitos eléctricos y electrónicos, desde la aparición del bulbo hasta los circuitos integrados a gran escala.

El desarrollo tecnológico de los componentes electrónicos comenzó en forma acelerada, con la aparición del transistor 1948, el que vino a sustituir al bulbo, mismo que tenía un gran consumo de potencia y disipaba mucha energía.

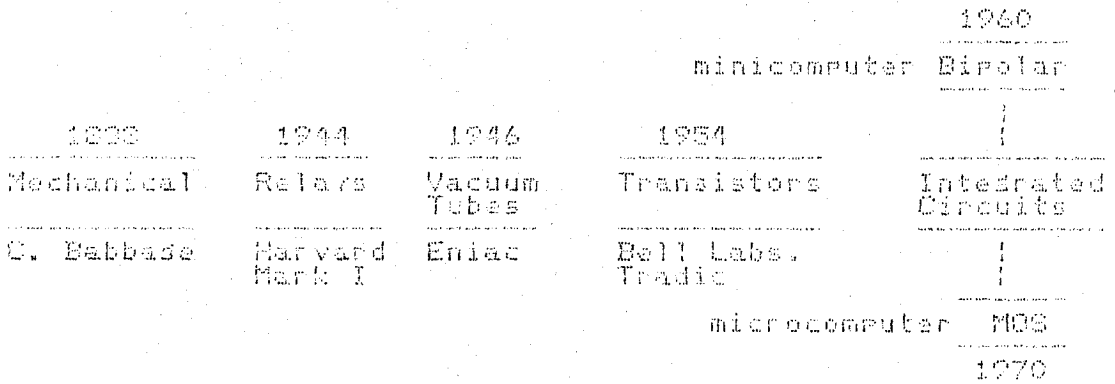
El siguiente paso fue el de unir varios componentes electrónicos en una sola tableta de silicio la cual era del tamaño de un transistor, y tenía de 80 a 100 elementos. De aquí en adelante la carrera entre las diferentes compañías de circuitos integrados ha sido el poder conjuntar más elementos electrónicos en un menor espacio, y el de reducir el producto velocidad-potencia por circuito.

Un circuito integrado es la combinación de diferentes elementos electrónicos fabricados en una sola pastilla o sustrato de silicio. El primer circuito integrado (C.I.) apareció al principio de los años 60'S. A partir de este momento se comenzó a investigar sobre distintas técnicas de integración de circuitos. Una de ellas muy utilizada es la tecnología MOS la cual fue utilizada primeramente en circuitos de memoria. El tamaño pequeño y bajo consumo de potencia de los circuitos MOS los hizo ideales para memorias, circuitos de comunicación, equipos militares y computadores. Para 1970 las calculadoras y las terminales fueron una aplicación importante para la tecnología MOS LSI, que es la que mas actividad ha tenido en cuanto al desarrollo de los microprocesadores, debido al alto grado de integración que permite (alrededor de 3000 transistores por chip).

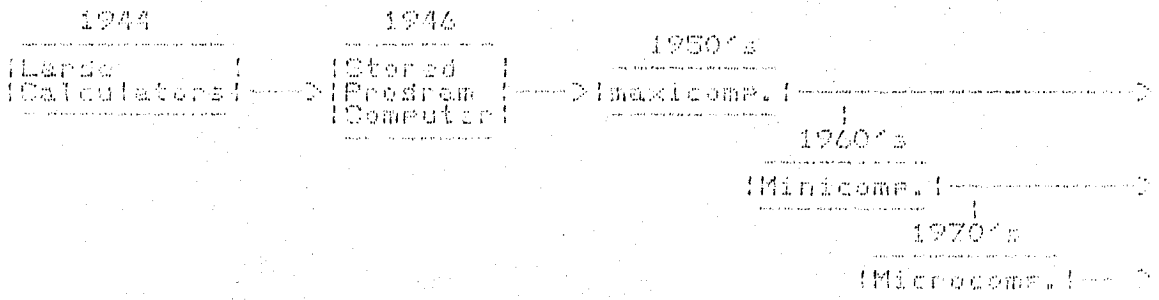
Evolución de las Computadoras.

Todos los avances tecnológicos antes vistos, han permitido a la industria de la computación evolucionar de una manera muy rápida y casi a la par que los avances de los diseños de los circuitos electrónicos. Los antecedentes históricos de la computación comienzan de una manera mas

formal cuando en 1900 J.A. Turing describe su máquina universal y en 1940 John Von Neuman describe su concepto de programa almacenado. De aquí en adelante y al paralelo con el desarrollo tecnológico comienza la evolución de la computadora. Aunque para ser más estrictos en el sentido de la historia habría que hablar de las máquinas de Blaz Pascal y Charles Babbage, pero dado que no es este el tema principal de la materia sus nombres solo se mostrarán brevemente a través del siguiente diagrama, la evolución tecnológica de las computadoras:



Por otro lado tenemos el diagrama de la evolución de las arquitecturas de las computadoras:



La evolución de los transistores hacia los circuitos integrados fue muy significativa, en el sentido que permitió un desarrollo más acelerado de nuevas arquitecturas de computadoras cada una de ellas ofreciendo mayores facilidades y eficiencia que su predecesor.

Como resultado directo de los circuitos integrados a gran escala tenemos el desarrollo de los microprocesadores los cuales tienen las mismas funciones que una computadora sólo que estos están en un solo chip (C.P.U. unidad de lógica y aritmética, registros de índice, contador de programas, etcétera).

Tal ha sido la evolución de estos microprocesadores que rápidamente han pasado de los laboratorios a la industria de control primeramente y después como auxiliar en los negocios pequeños y aun más como ayuda y/o diversión en la casa, a tal grado de evolución que hoy en día tienen las microcomputadoras la capacidad de procesamiento que tenía una computadora de los años 60's y con un costo realmente bajo.

El hecho de que las microcomputadoras hayan tenido en un inicio una repercusión más grande en campos fuera de los grandes centros de procesamiento de datos, como computadora personal de ayuda y/o diversión en la casa y en el negocio, que como una ayuda a los centros de procesamiento de datos, se debe principalmente al software de desarrollo con que contaban estas microcomputadoras en un principio, el cual no tenía la confiabilidad ni la flexibilidad que existía en el software en los equipos grandes, como es:

- Solo se escribía en lenguajes para el desarrollo como Fortran, Basic y Pascal.
- Rutinas muy primitivas para el manejo de archivos.
- Serpente de programas de aplicación limitados.
- Documentación limitada.
- Pocas fuentes secundarias de componentes.
- Baja velocidad de ejecución.
- Tamaño pequeño de palabra.
- Poca capacidad de almacenamiento.

Por estas razones era muy difícil que el serpente de un centro de procesamiento de datos incluyera entre su equipo este tipo de microcomputadoras, para confiarles el desarrollo de sistemas.

Conforme se fue desarrollando el software de los sistemas y avanzó la tecnología se fueron refinando las microcomputadoras y apareció cada vez más software de aplicación así como dispositivos de almacenamiento cada vez de mayor capacidad, además se comenzó a tratar de estandarizar el software desarrollado para micros, debido al número cada vez mayor de microcomputadores distintos que fueron apareciendo en el mercado.

Esta estandarización ha sido no solo en los lenguajes de desarrollo como Basic, Pascal, etc. si no que también en los sistemas operativos tal es el caso de CP/M y UNIX. Además de esta estandarización con el incremento de compañías que lanzan al mercado nuevas microcomputadoras se ha dado una competencia entre el software que maneja cada una de ellas, ofreciendo software más confiable, con más facilidades para el usuario y más sencillos en su utilización.

Una más de las razones, para la utilización de los microcomputadores es el hecho de la disminución del costo de los equipos, debido al constante avance en la tecnología, lo cual ha hecho que los equipos al término de un cierto tiempo se vuelvan obsoletos, aunque esto también es un problema al momento de decidir la compra de algún equipo en especial.

Las características básicas que definen las posibilidades y potencia de un microcomputador se podrían enumerar en los siguientes conceptos (estos de alguna forma condicionan la aplicación que pueda tener el microcomputador)

- Longitud de palabra (1,2,16,32 bits)
- Velocidad de ejecución de una instrucción (2,4,6 mpc)
- Capacidad de memoria directamente direccionable (4k, 16k, 64k)
- Capacidad de manejo de periféricos (interfases que maneja).
- Arquitectura.
- Flexibilidad de programación.
- Mecanismos disponibles para el diseño.
- Crecimiento modular.

Y además de las anteriores, las características que definen a un microprocesador son:

- Número de instrucciones.
- Número de acumuladores.
- Número de registros de trabajo.
- Formas de direccionamiento disponibles (directo, indirecto, indexado, etcétera).
- Capacidad de manejo de interrupciones.

Debido a todas las ventajas antes expuestas, que ofrecen los microcomputadores y del software de desarrollo en la actualidad se ha ido ampliando su utilización en diversos medios, en aplicaciones que van desde los sistemas para los negocios, hasta Bases de Datos complejas.

3.2 Equipo CROMEMCO.

3.2.1 Aspectos Generales del Sistema.

El sistema CROMEMCO cuenta con los siguientes elementos de hardware: una terminal modelo DM10, un módulo para las tarjetas del procesador, memoria y controlador de los diskettes, un módulo para dos controladores de diskettes < DDF >.

Sistema Operativo: CDOS (CROMEMCO Disk Operating System).

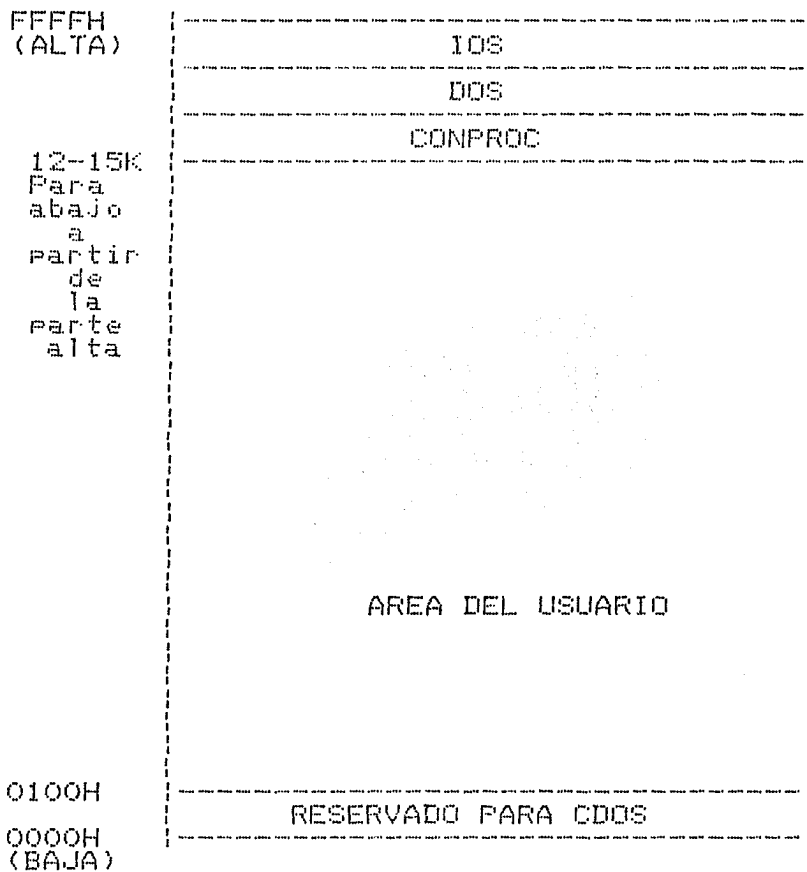
El Sistema Operativo de CROMEMCO (CDOS), es un producto diseñado y escrito originalmente en código de máquina de Z80 por la compañía CROMEMCO, para su propia línea de microcomputadoras. Sin embargo debido al gran número de programas que actualmente se pueden conseguir para correr bajo el sistema operativo CP/M, CDOS fue diseñado para ser compatible con CP/M. CROMEMCO tiene licencia de Digital Data Research para su utilización. Esto significa que la mayoría de los programas escritos para CP/M (de la versión 1.33 en adelante) correrán sin modificaciones bajo CDOS. Sin embargo, los programas escritos para CDOS no siempre funcionarán bajo CP/M. La función principal de CDOS es la de controlar la Entrada/Salida a ó de los dispositivos de almacenamiento masivo, tales como el disco duro ó el floppy. Además está diseñado para que los usuarios del sistema CROMEMCO puedan generar y manipular archivos del tipo secuencial y aleatorio utilizando nombres simbólicos.

Por otro lado CDOS contiene también las rutinas de manejo de la impresora, así como un gran número de llamadas para el manejo de pantalla, por medio de lo cual se puede hacer gráficas, manejar una línea de mensaje en la pantalla, etcétera.

Distribución de la Memoria.

La memoria de la microcomputadora bajo CDOS está dividida en dos grandes partes. La primera parte es una parte de memoria RAM que CDOS se reserva para él mismo. CDOS ocupa la memoria de la localidad 0000H a la 0100H (parte baja) así como también se apropia aproximadamente de 11K a 18K de la parte alta de la memoria RAM. La segunda parte es la memoria RAM del usuario. La memoria del usuario ocupa la memoria de la localidad 0100H a la parte alta de la misma hasta donde comienza la parte alta de CDOS. La longitud del área de memoria depende del sistema operativo que se esté utilizando, y éste a su vez del número de controladores de disco que se tengan declarados en el sistema operativo, si se utilizan las teclas de funciones para la terminal, etcétera, generalmente esta área es de 48K aproximadamente.

A continuación se muestra un mapa de la memoria con el sistema operativo.



Mapa de memoria para CDOS

Refiriéndonos al mapa de memoria, podemos ver que ésta se encuentra dividida en dos grandes partes, que son las siguientes:

PARTE ALTA DE MEMORIA.

CDOS tiene las rutinas básicas de Entrada/Salida para el manejo de periféricos (consola, impresora, perforadora, lectora de tarjetas y discos). Además contiene las rutinas de manejo de archivos, que son responsables de la creación, apertura, lectura y escritura de éstos en disco. También es

el encarsado de llamar a los programas del usuario y editar los comandos de la consola. Así mismo CDOS contiene algunas funciones internas llamadas comandos intrínsecos.

MEMORIA DEL USUARIO.

Es en esta parte en donde los programas del usuario son almacenados para su ejecución.

PARTE BAJA DE MEMORIA.

Esta parte de la memoria esta reservada por CDOS para las siguientes funciones:

0000H - 0002H	Vector del sistema Warm Start
0003H	Byte de E/S
0005H - 0007H	Vector de llamadas al sistema para presuntas del usuario
0008H	Especifica si se está corriendo bajo CDOS si se tiene FFH y bajo CROMIX si se tiene C3H
0030H - 0032H	Puntos de ruptura para DEBUG
0038H - 003AH	Salto al mensaje de: "INVALID JUMP"
0040H - 005BH	Reservada para el sistema
005CH - 007BH	File Control Bloks estandares para el usuario
0080H - 00FFH	Buffer estandar de Entrada/Salida al disco.

CDOS es cargado al área del sistema desde el disco por medio de la rutina de BOOTSTRAP.

El sistema operativo que se entrega al usuario puede ser cargado sin ningún problema ni modificación, para la configuración de equipo que comprende dos controladores de discos de 5 1/4 de pul., sin embargo si el usuario requiere de alguna configuración diferente de hardware (perifericos) se le debera de generar un nuevo sistema operativo, por medio de "CDOSGEN" (CROMEMCO Disk Operating System Generator) el cual nos permite definir la configuración que deseemos y generar un sistema operativo distinto.

Para poder hacer uso tanto de las utilerías, como de los paquetes debemos de estar bajo control del sistema operativo de CROMEMCO, lo cual se reconoce porque nos pone en la pantalla un "PROMPT" o señal de control que nos indica que podemos darle cualquier comando válido. Este control es una letra seguida de un punto, la letra nos indica en que controlador de disco estamos trabajando.

Manejo de Diskettes.

Los diskettes están compuestos de las siguientes partes:

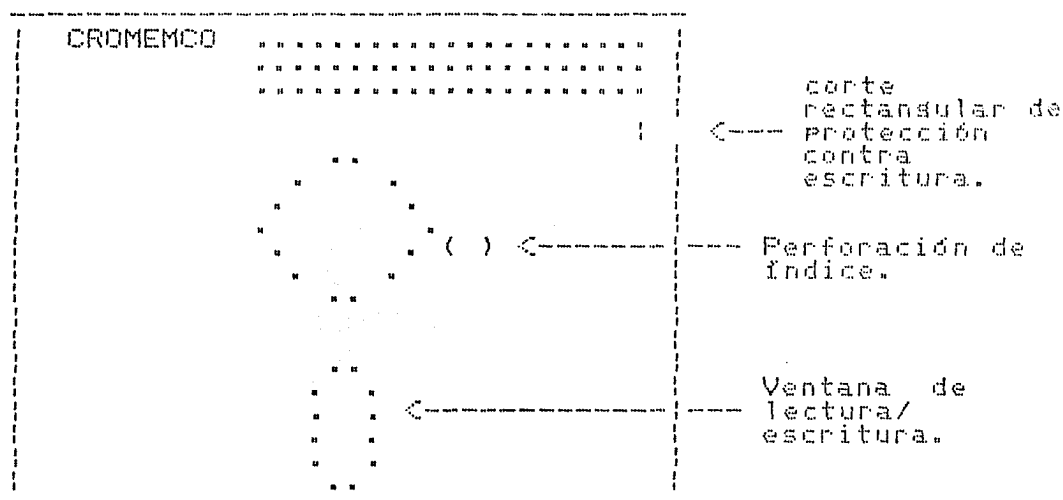
- Una etiqueta de identificación del diskette con su contenido.

- El corte rectangular a la derecha del diskette el cual permite habilitar o deshabilitar la escritura.

- Una porción del diskette en forma de elipse alarsada por la cual los controladores del diskette pueden leer o escribir en el disco.

- Un orificio circular al centro del diskette. Por medio del cual se presiona a éste y se hace girar.

- Una pequeña perforación circular de índice el cual sirve para sincronización del primer sector del disco además en los discos de 8 pul. indica si el diskette es de un solo lado o de doble lado.



Cuidado de los diskettes.

Existen una serie de precauciones que se deben tomar en cuenta para que no se dañen los diskettes y para que tengan un periodo de vida mas provechoso para el usuario; éstos cuidados se enumeran a continuación:

- Cuando un diskette no se utilice hay que guardarlo en su sobre protector y en su respectiva caja para evitar que se empolve.

- No doblar el diskette.

- No tocar la superficie interior del diskette ni tratar de limpiarlo, esto podría causar la pérdida de la información.

- No poner el diskette cerca de una fuente que genere campos magnéticos, tales como transformadores de potencia, solenoides, etc.

- No exponer el diskette a temperaturas extremas 10 grados C - 50 grados C, ni directamente a los rayos del sol.

- Se susiere que los diskettes no sean introducidos en los controladores, mientras se esta encendiendo o apagando el equipo, yá que bajo estas condiciones se puede llesar a grabar basura en aquellos. Si ocurre una falla de energía se deben de checar los diskettes cuando se reestablezca la energía eléctrica.

La forma correcta de insertar los diskettes en los controladores es la siguiente: con el filo mas cercano a la ventana elíptica hacia el frente y la etiqueta del diskette hacia arriba.

Además de las precauciones anteriores se deben de tomar en cuenta los siguientes aspectos:

- Mientras se esté ejecutando algún programa, no intercambiar los diskettes, a menos que aquel dé la opción de realizar ésto. El terminar la ejecución de un programa por medio del comando cntrl-c, discontinuara la ejecución de éste, pero sin cerrar los archivos que se estén utilizando dentro del programa.

- Ejecutar la utilería de "STATUS" de vez en cuando para verificar el directorio del diskette.

- Como una medida adicional de seguridad se recomienda tener respaldos de los diskettes, para en el caso eventual de que se llesue a dañar alguno, no se pierda completamente la información.

Protección contra escritura de los diskettes.

Los diskettes de 5.25" se pueden proteger de la escritura o se pueden dejar libres para ésta, de la siguiente manera:

Si se desea proteger el diskette, para que no pueda ser grabada información en él, se tiene que poner una etiqueta que cubra el corte rectangular de protección de éste.

Si se desea grabar en el diskette se tendrá que dejar libre este corte de protección.

También se pueden proteger los archivos del diskette, contra escritura, utilizando el intrínseco del sistema "ATTR", esta es una protección de software solamente.

Especificaciones de los diskettes.

Los diskettes que maneja el sistema "CROMEMCO System Zero" (CS/Z) son del tipo pequeño de 5 1/4 de pul. Las capacidades de almacenamiento de estos diskettes varían de acuerdo a sus características de grabación, a continuación se da una tabla con los diferentes tipos de diskettes:

<u>Disco</u>	<u>Tracks en el área del sistema</u>	<u>Area de archivos aproximada</u>
5" SS SD	3	81 K
5" DS SD	3	171 K
5" SS DD	2	188 K
5" DS DD	2	386 K

(SS=Single Sided, DS=Double Sided, SD=Single Density, DD=Double Density.)

Cada diskette está dividido en dos áreas: La primera es el área del sistema, la cual solo puede ser accesada por el usuario a través de la utilería "WRTSYS", La segunda área es la de archivos a la cual tiene libre acceso el usuario.

A continuación se da una tabla de las especificaciones de los diskettes:

<u>Disco</u>	<u>Cilindros</u>	<u>Superficies</u>	<u>Sectores/ Track</u>	<u>Tamaño del sector</u>
5" SD	40(0-27H)	2	18(1-12H)	128 BYTES
5" DD	40(0-27H)	2	10(1-0AH)	512 BYTES

Archivos.

Un nombre de archivo en CROMEMCO está formado de tres partes:

(X:) NDRMARCH (.EXT)

La primer parte es un identificador de "DRIVE", que nos sirve para especificar en que DRIVE "A" ó "B" esta colocado el diskette que contiene nuestro archivo, este identificador es opcional y consta de una letra (A ó B) seguido de dos puntos ' : '.

La segunda parte es el nombre del archivo (hasta 8 caracteres), el cual puede estar formado de letras , números y cualquier caracter ASCII excepto los siguientes:

\$ * ? = / . , : espacio

La tercera parte es la extensión de este nombre de archivo la cual tambien es opcional y consta de un punto y tres caracteres, esta extensión nos sirve para identificar distintos tipos de archivos, hay algunas extensiones que son estandares en el equipo CROMEMCO, éstas se describen a continuación:

- .BAK Archivo de respaldo del Editor
- .BAS Archivo fuente de Basic
- .CMD Archivo de comandos "Batch"
- .COB Archivo fuente de Cobol
- .COM Programa ejecutable de ensamblador
- .FOR Archivo fuente de fortran
- .HEX Archivo objeto con formato hexadecimal (8080)
- .LIS Listado de archivo fuente de Basic
- .PRN Archivo de listado o impresora
- .REL Módulo relocizable
- .SAV Archivo fuente de Basic
- .SYS Archivo imagen del sistema
- .TXT Archivo de texto de entrada para Formatter
- .Z80 Archivo fuente de ensamblador

Como ejemplo tenemos un archivo cuyo nombre es carta y que tiene una extensión de "TXT" y además está en el DRIVE "B", el nombre por el cual nos referiremos a este archivo es:

B: CARTA.TXT

Existen además tres formas de hacer referencia a uno o más archivos, lo cual es llamado referencia ambigua de archivos, que se lleva a cabo por medio de caracteres de reemplazo, estos tipos de referencia se describen a continuación:

i) <*> Asterisco. Por medio de este carácter podemos sustituir cualquier carácter(es) hacia la derecha de donde está el asterisco hasta el siguiente delimitador (punto ".", interrogación "?", Return "CR"). Por ejemplo si nosotros tenemos lo siguiente:

PROG*.*

Nos hará referencia a los siguientes archivos:

PROGRAMA.BAS
PROG1.INT
PROG2.Z80 PROG.FOR

ii) <?> Interrogación. Con este carácter podemos reemplazar cualquier carácter sobre la posición donde está el carácter "?". Como ejemplo tenemos:

?osa.txt

lo cual nos referirá a los siguientes archivos:

fosa.txt
cosa.txt
rosa.txt

iii) Paréntesis rectangulares los cuales nos permiten reemplazar una serie de caracteres, en una misma posición. Este tipo de referencia solo puede ser usado en los programas "xfer" y "stat". Por ejemplo:

prueba[abcd].z80

nos referirá a los siguientes archivos:

pruebaa.z80
pruebab.z80
pruebac.z80
pruebad.z80

Controles especiales del teclado.

Para el manejo adecuado tanto de la terminal como de las impresoras existen una serie de controles especiales los cuales son ejecutados a través de la consola, oprimiendo la tecla de "Control" y la letra que designe el control necesario a ejecutar. A continuación se da una lista de estos controles.

CNTRL-S Con este control se puede detener un listado que esté apareciendo en la pantalla, para continuar el listado se oprime cualquier otra tecla.
CNTRL-K Mueve el cursor hacia arriba.
CNTRL-L Mueve el cursor hacia la derecha.
CNTRL-J Mueve el cursor hacia abajo.
CNTRL-H Mueve el cursor hacia la izquierda.

Los siguientes controles son para el manejo de la impresora:

CNTRL-L Envía un salto de página a la impresora.
CNTRL-P Prende la opción de la impresora, esto es, todo lo que aparece en pantalla saldrá también hacia la impresora.
CNTRL-T Apaga la opción de salida a impresora. Este control debe de ser enviado por medio de algún programa del usuario, si se utiliza desde la terminal no tiene ningún efecto.
CNTRL-W Este control también prende la opción de salida a impresora, la diferencia es que éste debe ser enviado desde un programa del usuario, para habilitar la salida a la impresora.

3.2.2 Intrínsecos del Sistema.

Los intrínsecos de un sistema son un conjunto de comandos, para diversos usos, que contiene el sistema operativo bajo el cual se está trabajando.

Atributos de un archivo. <ATTR>

Por medio de este intrínseco del sistema podemos asignar o modificar las características de un archivo en cuanto a su seguridad, esto es, podemos proteger nuestros archivos contra escritura sobre ellos, lectura de estos o protegerlos en contra de que sean borrados los archivos. Esta protección es solo de software, si se desea una mayor seguridad se recomienda utilizar la protección del diskette contra escritura. Estos tipos de protección de los archivos se describen a continuación:

<E>rasead previene el que un archivo sea borrado del diskette.
<R>eadad previene que un archivo pueda ser leído.
<W>ritead previene el que se pueda escribir sobre el archivo.
<S>ystemad es archivo del sistema.
<U>serad es archivo del usuario.

Para asignar estos atributos se tendrá que teclear "ATTR" o "ATTRIB" seguido por el nombre del archivo y las letras correspondientes a la protección deseada (E, R, W).

Si se quiere quitar los atributos de protección de un archivo sólo habrá que teclear el comando "ATTR" seguido del nombre de éste, al cual se le quieren quitar los atributos, (no debe de ir ninguna letra después de el nombre del archivo).

Si se desea añadir algún atributo a un archivo además de los que ya tenga, se tendrá que teclear "ATTR" seguido del

nombre del archivo, un signo mas "+" y las letras de la protección ó protecciones que se deseen añadir.

Como ejemplo se tiene un archivo llamado "CARTA.TXT", al cual se le desea proteger, para que no pueda ser leído, para esto solo tendremos que teclear:

```
ATTR CARTA.TXT R
```

A este mismo archivo se va a proteger también, para que nadie pueda escribir en él, de la siguiente forma:

```
ATTR CARTA.TXT +W
```

Si queremos quitarle estas protecciones, para modificar el archivo, solo tendremos que teclear lo siguiente:

```
ATTR CARTA.TXT
```

Directorio de un diskette. <DIR>

Para poder ver el contenido de un diskette dado, solo tenemos que teclear el comando "DIR" e inmediatamente será desplegado en la pantalla el contenido del directorio indicándonos en forma tabular la siguiente información:

- + El nombre del archivo
- + La extensión del archivo (calificativo) si existe.
- + Tamaño en caracteres entre un factor de 1024 (kilobytes).
- + Los atributos de protección del archivo.

Al final del directorio nos da una línea en la cual se indica cuanto espacio hay ocupado en el diskette, cuanto espacio disponible queda en éste, así como cuantos archivos tiene el diskette.

A continuación se da un ejemplo de un directorio de un diskette que se encuentra en el DRIVE "A".

```
A. DIR
prog1  for   7k      datos  txt   10K
texto  txt  20K      inicio com   2K
prueba dat   1K
```

```
*** 5 Files, 7 Entries, 40 K Displayed, 346 K Left ***
```

También se puede obtener el directorio de un diskette que no esté en el "Drive" en el cual estamos ejecutando el comando, esto se puede realizar de la siguiente manera:

Supongamos que estamos ejecutando desde el controlador "A", entonces para listar el directorio del diskette que se encuentra en el controlador "B", necesitamos teclear el comando de la siguiente manera:

A. DIR B:

Lo cual nos desplegaría el directorio del diskette que se encontrara en el controlador "B".

Borrado de archivos. <ERA>

Para borrar uno o mas archivos de un diskette se utiliza el intrínseco del sistema "ERA". Para borrar un archivo llamado CARTA.TXT solo se tendrá que teclear:

ERA CARTA.TXT

También se pueden borrar una serie de archivos por medio de la referencia ambigua de archivos o de caracteres de reemplazo. Así mismo se pueden borrar todos los archivos que contiene un diskette de la siguiente manera:

ERA *.*

Cambio de nombre a un archivo. <REN>

Por medio de este intrínseco del sistema nosotros podemos cambiar el nombre a un archivo, (sin alterar su contenido). La forma de cambiar el nombre a un archivo es el siguiente: primero teclear "REN" seguido del nuevo nombre del archivo, un signo de igual "=", y el nombre del archivo que tiene actualmente. Por ejemplo si queremos cambiar el nombre a un archivo que se llama "CARTA.TXT", y ahora queremos que

se llame "SOBRE.TXT", tendremos que teclear lo siguiente:

```
REN SOBRE.TXT = CARTA.TXT
```

Con lo cual el archivo aparecerá en el directorio del diskette con el nombre de "SOBRE.TXT".

Protección de parte de la memoria. <SAVE>

"SAVE" nos permite guardar en disco partes de memoria, que puede acceder el usuario, esto es de la localidad de memoria 100H hacia arriba y dependiendo del número de páginas que se le especifiquen. Este comando es útil para recuperar archivos que el programa encadenador generó en memoria. Como ejemplo vamos a guardar 5 páginas de 256 bytes de memoria en el disco y se va a llamar ARCH.COM:

```
SAVE ARCH.COM 5
```

La extensión de este tipo de archivos debe de ser "COM" y será un archivo ejecutable de comandos.

Listar un archivo. <TYPE>

Para listar un archivo en la pantalla solo tendremos que teclear:

```
TYPE ARCH.TXT
```

Donde ARCH.TXT es el archivo que queremos listar en la pantalla. Para detener un listado que está siendo desplegado en la pantalla se deberá de teclear CNTRL-S con lo cual se detendrá, y para continuar el listado se teclaea otra vez CNTRL-S. También se puede mandar el listado a la impresora con el comando CNTRL-P. Además si nuestro archivo contiene el comando CNTRL-W el listado saldrá directamente a la impresora.

3.2.3 Utilerías del Sistema.

Las utilerías son una serie de programas que se entregan con un sistema de cómputo, los cuales ofrecen facilidades para el manejo de archivos y de dispositivos periféricos, en éste caso el manejo de diskettes y de los archivos que contienen.

En este capítulo se dará una explicación acerca de cómo inicializar un diskette para que lo pueda utilizar el sistema para grabar información, cómo transferir archivos entre diskettes, como verificar de un diskette su directorio, etc, analizar archivos, y la facilidad de proceso en "batch".

Proceso en 'Batch'. < @ >

La utilería de "Batch" ó Proceso en lote nos permite ejecutar una serie de comandos válidos para "CDOS", secuencialmente en forma inmediata, ó en forma de archivo de comandos, estas dos opciones se describen a continuación:

i) Forma de ejecución inmediata. Si se tiene que ejecutar una serie de comandos una sola vez se puede utilizar esta forma de proceso en lote. La forma de llevar a cabo esto es la siguiente: Primero se tecléa una arroba "@" a lo cual el sistema nos responderá con un signo de exclamación "!" e inmediatamente a este signo podemos tecléar el comando que se requiera finalizando con un "Return" cada línea y así seguir hasta que hallamos tecléado todos los comandos. Para finalizar debemos de tecléar "Return" al comienzo de la última línea. En este momento se comenzarán a ejecutar los comandos en el orden que se tecléó. Por ejemplo:

A.<RETURN>	(Indica al sistema que se van a teclear una serie de comandos para ejecutar inmediatamente)
!DIR<RETURN>	(Nos presenta el directorio del diskette)
!TYPE TEXTO.TXT<RETURN>	(Nos despliega el archivo TEXTO.TXT)
!DIR B:<RETURN>	(Nos presenta el directorio del diskette que esta en el drive B)
!<RETURN>	(Comienza la ejecución de los comandos anteriores)

ii) Si se tiene una serie de comandos los cuales se tienen que ejecutar continuamente, se puede crear un archivo con estos comandos, por medio de alguno de los editores del sistema, y ejecutarlos cuando se requiera. Este archivo debe de contener un comando por línea y además el nombre de este archivo debera de tener la extensión de "CMD". Por ejemplo si se tiene un archivo de comandos llamado:

COMANDOS.CMD

Para ejecutarlo tendremos que teclear:

@ COMANDOS

También existe la opción de pasar parámetros al archivo de comandos. Esto se puede realizar de la siguiente manera: dentro del archivo de comandos se pondrá en el lugar donde se quiera reemplazar por el parámetro, el comando "^1" que le indicará al sistema que el primer parametro que se le pase sera substituido en este lugar. Se pueden utilizar del 0...9 para referenciar a los parámetros. Por ejemplo si nosotros creamos un archivo llamado ejemplo, de la siguiente manera:

DIR ^1
TYPE ^2

Al momento de ejecutar el programa "Batch" pasándole como parámetros el drive de donde queremos el directorio y el archivo que queremos listar, de la forma siguiente:

@ EJEMPLO A: ARCH.TXT

Esto nos va a ejecutar los comandos siguientes:

```
DIR A:  
TYPE ARCH.TXT
```

Analizar un Archivo. < DUMP >

Este programa es util cuando queremos analizar el contenido de un archivo que no podemos listar por medio del comando "TYPE". En éste caso podemos obtener un listado en hexadecimal de cualquier archivo, de la siguiente manera:

```
DUMP arch.txt
```

Lo cual nos dará como resultado un listado en Hexadecimal con su correspondiente caracter en ASCII, del archivo llamado arch.txt. Este listado estará compuesto de registros de 128 bytes.

Inicialización de Diskettes. < INIT >

El proceso de inicialización de un diskette nos sirve, cuando tenemos uno nuevo que nunca ha sido utilizado, el cual hay que formatear para que lo reconozca el sistema y pueda grabar y leer información de él.

Para esta inicialización tenemos que realizar los siguientes pasos:

- 1) Poner el diskette del sistema en el Drive A.
- 2) Poner el diskette a inicializar en el Drive B.
- 3) Teclar el comando INIT

Al teclar este comando el sistema nos preguntará que disco se va a inicializar a lo cual contestaremos oprimiendo la letra B, en seguida el sistema nos hará 5 preguntas concernientes a algunas características de los diskettes, a

estas solo tendremos que oprimir la tecla de RETURN, lo cual nos da las respuestas por omisión asociadas a cada pregunta.

Al dar el RETURN de la última pregunta el sistema procederá a inicializar el diskette y al término de este paso el sistema nos contestara con "A.", esto significa que podemos continuar con el proceso.

Hasta este punto ya esta inicializado el diskette, pero con un área disponible para el usuario de 193 KBytes, para incrementar esta área a 386 K se tiene que etiquetar el diskette, por medio de el programa "STAT" el cual se describe a continuación.

4) Teclar el comando STAT/L B:

En este punto lo primero que hará el sistema será desplegar en la pantalla las características del diskette y a continuación nos hara 4 preguntas, a la primera sólo tendremos que oprimir la tecla de RETURN, la segunda es el nombre del diskette, que puede ser cualquier nombre que se desee, la tercera pregunta es la fecha la cual se dará en la forma siguiente MM/DD/AA, y a la última pregunta solo tendremos que oprimir la tecla de RETURN. Con esto finalizamos el proceso de inicialización y ya podemos utilizar el diskette para grabar información en el.

A continuación se presenta un ejemplo de inicialización, en el cual las palabras o letras que aparecen subrayadas es lo que nosotros tenemos que teclear y lo demas es lo que el sistema nos despliega. (un "#" significa la tecla de RETURN).

A. INIT

Initialize Disks version 02.15

Press: RETURN to supply default answers shown in brackets.
CTRL-C to abort program and return to CDOS.

Disk to initialize (A,B,C,D) ? B

Testing Drive:
Completed

Initialization:
This portion can destroy the data on a disk with no chance of recovery.
Press ESC during initialization to abort the process.

Single or double sided (S/D)? <D> #
Single or double density (S/D)? <D> #

First cylinder (0-27H)? <0> #
Last cylinder (0-27H)? <27> #
Surfaces (0, 1, or Both)? #

Cylinder, Surface: 10H,0

A.

Obtener el estado de un diskette. < STAT >

Por medio de esta utileria podemos obtener un informe acerca del estado de algún diskette o del sistema, tambien se pueden modificar ciertas variables del sistema. La forma de utilizar esta utileria es la siguiente: Primero hay que teclear "STAT" sesuido de una diagonal y una o mas letras las cuales denotan la opción de la utileria que se quiera ejecutar, ademas tambien se puede especificar otro "Drive" distinto del cual se ejecuto la utileria, poniendo la letra del "Drive" que se desee obtener información. A continuación se explican brevemente las opciones con que cuenta esta utileria:

- /a Despliega en orden alfabético los archivos que contiene el diskette, con el espacio utilizado de cada uno de ellos.
- /b Da una breve descripción del espacio utilizable del diskette.

- /d Por medio de esta opción podemos poner la fecha al sistema.
- /e Permite borrar archivos selectivamente.
- /l Con esta opción se etiqueta un diskette, esto es, se le pone nombre, fecha a un diskette. Es útil cuando se inicializa un diskette.
- /t Por medio de esta opción se pone la hora del día al sistema, la cual aparecerá en la línea de estado del sistema.

Además se puede teclear solo la palabra "STAT" y se obtendrá un informe detallado del diskette y del sistema, a continuación se presenta un ejemplo:

```
A.stat
STAT (System Status) version 02.09
SYSTEM MEMORY:
Operating system version      02.36
Total system memory           64K
Operating system size         13K
User memory size              50K
DISK MEMORY:
Disk label                     pruebas
Date on disk                   08-22-82
Total disk space               390 K
Disk space used by directory   4 K
Disk space used by files       0 K
Disk space left                386 K
DRIVE:      Double sided, Dual density
DISKETTE:   Double sided, Double density

DEVICE CONFIGURATION:
CON:= Console 0
PRT:= Printer 0 (PAR:)
RDR:= Reader 0
PUN:= Punch 0
DISK CONFIGURATION:
Cluster size
Sector size
Total directory entries
Directory entries used
Directory entries left
```

Escribir en partes de solo acceso
del sistema < WRTSYS >.

Por medio de esta utilería podemos leer o escribir la imagen residente del sistema operativo de el área destinada para el sistema de un disco (los primeros sectores de éste). La forma mas sencilla de utilizar esta utilería es la siguiente:

WRTSYS B:=A:

La cual copiará del área del sistema del diskette en el drive A, hacia el área del sistema del diskette en el drive B.

Transferencia de Archivos entre Diskettes
< XFER >.

"XFER" nos permite copiar archivos entre dos diskettes ó transferir archivos de algún dispositivo periférico a otro (siempre y cuando sea válido). Existen varias opciones para esta utilidad, las cuales se describen a continuación:

- /v Verifica la copia de los archivos.
- /a Con esta opción se eliminan las marcas de fin de archivo, y es muy útil cuando se están concatenando varios archivos en uno sólo.
- /t Expande los tabuladores que se encuentre en el archivo, por espacios, esta opción es útil cuando se quiere mandar un archivo de texto hacia la impresora.
- /c Compara dos archivos sin transferencia.
- /f Filtra caracteres ASCII ilegales.
- /r Transfiere archivos que tienen la protección contra lectura.
- /z Elimina los datos estadísticos de "XFER" al finalizar la ejecución.

La forma de utilizar esta utilidad, es la siguiente: Primero hay que teclear la palabra "XFER" seguido opcionalmente de una diagonal y una o varias letras para definir el tipo de transferencia que queramos, y a continuación el especificador de disco hacia el cual se va a hacer la transferencia seguido de un signo de igual "=" y el nombre del archivo que se desea transferir. El especificador de disco puede contener también un nombre válido de archivo y será como se llame el archivo al finalizar la transferencia. A continuación se dan una serie de ejemplos :

XFER/V B:=A:TEXTO.TXT

Se transfiere el archivo llamado "TEXTO.TXT" que está en el drive A hacia el diskette que está en el drive B, y además verifica que la copia no contenga errores.

XFER/A A:TESIS.TXT=B:CAP1.TXT,B:CAP2.TXT

De esta forma se están uniendo los archivos "CAP1.TXT" y "CAP2.TXT" que están en el drive B, hacia el drive A y en el archivo llamado "TESIS.TXT".

XFER/T PRT:=CARTA.TXT

Se manda a la impresora el archivo "CARTA.TXT" expandiendo los tabuladores que contenga.

3.2.4 Paquetes del Sistema.

Administrador de Bases de Datos. < DBMS >

Este es un paquete desarrollado por CROMEMCO para el manejo de bases de datos. Por medio de este sistema se pueden captar los datos que van a conformar la base de datos y posteriormente hacer recuperaciones de la información haciendo ecuaciones lógicas sobre los datos a recuperar y además especificar que datos en especial se requiere que aparezcan en los listados.

Este sistema está constituido de un programa llamado "RUN" que es el que ejecuta al sistema y el paquete en sí que se llama "DBMS".

Generador de reportes de Bases de Datos. < DBR >

Este es un sistema de fácil uso para la emisión de reportes de alguna base de datos (no necesariamente de DBMS de CROMEMCO), éste es un lenguaje muy similar al basic por medio del cual se especifican la forma de presentar los datos, verificación de éstos y si se desea ordenar los datos por alguno o varios campos con diferentes criterios.

Los archivos necesarios para la utilización de este paquete son: DBR, DBRPREP, DBRLIB, DBRCO.

El procedimiento para generar un reporte de una base de datos es el siguiente: Primero hay que generar un archivo con las especificaciones del reporte por medio de algún editor de texto (Screen), ya que se tiene éste se compilara con DBR

para los errores de sintaxis, a continuación se tiene que preparar este archivo con DBRFREP y DBRLIB y posteriormente se ejecutará el reporte con el programa DBRGO.

Editor de textos "SCREEN"

Mediante este editor de textos, podemos generar cualquier tipo de archivo, ya sea un archivo que contenga instrucciones de WFL o todo un texto.

Para generar o actualizar un archivo, se invoca al editor con el nombre de éste, y con el nombre del archivo del usuario de la siguiente forma:

```
SCREEN B:NOMARCH.TXT
```

Lo cual nos genera un archivo nuevo si es que no existe ya uno con ese nombre, el cual residirá en el DRIVE B y se llamará NOMARCH.TXT. Una vez hecho esto, nos presentará en la pantalla el MENU, que corresponde al conjunto de instrucciones con que cuenta el SCREEN. Los controles para el movimiento del cursor son los siguientes:

```
CNTRL-K Mueve el cursor hacia arriba.  
CNTRL-J Mueve el cursor hacia abajo.  
CNTRL-L Mueve el cursor hacia la derecha.  
CNTRL-H Mueve el cursor hacia la izquierda.
```

Editor de Textos. < EDIT >

Este editor de textos llamado Edit permite lo mismo que Screen el crear editar y guardar archivos tipo ASCII. Este editor puede manejar las ediciones en base a caracteres, palabras o líneas enteras. El manejo de este editor es ya un poco obsoleto dado que Screen es mucho más poderoso, por lo tanto se recomienda el uso de éste último. Además si no se tiene una pantalla del tipo 3102 no se puede utilizar Screen porque éste es un editor de pantalla.

3.2.5 Lenguajes de Programación.

Sbasic.

Este lenguaje es un intérprete interactivo el cual tiene muchas características de los lenguajes estructurados como Pascal, Algol etc. SBasic fue desarrollado por CROMEMCO. Dado que es un lenguaje que permite la ejecución directa de los comandos tiene integrado un editor de programas. Además permite la facilidad de generar varias versiones del basic por medio de "BASIDGEN" que cubran las necesidades del usuario, reduciendo el espacio de memoria que ocupa el intérprete el cual va desde 20K (versión mínima del Basic) hasta 32K, ocupando todas las facilidades que da este lenguaje. Entre las características más importantes de este lenguaje se destacan las siguientes:

Nombres de variables de hasta 31 caracteres.

Las etiquetas de las proposiciones pueden ser alfabéticas en vez de números de línea.

Se permite el uso de las estructuras REPEAT, WHILE, IF .. THEN .. ELSE y PROCEDURE.

Instrucciones para manejo de módulos los cuales pueden ser cargados de archivos en disco de biblioteca de procedimientos a memoria. Esto permite el crear programas muy grandes y poder dividirlos en módulos para su ejecución en la máquina sin tener que preocuparse por el espacio en memoria.

Manejo poderoso de cuerdas (strings).

Facilidad de crear archivos KSAM Keyed Sequential Access Method para recuperación de información por medio de llaves de acceso.

Un editor de textos por líneas integrado al sistema.
Para ejecutar este programa sólo hay que teclear

A.SBASIC

Y estaremos bajo el control de SBasic.

Cbasic.

Este lenguaje es un compilador/interprete diseñado para trabajar bajo el sistema operativo CP/M. Y como CDOS es compatible con CP/M se puede ejecutar en los sistemas CROMEMCO.

CBasic tiene una serie de características adicionales a el Basic estandar, estas características adicionales son el uso de las construcciones IF .. THEN .. ELSE y WHILE , acceso a archivos de disco secuencial y en forma aleatoria (Random). Permite que los nombres de variables sean de 31 caracteres máximo, además de la definición de funciones de multiples líneas, variables enteras, reales y tipo cuerda (strings). Tambien se pueden encadenar varios programas teniendo un área común de datos.

Cbasic consiste de dos programas. El primer programa es el compilador, traduce el programa fuente del usuario en código objeto el cual es puesto en un archivo intermedio de disco. El segundo programa es el monitor de ejecución, ejecuta directamente el código del archivo intermedio.

El programa fuente del usuario puede ser creado mediante algun editor del sistema y deberá de tener la extensión de "BAS". El compilador se llama CBAS.COM. El programa intermedio de código tendrá el mismo nombre que el fuente solo que su extensión sera "INT". El programa que ejecuta al programa objeto del usuario se llama "RUN.COM".

Para compilar y ejecutar un programa que se llama "prueba.bas" y que fue generado por medio de "Screen" se siguen los siguientes pasos:

i) Compilar el programa :

A.CBAS PRUEBA

El cual nos generara un archivo llamado "Prueba.int".

ii) Ejecutar el programa:

A.RUN PRUEBA

Hay que hacer notar que no es necesario poner las extensiones de los archivos tanto en la compilación como en la ejecución.

Para una definición detallada del lenguaje consultar el manual "CBASIC REFERENCE MANUAL." Agosto 1978. version 2.

EBasic.

Este es un Basic de menor tamaño que SBasic (16 K) el cual tambien es un intérprete esto significa que se pueden teclear una línea de código y ejecutarla inmediatamente. El Basic Extendido de 16K es un compilador "incremental" escrito en código de Z-80. Este compilador "incremental" es un tipo de intérprete que no solamente chequea la sintaxis completamente de una línea de código sino que también resuelve inmediatamente cualquier referencia de elementos no definidos. La mayor ventaja de el compilador "incremental" es la velocidad de ejecución. Como las referencias a las variables y a los números de línea están ya resueltos no se pierde tiempo de ejecución en la búsqueda en tablas de los elementos tales como las variables. Las instrucciones que soporta este Basic son las comunes a los Basic Estándar que se encuentran en el mercado.

La forma de utilizar este Basic es simplemente teclear:

A.EBASIC

Y estarémos bajo el control de éste Basic.

Fortran.

Este es un lenguaje universal el cual es soportado por la gran mayoría de computadoras en el mercado. El Fortran de CROMEMCO es el Fortran aceptado por la ANSI en el documento X3.9-1966 lo cual significa que es un Fortran estandar con algunas extensiones y restricciones. La forma de utilizar este lenguaje es la siguiente:

Primero se tiene que compilar el programa fuente como se describe a continuación: Teclrear "FORTRAN" con lo que se ejecuta el compilador de Fortran y nos responderá con un asterisco "*" para indicarnos que le podemos dar el comando necesario, por ejemplo:

```
A.FORTRAN *=PRUEBA
```

Lo cual compilará el programa llamado prueba.for y nos producirá un archivo objeto llamado prueba.rel

De esta forma, ya se tendrá el programa compilado faltando solo el siguiente paso que es el encadenado y cargado en memoria para su ejecución.

Existen una serie de opciones al momento de la compilación, que se pueden encontrar en el manual de Fortran.

El segundo paso es linkar el programa por medio de "LINK", para ejecutarlo hay que teclrear:

```
A.LINK
```

A lo cual nos responderá con un "*" indicandonos que podemos darle algún comando valido para LINK.

La forma mas sencilla es darle solo el nombre del programa seguido de una diasonal y la letra N para que lo prepare y nos genere un archivo con la extencion de ".COM" el cual ya podrá ser ejecutado. Existen una serie de opciones que pueden ser encontradas en el manual de Link.

Ratfor.

Ratfor es un acrónimo de fortran racionalizado y es preprocesador del lenguaje el cual transforma a fortran estandar para que pueda ser compilado y lisado para su ejecución. Este lenguaje provee todas las facilidades de Fortran y ademas permite las estructuras de control siguientes: FOR, REPEAT...UNTIL, WHILE las cuales son una ventaja para la programación estructurada. RATFOR esta descrito en detalle en el libro "SOFTWARE TOOLS" de B.W. Kernishan y P.J. Plausen. El Ratfor de CROMEMCO cumple con todas las especificaciones señaladas en dicho libro. Para preprocesar un programa escrito en RATFOR sólo hay que invocar al preprocesador de Ratfor que nos producirá un archivo con un programa en fortran el cual podrá ser compilado en la forma normal. Existen una serie de opciones para este programa los cuales vienen descritos en el manual de RATFOR.

Assembler.

El propósito de un ensamblador es el de tener un medio de traducir de mnemonicos facilmente recordables y entendibles, a código objeto, el cual puede ser cargado en la memoria y ejecutado como un programa. El Macro Ensamblador Relocalizable que reside en el disco para el Z80, es un ensamblador de dos pasadas, que lee el programa fuente de un archivo en el disco, lo ensambla, y produce un código objeto relocalizable y/o un archivo imprimible.

El Encadenador/Cargador puede ser utilizado después, para localizar el código objeto en cualquier parte de memoria. El código de máquina completamente ensamblado y lisado debe de ser guardado en el disco como un archivo tipo *.COM para su ejecución como un programa de comandos.

La utilización de un ensamblador relocalizable y un ligador, provee una de las maneras más versátiles de generar programas en lenguaje de máquina para las computadoras en general. Por medio de estos dos programas, es posible crear y ensamblar un gran número de módulos separados, y posteriormente ligarlos todos juntos al momento de ejecutarlos. O bien se puede ligar un programa previamente ensamblado desarrollado por el usuario a la biblioteca de programas objeto. Además es posible el ensamblar programas utilizando un compilador (por ejemplo el FORTRAN), y ligar estos módulos objeto, para generar subrutinas o programas. al mismo tiempo es posible que el programa se pueda poner y ejecutar en cualquier parte de la memoria. Este ensamblador soporta Macros así como los condicionales.

Cobol.

El COBOL de CROMEMCO es estandar con las especificaciones ANSI X3.23-1974, además tiene una serie de características muy importantes en el manejo de la pantalla para efectos de captura de datos y además maneja archivos ISAM.

Para poder utilizar el sistema cobol se necesitan los siguientes archivos en el diskette:

El compilador de cobol: cobol.com

cobol1.ovr cobol3.ovr
cobol2.ovr cobol4.ovr

El sistema para el momento de ejecución: coblib.rel

además del siguiente software de utileria:

link.com
lib.com

La preparación de un programa en cobol comprende los siguientes pasos:

- i) Crear el archivo fuente por medio de algun editor de textos.
- ii) compilar el archivo con el compilador de cobol.
- iii) Preparación y cargado del programa con programa de LINK.

A continuación se dará un ejemplo para poder ejecutar un programa en cobol, de la manera más sencilla, para mayores detalles y una descripción detallada de la sintaxis de los comandos referirse al manual de este lenguaje.

- i) Se crea un archivo por medio de SCREEN cuya extensión sea ".COB".

A.SCREEN prueba.cob

- ii) A continuación se compila este programa generando una archivo tipo ".REL", de la siguiente manera:

A.COBOL PRUEBA.REL,TTY:=PRUEBA.COB

El cual nos generará un archivo llamado prueba.rel y además nos desplegará en la pantalla el resultado de la compilación del archivo prueba.cob.

- iii) El siguiente paso es encadenar el programa y cargarlo en memoria, por medio de "LINK". En este paso se convierte el código objeto relocable generado por el compilador en una versión ejecutable y además lo encadena con el sistema de ejecución de cobol. Esta versión ejecutable es construida en memoria y guardada en disco posteriormente mediante el siguiente comando:

A.LINK PRUEBA,PRUEBA/N/E

El programa "LINK" asume que el programa prueba tiene una extensión "REL". Con este comando se generará un archivo llamado prueba.com el cual será el que se ejecute posteriormente, de la siguiente forma:

A.PRUEBA

De esta forma se puede compilar y ejecutar un programa en cobol.

3.2.6 Características de Hardware

El sistema CROMEMCO es un sistema basado en un microprocesador de 8 bits, el Z80A ; cuenta con gran cantidad de tarjetas de Hardware para su completo desarrollo; entre ellas tiene:

- Procesador Z80A
- Memoria de 64KB
- Controlador de discos.
- Controlador de impresora.
- Interfaz Serie-Paralelo, Paralelo-Serie.
- Convertidor analógico/digital y digital/analógico.
- Grabador de EPROM
- Computador en una sola tarjeta.
- Procesador de Entrada/Salida.
- Puertos de Entrada/Salida.
- Disco duro de alta velocidad.
- Monitor de color
- Tableta de graficación.

Todas las tarjetas anteriores son compatibles con el Bus S-100.

Este sistema tiene una gran expansión en la actualidad, pues son más los periféricos, así como la paquetería que cada día sale al mercado para esta máquina.

Para fines del año de 1982 fué anunciado el poder trabajar con un procesador de 16 bits (68000). Para esto es necesario adicionar una tarjeta de hardware que contará con el Z80 y el 68000, una tarjeta de memoria de 256KB RAM y el sistema operativo correspondiente.

En general la arquitectura mínima del sistema CROMEMCO es la siguiente:

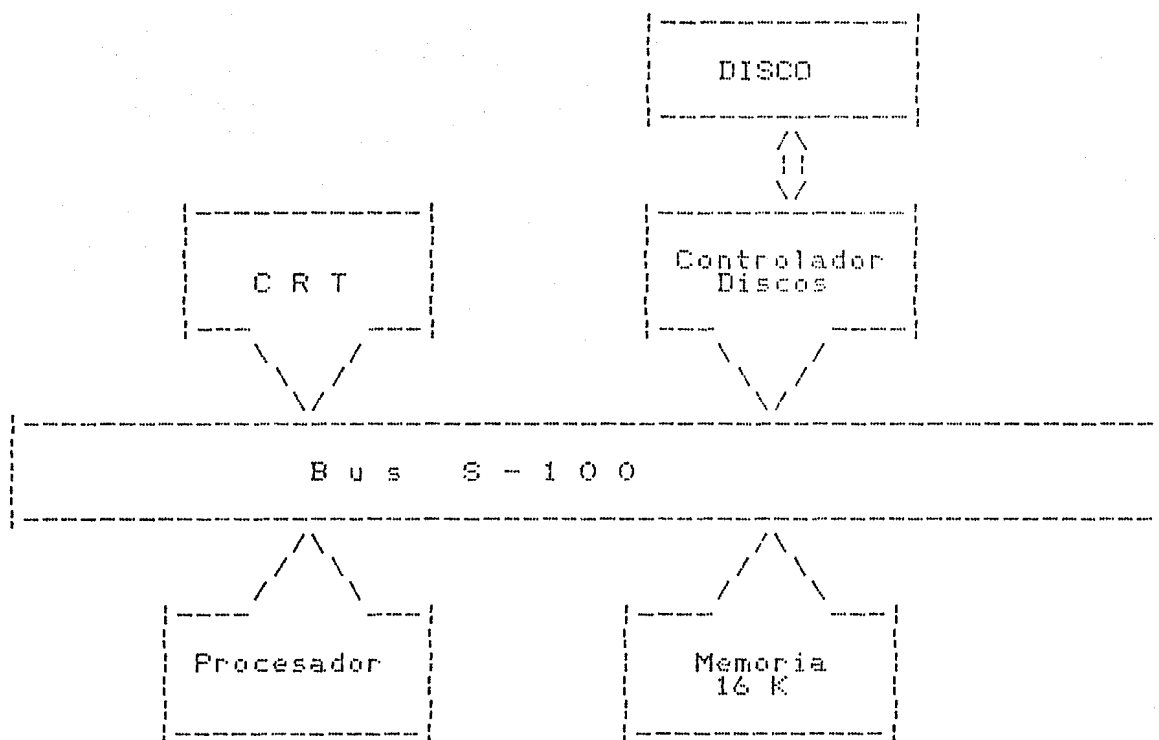


Diagrama de la arquitectura mínima del sistema CROMEMCO.

Con esta configuración básica es posible desarrollar los programas de aplicación que se requieran, pues se tiene todo lo necesario.

Es posible expandir la memoria hasta 64K, implementar un controlador de impresora, discos de mayor capacidad y velocidad, y que varios usuarios compartan los recursos de la máquina simultáneamente (para esto es necesario la adición de algunas tarjetas de hardware y un sistema operativo especial llamado CROMIX).

Hardware Utilizado.

A continuación se describirán en forma general cada una de las tarjetas que componen la arquitectura básica de este sistema. Primeramente se describirán las tarjetas de propósito general, y posteriormente una tarjeta de comunicación de datos.

ZPU. Unidad central de procesamiento con Z-80.

- Procesador.

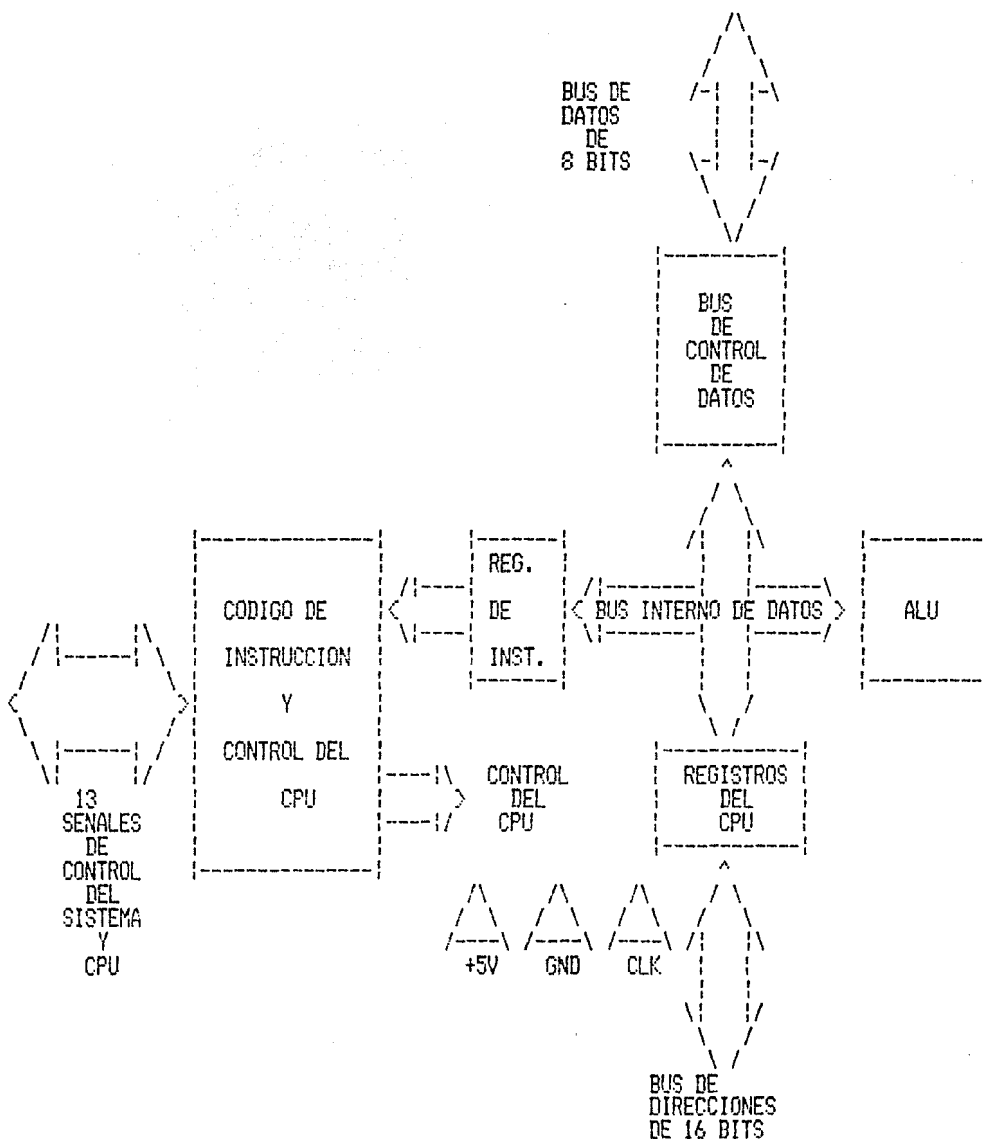
Se utiliza el Z-80A, que es una versión del Z-80, pero que es capaz de utilizar un reloj de 4MHZ.

Este procesador tiene las siguientes características:

Registros de 8 bytes:	16
Registros de 16 bytes:	10
Registros de índice:	2
Modos de dirección:	10
Dirección de E/S:	256
Bits de bandera:	6
Voltaje requerido:	+5V
Velocidad de reloj máxima:	4MHZ
Compatible con TTL :	SI
Modos de interrupción :	4
Refrescamiento de memoria automático:	SI
Velocidad relativa en ejecución:	1.0

- Arquitectura del CPU.

Podemos representar la arquitectura interna del CPU a través de un diagrama de bloques como el que se muestra a continuación:



- Especificaciones técnicas de la tarjeta del ZPU.

Frecuencia del reloj:	Versión en 4MHz del Z-80A
Conjunto de Instrucciones:.....	158 instrucciones incluyendo las 78 instrucciones del procesador 8080.
Salto automático al encendido:	Habilitado por Hardware.
Direcciones posibles al salto automático:	16 direcciones seleccionables por switches.
Generación de estados de espera:.....	0-4 estados de espera seleccionables por Hardware.
Ciclo de espera de máquina:	Seleccionables por Hardware.
Compatibilidad de BUS:	S-100
Necesidades de alimentación:	+8 V DC y 1.1 Amp.
Rango de operación:	0-55 grados Centígrados.

- Salto automático de memoria al encendido.

El circuito de salto automático instalado en la tarjeta del ZPU, permite a ésta el ser usada en un sistema de BUS S-100 sin controles en los paneles externos de la microcomputadora.

Cuando el sistema es encendido, el hardware del ZPU fuerza un salto automático a una de la 16 posibles localidades de memoria disponible, seleccionables por el switch de cuatro posiciones.

En la dirección que se selccione con estos switches, se encuentra el sistema operativo residente en ROM (RDOS).

El salto automático a las direcciones de memoria correspondiente a cada codificación de los switches, se tabula a continuación:

A15	SWITCH A14	A13	A12	Cuando se enciende SALTA A:
0	0	0	0	0000H
0	0	0	1	1000H
0	0	1	0	2000H
0	0	1	1	3000H
0	1	0	0	4000H
0	1	0	1	5000H
0	1	1	0	6000H
0	1	1	1	7000H
1	0	0	0	8000H
1	0	0	1	9000H
1	0	1	0	A000H
1	0	1	1	B000H
1	1	0	0	C000H
1	1	0	1	D000H
1	1	1	0	E000H
1	1	1	1	F000H

Cuando se carga el sistema operativo a la memoria una señal llamada "Phantom", deshabilita la memoria ROM (en donde esta RDOS) para tener 64K de memoria RAM (65,536 bytes de memoria de acceso aleatorio).

- Selección de frecuencia del reloj.

El microprocesador Z-80A puede trabajar con un reloj de 4 MHz (con un ciclo de tiempo de 250 nseg.) ó 2 MHz (con un ciclo de tiempo de 500 nseg.) . La frecuencia de operación es seleccionable por medio de un switch tipo "toggle" . La línea del "bus" S-100 previamente etiquetada como "STCK" es utilizada por el ZPU como una línea indicadora de 4 MHz.

- Selección de estados de espera.

El ZPU provee un generador de estados de espera en la misma tarjeta, de tal manera que son compatibles la frecuencia del reloj del Z-80A y el tiempo de acceso al sistema de memoria que se tenga.

El ZPU permite la inserción de dos tipos de estados de espera durante cada ciclo de máquina.

- Selección de memoria espejo.

El microprocesador 8080 repite ó espejea la dirección de los 8 bits del puerto de Entrada/Salida en el orden alto y bajo de la dirección del BUS. A pesar de que esta característica no pertenece al procesador Z-80A, la tarjeta del ZPU está diseñada para imitar esta característica a través de una circuitería especial, asegurando que será compatible con un sistema que se actualice de 8080 al Z-80A.

- Refrescamiento de memoria.

Todos los tipos de tarjetas de memoria dinámica requieren un refrescamiento de las direcciones proporcionadas por el Z-80A reflejadas en las líneas de las direcciones de orden alto.

16FDC

El 16FDC es un controlador de discos de la segunda generación que lanza al mercado CROMEMCO. Esta tarjeta está provista de un sistema completo para la operación de los floppies, que pueden ser indistintamente de 8 pulsadas y 5 pulsadas. Incluye un puerto serie de Entrada/Salida (I/O) para una terminal de video con interfaz RS-232, una memoria pre-programada ROM (Read Only Memory), en donde se encuentra grabado un programa monitor llamado RDOS (Resident Disk Operatins System), por medio del cual se inicia el sistema (boot), así mismo posee comandos por medio de los cuales se realiza un diagnóstico básico de la memoria y drives. Este monitor normalmente funciona solo cuando se enciende la máquina, ó después de un reset. Con este monitor se inicia el sistema y en este momento se desconecta el monitor transfiriéndose el control al sistema operativo.

El 16FDC normalmente maneja hasta 4 drives (el drive es el dispositivo físico en donde se pone el disco para su lectura y/o escritura) en una cadena de prioridades; sin embargo hasta 16 drives pueden ser encadenados si es necesario (si el drive es capaz de decodificar las direcciones de las líneas).

64KZ

La tarjeta CROMEMCO 64KZ es una memoria dinámica para Lectura/Escritura de 65,536 bytes (64K bytes), compatible con el bus S-100. Esta tarjeta incorpora una memoria dinámica RAM (Random Access Memory) para esto utiliza un circuito integrado llamado: TMS 4116-15 (16K x 1 bit). Opera en un tiempo de acceso máximo de 250 nsec. Esto significa que la 64KZ puede operar en los sistemas Z80 con un reloj de 4MHz sin utilizar estados de espera en lo absoluto. Esta tarjeta maneja la señal de selección de banco, es posible el aumentar lógicamente la memoria, pues se puede intercambiar bancos de memoria por software, esta es una opción muy importante, pues se utiliza grandemente en sistemas operativos para multiusuarios.

Esta tarjeta ofrece las siguientes características:

- 64K bytes de memoria para Lectura/Escritura (R/W) en una sola tarjeta de memoria para el bus S-100.
- Un tiempo máximo de acceso de 250 nanosegundos.
- Compatibilidad con el Z80 y el 8080 CPU.
- Organización en dos bancos independientes de memoria (A y B).
- Selección del banco, permitiendo una expansión de la memoria más allá de los 64K bytes.

Especificaciones técnicas:

Capacidad de Memoria:	65,536 BYTES (64K BYTES)
Tiempo de acceso a memoria:	250 Nanosegundos (maximo)
Estados de espera a 2Mhz:	No se Requieren
Estados de espera a 4Mhz:	No se Requieren
Compatibilidad con bus:	S-100
Requerimientos de potencia:	+ 8 volts y 1.8 amperes (max) +18 volts y 0.45 amperes (max) -18 volts y 0.08 amperes (max)
Temperatura de operación:	0 - 55 Grados Celsius
Tipo de memoria:	TMS 4116-15,16x1 RAM dinámico

TU-ART

La tarjeta TU-ART (Twin Universal Asynchronous Receiver and Transmitter ó Receptor y Transmisor Universal Asíncrono Doble) provee dos canales duplex serie, para el intercambio de información; dos canales de intercambio de datos en paralelo; y diez contadores de tiempo (Timers). Existe un puerto por el cual se le puede dar una cantidad al timer deseado, esta cantidad es decrementada en 1 cada 64 microsesundos a partir de la carsa inicial. Cuando la cuenta llega a cero, se produce una petición de interrupción y el timer es deshabilitado. Como la cuenta máxima es de 255, el intervalo de tiempo máximo que puede contar un timer es: 254×64 microsesundos = 16.32 milisesundos. La información del estado del dispositivo se puede obtener por interrupciones ó bien presuntando periódicamente (Polling). Dos líneas de petición de interrupción están disponibles para el usuario, a través de los puertos serie y/o paralelo. El TU-ART tiene su propio reloj controlado por un cristal de cuarzo, y la comunicación con el bus S-100 se realiza asincrónamente, así que la frecuencia del reloj del CPU no es crítica, es decir el CPU se puede operar a 2 MHz ó bien a 4 MHz sin que esto afecte la transmisión de información entre el CPU y el TU-ART. El TU-ART incorpora dos circuitos integrados, controladores de los puertos del número TMS 5501 NMOS I/O.

EL CPU normamente "ve" al TU-ART como un puerto dual de Entrada/Salida con capacidad de interrupción. El CPU lee los datos de esta tarjeta a través del bus S-100 ejecutando para ello una instrucción como la siguiente:

IN A,(PUERTO)

Existen catorce puertos utilizados para la Entrada/Salida de comandados por uno de los TMS 5501 y otros catorce puertos por el otro dispositivo (TMS 5501). Es posible el intercambiar por medio de software los TMS 5501 (el dispositivo A será el B y lo contrario). La dirección base de ambos dispositivos (A y B) es seleccionada por unos micro-switches que se encuentran en esta tarjeta. Toda la información entra o sale del TU-ART a través del acumulador, y por medio de los registros que tiene cada puerto.

Velocidades

El registro 00 de salida (Out) es el registro para la selección de la velocidad. El diagrama de bits es el siguiente:

D7	D6	D5	D4	D3	D2	D1	D0
Bits de STOP	9600	4800	2400	1200	300	150	110

D7 STOP

Un "uno" en el bit 7 selecciona un bit de stop para la recepción y transmisión serie. Un "cero" en este bit selecciona dos bits de STOP.

D6 - D0 VELOCIDAD

Un "uno" en alguno de los posibles bits selecciona la correspondiente velocidad. Si por equivocación se selecciona mas de bit se tomará en cuenta el bits más significativo. Por ejemplo, si queremos transmitir a 4800 bits/seg y 1 bit de stop, tendremos:

1 0 1 0 0 0 0 0
A O

Para fijar esta velocidad, se ejecuta la instrucción:

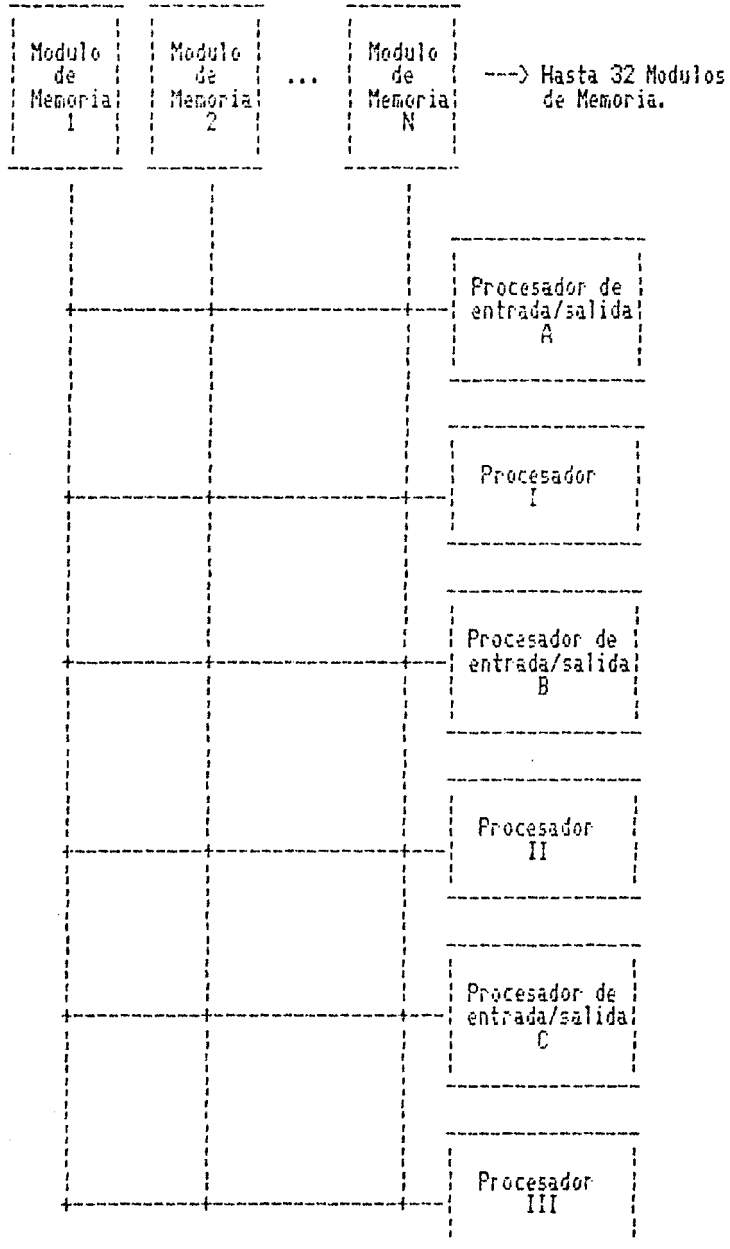
LD A,0A0H
OUT (PUERTO),A

3.3 Equipo Burroughs B-6700.

El equipo Burroughs que será descrito en el presente capítulo, es el modelo B6700, este equipo es el utilizado como computador central o anfitrión. A continuación se describirán las características de dicho equipo, tanto de "Hardware" como de "Software", poniendo especial énfasis en la parte de comunicación de datos, esta descripción no será muy profunda, dada la complejidad del equipo.

3.3.1 'Hardware' del Sistema.

La arquitectura básica del B6700 es un arreglo matricial de procesadores y módulos de memoria como se observa en la siguiente figura:



Descripción General del Sistema:

- Circuiteria Monolitica.
- Memoria expandible a 1,048,576 palabras.
- Tiempo de ciclo de memoria de 1.2, 1.5 microsegundos y 500 nanosegundos.
- Configuración periférica expandible a 256 unidades.
- Capacidad de hasta tres procesadores de entrada/salida permitiendo hasta 36 operaciones simultaneas de E/S.
- Software para comunicación de datos para proceso remoto y manejo de archivos.
- Almacenamiento en disco de hasta 36×10^{12} bytes.

Procesador Central.

A este sistema se pueden conectar de uno a tres procesadores centrales (CPU), los cuales estan divididos en dos partes: Recursos funcionales (FUNCTIONAL RESOURCES) y Algoritmos de operadores (OPERATOR ALGORITHMS). Los recursos funcionales es la parte del procesador, que se encarga de realizar las operaciones aritmeticas y lógicas, manejar la memoria, el "stack" las interrupciones etcétera y los algoritmos de los operadores son la lógica necesaria para procesar cada operador de la B6700 las cuales son instrucciones a ejecutar sobre el "stack". El equipo cuenta con dos procesadores modelo B6748, con un reloj de 5 MHertz para la unidad lógica y de 10 MHertz para la unidad aritmética. A continuación se muestra la organización antes mencionada:

<----- FUNCTIONAL RESOURCES -----><-- OPERATORS ALGORITHMS -->

ARITHMETIC UNIT 48 BIT ADDER	ADDRESS PROC. UNIT 20 BIT ADDER	MEMORY CONTROLLER	OP. FAMILY CONTROLLER-A	OP. FAMILY CONTROLLER-F
DATA REGISTERS (A,B,C,X,Y,P) (51 BITS)		PROGRAM SEQUENCE CONTROLLER	OP. FAMILY CONTROLLER-B	OP. FAMILY CONTROLLER-G
		STACK ADJUST CONTROLLER	OP. FAMILY CONTROLLER-C	OP. FAMILY CONTROLLER-H
		INTERRUPT CONTROLLER	OP. FAMILY CONTROLLER-D	OP. FAMILY CONTROLLER-J
		TRANSFER CONTROLLER	OP. FAMILY CONTROLLER-E	OP. FAMILY CONTROLLER-K
ARITHMETIC CONTROLLER	STRING OP. CONTROLLER			

Procesadores de Entrada/Salida.

Los procesadores de comunicación de datos, son los encargados de controlar la transferencia de datos entre la memoria principal y los diferentes dispositivos periféricos conectados al sistema, esto sin necesidad de interrumpir al procesador central. Se pueden conectar hasta tres procesadores de E/S y cada uno de ellos puede manejar hasta doce operaciones simultáneas de E/S. Además a cada uno de estos procesadores, se le pueden conectar hasta 4 "DCP" (Data Communications Processor) los cuales son los encargados de las comunicaciones en la parte de teleproceso de la B6700. El equipo cuenta con dos procesadores de I/O modelo B6780, a cada uno de los cuales está conectado un DCP. Al DCPO están conectados dos Clusters cada uno de los cuales tiene 16 Line Adapters, al DCP1 está conectado un Cluster con 16 Line Adapters, lo que nos da un total de 48 líneas disponibles.

Procesador de Comunicación de Datos.

El procesador de comunicación de datos (BCP) es una minicomputadora programable de propósito especial. Esta se encarga de la transmisión y recepción de mensajes de las varias líneas de comunicación, opera en forma independiente del equipo central y es la interfaz de comunicación entre el sistema central y los concentradores de puertos (Adapter Cluster), los cuales se pueden conectar desde uno hasta 16 a cada (BCP).

Las funciones básicas del "BCP" son las siguientes:

i) Controlar las estaciones remotas que se encuentren asociadas a cada línea de comunicación. Controlar el formato de los datos que son transmitidos y recibidos.

ii) Guardar los caracteres recibidos a través del "adapter cluster" y formar conjuntos de palabras de seis caracteres, los cuales serán transmitidos a la memoria principal.

iii) Tomar del sistema principal mensajes (palabra por palabra) para posteriormente transmitirlos al "adapter cluster" caracter por caracter.

iv) Traduce los códigos recibidos (según la programación que se le haya dado), normalmente de ASCII (manejado por las terminales) a EBCDIC (manejado por B6700).

v) Controla la recuperación de errores, verificación de paridad.

vi) Maneja las peticiones del sistema central de llamado o contestación de líneas telefónicas y participa en la reconfiguración dinámica de la red de teleproceso.

Los elementos basicos del "DCP" son:

i) "Scratch-Pad-Memory".- Memoria de circuitos integrados de ocho palabras de 52 bits cada una con un tiempo de lectura de 40 nanosegundos y 50 nanosegundos para escritura.

ii) Memoria local.- Memoria de nucleos magneticos de 4096 palabras de 52 bits cada una con un ciclo de 560 nanosegundos.

iii) Registros.- Estos son 14 divididos en registros de datos e información de control y registros de instrucciones.

Adapter Cluster.

El Adapter Cluster es la interfase entre el DCP y los Line Adapters. Cada DCP puede tener hasta 16 Adapter Cluster y cada uno de estos sirve como un multiplexor a 16 líneas o Line Adapters. Las funciones básicas del Adapter Cluster son:

- i) Ensamblado y desensamblado de caracteres.
- ii) Sincronización.
- iii) Operaciones de muestreo para mantener la disciplina de la línea.
- iv) Intercambio de información con uno o dos DCP's.

Los elementos basicos del Adapter Cluster son:

i) "Scratch-Pad-Memory".- Memoria de circuitos integrados de 16 palabras de 56 bits cada una con un tiempo de acceso de 40 nanosegundos en la lectura y 50 nanosegundos en la escritura.

ii) "Buffer Associative Register".- Este registro es utilizado por todos los line adapter y consiste de varios campos donde almacena temporalmente información de datos, control y tiempo.

Line Adapter.

Cada Line Adapter determina una línea, y es el encargado de manejar el intercambio de bits o caracteres entre el Adapter Cluster y la línea de comunicación de datos. Básicamente ajusta los cambios de niveles de voltaje entre ambos dispositivos y es donde el dispositivo conectado a la línea y el adapter cluster se sincronizan entre sí. El line adapter puede ser de varios tipos, dependiendo de las velocidades de transmisión y de la forma de conectar a este los dispositivos, algunos de estos dispositivos son:

- i) Cable directo y velocidades de 150 a 9600 Bauds.
- ii) Conexión a través de modem a velocidades de 150 a 2400 Bauds.
- iii) Unidad automática de llamado "ACU".
- iv) Línea de respuesta de audio "ARL".
- v) Entrada de teléfono tipo "touch tone".

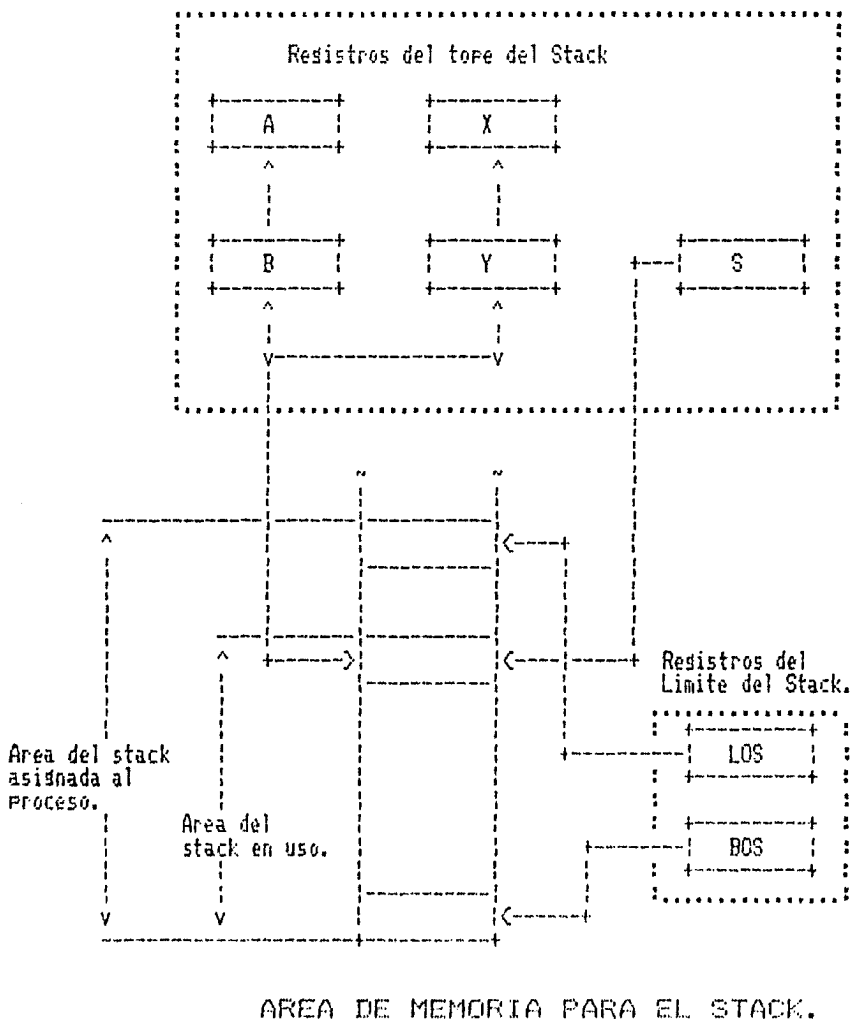
3.3.2 'Software' del Sistema.

Describir completamente el Software del equipo B-6700 seria muy ambicioso, para tratarlo completamente en un solo capitulo, por lo cual nos centraremos en las partes más sobresalientes de éste. En el presente capitulo, comenzaremos por describir lo que es un 'Stack' y su funcionamiento y a continuación se hablara acerca del sistema operativo y de las partes que lo componen, despues se enumerarán las utilerias con que cuenta el sistema y se finalizará con una breve explicación del manejo de memoria.

Concepto de 'Stack'.

Las variables y constantes de un programa son guardados en una área de memoria llamada 'Stack' o pila en español. Esto es, el stack es un area de memoria asociada a un proceso y nos sirve para almacenar en forma dinamica el código y variables y constantes, utilizadas por un proceso, su tamaño es fijo para un determinado proceso pero el manejo de las localidades ocupadas dentro del stack varian constantemente y puede ser extendido en caso necesario.

El funcionamiento del stack es análogo a una Pila física de objetos, donde el último elemento puesto, es el tope del stack. Cuando los elementos son quitados (uno a la vez) del stack el elemento en el tope del stack es el primer elemento a ser quitado. El elemento en el fondo del stack permanece ahí hasta que todos los los elementos del stack han sido removidos. El stack no solo provee un medio sencillo y dinámico de llevar la historia de un proceso dado, sino que permite el uso de cuerdas de código (de un programa) basados en notación polaca.



AREA DE MEMORIA PARA EL STACK.

Registros del tope del stack:

- A,B,X,Y: Registros de alta velocidad donde son efectuadas las operaciones.
- S ('Stack pointer register'): Registro que apunta a la ultima palabra colocada en el stack.

Registros de limite del stack.

- LOS ('Limit of stack register')
- BOS ('Base of stack register')

Estos registros son usados para definir la base o el limite del área de memoria asignada al Stack.

El funcionamiento del stack como ya se dijo se basa en la estructura LIFO ('Last in first out'), en donde los datos son colocados en el stack a través de los registros del tope del stack ('TOS'), cuando un operando es retirado del tope del stack es colocado en el stack de memoria y el contenido del registro 'S' es incrementado en uno, cuando un operando es colocado en el tope del stack ('TOS') es traído el operando apuntado por 'S', y este es decrementado en uno. El registro 'S' siempre esta apuntando a la ultima palabra en el Stack. Si durante la ejecución de un proceso el registro 'S' asume el valor contenido en el registro 'BOS' se genera una interrupción de Hardware suspendiendo el proceso por 'Stack Underflow'. Asimismo si durante la ejecución el registro 'S' asume el valor del registro 'TOS' se genera una interrupción de Hardware por 'Stack Overflow'.

Palabras de Control B-6700.

1) Operandos. Son usados para representar información numerica y/o l6gica, estos pueden ser:

a) Precisi6n Simple. Ocupan una sola palabra de 48 Bits.

[50:3] = 0 0 0
[47:1] = no utilizado
[46:1] = signo del operando
[45:1] = signo del exponente
[44:6] = exponente
[38:39] = mantisa

b) Doble precisi6n. Utilizan dos palabras de precisi6n sencilla.

[50:3] = 0 1 0

c) L6gicos. Ocupan un solo Bit de una palabra.

[50:3] = 0 0 0
[0:1] = 1-Verdadero, 0-Falso.

2) Palabras de control. Las palabras de control son utilizadas como su nombre lo indica para el control de la ejecuci6n de un proceso. Estas pueden ser:

a) 'Word Data Descriptor'.- Esta palabra referencia 6reas de datos, incluyendo buffers de entrada salida.

b) 'String Descriptor'.- Referencias a cadenas de 4,6 6 8 Bits.

c) 'Segment Descriptor'.- Referencia a un segmento de programa (c6digo).

d) 'Mark Stack Control Word'.- Es usada para mantener el control dentro del stack y proveer una historia de las entradas y salidas de los procedimientos o bloques.

e) 'Program Control Word'.- Es utilizada para la entrada a procedimientos.

f) 'Return Control Word'.- Es utilizada para la salida de los procedimientos o bloques.

g) 'Indirect Reference Word'.- Referencia a una variable dentro del ambiente correspondiente de direccionamiento, a trav6s del 'Address Couple'. La referencia es relativa al registro 'D' especificado en el 'Address Couple'.

h) 'Stuffed Indirect Reference Word'.- Referencia a una variable fuera del ambiente corriente de direccionamiento. La referencia es relativa a la base del stack en el cual la variable está localizada.

Sistema Operativo.

El sistema operativo en el equipo B-6700 se llama MCP (MASTER CONTROL PROGRAM) es el encargado de administrar los recursos de la máquina. El MCP está formado de los siguientes módulos:

- 1) CONTROLLER
- 2) WORK FLOW LANGUAGE
- 3) JOBFORMATTER
- 4) MAINTENANCE
- 5) SORT
- 6) MCP

Las funciones generales de cada uno de los módulos son:

CONTROLLER

Es el encargado de recibir los comandos del operador y ejecutarlos, también despliega en la pantalla del operador los mensajes del sistema, una característica de este módulo es que siempre se encuentra en ejecución; cuando sucede lo contrario es necesario dar HALT/LOAD.

WFL

El lenguaje de flujo de trabajos (WFL), es el medio por el cual es posible indicar y describir al sistema BURROUGHS, cuales trabajos (JOBS) deberá de ejecutar y en qué secuencia.

Este lenguaje permite al usuario describir cada trabajo como un conjunto de tareas relacionadas entre sí, y que serán ejecutadas. Es posible definir además la ejecución serie y/o paralelo de los trabajos especificados.

WFL es un compilador; el cual produce código para controlar las tareas que el usuario describe en los "JOBS".

El compilador de WFL realiza las siguientes funciones:

- a) Detecta errores sintácticos de un programa en WFL.
- b) Genera código para manejar las tareas que constituyen un "JOB" que se especifican en él.
- c) Genera el archivo en disco necesario para la ejecución del "JOB".

JOBFORMATTER

Se encarga de imprimir los encabezados .

MAINTENANCE

Es un paquete de pruebas de mantenimiento que puede ejecutarse en línea.

SORT

Se encarga de ordenar los archivos de la manera que se le indique. Si se invoca al MERGE es posible mezclar archivos.

Master Control Program.

Es el administrador del sistema propiamente dicho, él es el encargado de asignar la memoria; iniciar, terminar o suspender tareas; asignar prioridades, etc.

Existen procedimientos del MCP que siempre deben estar en memoria , éstos ocupan las localidades bajas. A tales procedimientos se les denomina residentes.

Las obligaciones del SISTEMA OPERATIVO respecto a la entrada/salida son:

- 1) Construir las palabras de control necesarias para realizar una operación física de entrada/salida
- 2) Iniciar las operaciones de entrada/salida.
- 3) Esperar la terminación de la entrada/salida.
- 4) Terminar la entrada/salida y notificar al JOB para que éste continúe
- 5) Manejar los errores físicos de la entrada/salida.
- 6) Reintentar la entrada/salida si es posible.
- 7) Entrar a la rutina de error del usuario si es declarada o descontinuar el programa.

Por otra parte el SISTEMA OPERATIVO provee al usuario con las siguientes funciones básicas para la entrada/salida.

- 1) Satisfacer las demandas de open y close de los programas.
- 2) Actualizar y checar todos los atributos de archivo.
- 3) La creación y actualización del lugar donde se almacena la información correspondiente a los atributos de archivo y la localización actual de los buffers.
- 4) Switcheo en los caracteres de entrada/salida en todas las cintas magnéticas.
- 5) Lectura y escritura de etiquetas especiales declaradas para los usuarios.
- 6) Formateo
- 7) Bloqueaje y desbloqueo de los registros si es requerido.
- 8) Traducción de caracteres si es requerido.

Intrinsecos.

Los intrínsecos son rutinas de uso común, que facilitan el trabajo de programación, tales procedimientos pueden ser llamados desde cualquiera de los lenguajes con que cuenta la máquina. Los INTRINSECOS siempre deben estar cargados en el sistema en un archivo de intrínsecos que se forma de tres partes, las cuales se explican a continuación:

1) Intrinsic Information Table.

Es una tabla en la que hay información para cada intrínseco, ésta es usada por los compiladores para establecer la cadena entre el programa y el intrínseco que se hace referencia en el mismo.

2) Intrinsic Stack.

Es un stack que se encuentra siempre presente, contiene palabras PCW que apuntan hacia donde está el código del intrínseco referenciado.

3) Intrinsic Number Table.

Es una tabla, en la que se le asocia un número a cada intrínseco y contiene la información necesaria para determinar si éste es del sistema o local.

Es importante hacer notar que sólo puede estar presente en el sistema un archivo de intrínsecos.

A partir de la versión 3.1 del SISTEMA OPERATIVO, se introdujo el concepto de LIBRARY. Estos son encadenamientos de un programa con una biblioteca de programas. Un LIBRARY puede invocar a otros, pero el sistema verifica que tales invocaciones no formen cadenas cíclicas. Los LIBRARY presentan varias ventajas:

- a) Pueden ser programados en cualquier lenguaje.
- b) Pueden ser del sistema o locales.
- c) Pueden estar corriendo varios al mismo tiempo.

Cada LIBRARY genera un stack, que puede ser "PUBLICO" o "PRIVADO". En el primer caso el stack puede ser usado por cualquier número de usuarios.

Compiladores.

Los compiladores con que cuenta el sistema B6700 SON:

ALGOL (DMALGOL,DCALGOL)
COBOL (DMCOBOL)
FORTRAN
PL/I
BASIC
NEWP
NDL
ESPOL
PASCAL

Utilerías.

Las utilerías, son programas de uso común, que son provistos por el vendedor del equipo y son principalmente para el manejo de archivos entre los dispositivos periféricos y para manejo interno del sistema. Entre las más importantes se pueden mencionar a:

1) SYSTEM/BACKUP .- El funcionamiento de las impresoras y perforadoras de tarjetas es lento respecto a la velocidad con que el procesador lleva a cabo sus operaciones, no obstante, un programa puede tener "control" de tales periféricos, así como de otros recursos del sistema.

Cuando se utiliza la técnica de BACKUP, un programa puede solicitar una impresora o una perforadora de tarjetas, para estos casos, el sistema operativo le asigna dispositivos que tienen una mayor velocidad - si fueron solicitados- (disco, cinta o pack). Estos simulan a los anteriores, y así el programa escribe en ellos "lógicamente". Cuando aquellos dispositivos "lentos" los ha liberado otro proceso, será posible enviar a ellos la información que quedó grabada en los periféricos veloces (utilizando el SYSTEM/BACKUP) de esta manera se logra optimizar tiempo en la máquina.

2) SYSTEM/BLOCKCHAR.- Sirve para imprimir letreos.

3) SYSTEM/CARDLINE.- Se utiliza para imprimir información en tarjetas en BCL, EBCDIC y BINARIO.

4) SYSTEM/COMPARE.- Sirve para comparar uno o más pares de archivos, la comparación se lleva a cabo bit por bit en cada registro.

5) SYSTEM/DCSTATUS.- Se utiliza para desplegar información del estado del subsistema de comunicación de datos.

6) SYSTEM/DUMPALL.- Por medio de esta utilería es posible realizar copias de archivos de un dispositivo a otro haciendo modificaciones, si se desea.

7) SYSTEM/DUMPANALYZER.- Este es un programa que analiza el estado de la memoria.

8) SYSTEM/FILEDATA.- Se utiliza para obtener directorios y mapas del disco o pack.

9) SYSTEM/GUARDFILE.- Se utiliza para definir la seguridad de los archivos. Por medio de él se define que usuarios y que programas tienen acceso a que archivos, también es posible definir el tipo de acceso.

10) SYSTEM/LOGANALYZER.- Utiliza la bitácora del sistema (SYSTEM SUMLOG) para obtener reportes de mantenimiento.

11) SYSTEM/LOGGER.- Con él es posible obtener reportes que permiten contabilizar los recursos utilizados en un periodo deseado.

12) SYSTEM/MAKEUSER.- Ya que el acceso al sistema se hace por medio de claves, éstas deben ser dadas de alta, baja o modificadas, para ello es posible utilizar esta utilidad.

CANDE (Command and Edit)

CANDE es un MCS (Message Control System) proporcionado por BURROUGHS, está escrito en DCALGOL; el propósito específico de este MCS es el comunicar a un usuario situado en una terminal remota, con la computadora central.

Se disponen de comandos por medio de los cuales es posible:

- Edición y corrección de archivos.
- Ejecución de programas.
- Lanzado de trabajos.
- Consulta de el estado de una cierta línea.
- Consulta de los recursos de máquina que un programa ha consumido hasta un determinado momento.
- Reconfigurar la red dinámicamente.
- Investigar los recursos de disco consumidos.
- Comunicarse con otro usuario conectado a la red de teleproceso.
- Descontinuar trabajos lanzados por el usuario.

Memoria.

La memoria del sistema B6700 consta de módulos de 16K palabras y puede irse incrementando de acuerdo a las necesidades del usuario.

Cada palabra de memoria consta de 60 bits, de los cuales, 48 bits son para almacenar información, 8 bits para hardware, 3 bits de control y 1 bit de paridad. Los tres bits de control se usan para identificar a que tipo de palabra pertenece, y además proveen de protección a la memoria, describen el tipo de dato que contienen e información de control adicional. Para direccionar la memoria se utilizan 20 bits, que nos dan una capacidad de direccionar 1,048,576 localidades de memoria. El bit de paridad es usado para checar la validez del almacenamiento y transferencia de la información.

Características de la Memoria.

1.- Completamente Dinámica. Si se requiere un área de memoria se asigna, si no es ocupada el área se desasigna. No necesariamente está el código en localidades contiguas de memoria, puede estar disperso y por medio de apuntadores se liga.

2.- Memoria Virtual. Cuando se manejan programas o códigos muy grandes, sólo se tiene una parte en memoria principal, el resto se encuentra en disco, y éste se considera como una extensión de la memoria principal.

3.- Areas de Longitud Variable. No hay particiones fijas de la memoria. Se asigna el área que se necesite.

4.- Workings-Set. Es la cantidad mínima de segmentos de código que deben residir en memoria para que el programa se ejecute eficientemente. Este número es calculado al compilar el programa.

5.- Protección de Memoria. El sistema se vale de varios mecanismos para evitar que un programa accese memoria que no tiene asignada.

Las 'Memory Links', son palabras que le indican al MCF si el área está ocupada, disponible ó asignada.

El sistema operativo estructura la memoria de tal manera que, a cada área se le asocian dos grupos de palabras de control (antes y después de la misma), tales palabras forman la estructura de listas que mantiene el sistema. El sistema operativo forma dos tipos de listas, una para áreas en uso y otra para las áreas disponibles.

Para las áreas en uso se utilizan cuatro palabras, tres de éstas la preceden (LINKA, LINKB, y LINKC), y una la sigue (LINKZ).

Para las áreas de memoria disponible, también se tienen cuatro palabras asociadas a cada una, para este caso, se usan dos palabras antes (AVAILA y AVAILB), y dos la siguen (AVAILY y AVAILZ).

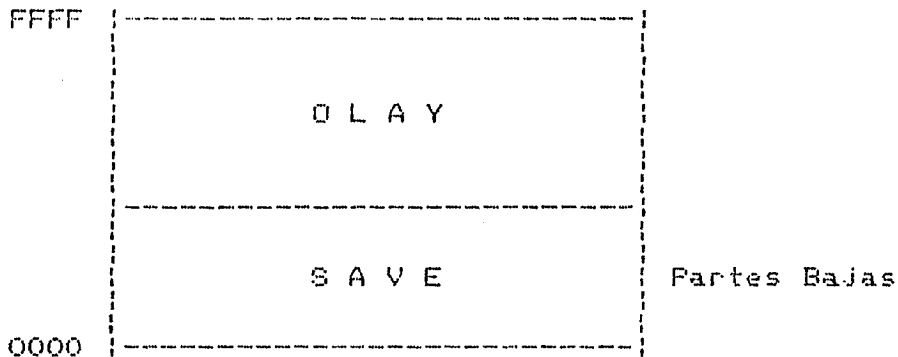
+	+	+	+	+	+	+	+			
	AVAILA		AVAILB		área disponible		AVAILY		AVAILZ	
+	+	+	+	+	+	+	+			

Cuando el área está en uso, la rodean otras 4 palabras:

+	+	+	+	+	+	+	+			
	LINKA		LINKB		LINKC				LINKZ	
+	+	+	+	+	+	+	+			

Si se desocupa un área consecutiva, la concatena a la anterior, quedando con un solo juego de palabras AVAIL.

La memoria principal se divide en dos partes: la memoria SAVE y la memoria OLAY.



Memoria SAVE .- Es aquella que tiene que permanecer residente, generalmente está ocupada por algunos procesos del sistema operativo, así como sus stacks de trabajo.

Memoria OLAY .- Aquí reside toda la información que se puede mandar a disco cuando se necesite espacio.

La división de la memoria en estas dos partes es dinámica como se mencionó anteriormente.

La lista de áreas disponibles SAVE empieza en la palabra "1" de memoria, y están ordenadas de la más pequeña a la más grande. La razón para esto es que, se asigna el menor tamaño de memoria suficiente.

La lista de áreas disponibles OLAY empieza en la palabra "0", y están ordenadas de manera que el último segmento que se desocupó es el primero en la lista. Si el área que se necesita es menor que la disponible, se desencadena la necesaria y se modifican los apuntadores del área restante.

Existe un procedimiento del sistema operativo que se encarga de poner como disponibles las áreas que se van desocupando. Cuando se desocupa un área, se cambian los apuntadores y despues se borra la información del stack.

La máxima capacidad de memoria para este equipo es de 32 módulos de memoria, 'CDRE' de tipo Planar con un ciclo de 1.6 microsegundos. Cada módulo tiene una capacidad de 16 KW (96 Kbytes), lo que nos da una capacidad de almacenamiento en memoria principal de 3072 Kbytes.

Manejo de Memoria.

La computadora B6700, al igual que las otras de su serie, realiza el manejo de memoria utilizando la técnica de "Memoria Virtual", para ello se basa en la segmentación de los programas, tales segmentos son de longitud variable. La segmentación de los programas se realiza al momento de la compilación, y difiere para cada uno de los compiladores con que cuenta el sistema: en Algol, realiza la segmentación a nivel de bloques; en Fortran, lo hace a nivel de subrutinas; y en Cobol a nivel de secciones.

Al dividir los programas de esta manera, es posible:

- 1) Eliminar los segmentos de Programa innecesarios en memoria.
- 2) Solicitar solamente la memoria suficiente para contener un segmento de Programa.

Por tanto, con la segmentación de los programas se logra tener presente en memoria principal, sólo aquellos segmentos que requiere la tarea que se está ejecutando.

A cada segmento de Programa, se le asocia una palabra de control denominada SEGMENT DESCRIPTOR (Descriptor de Segmento), que en su conjunto forman un diccionario de

segmentos de programa, estas palabras, así como los segmentos no utilizados, se mantienen en disco mientras no se ocupan. Es frecuente que un programa utilice áreas de datos (arreglos), a los cuales también se les asigna una palabra de control, en este caso, la palabra se llama DATA DESCRIPTOR (Descriptor de Datos) a cada arreglo utilizado se le asocia una palabra de este tipo.

Los bits denominados "TAG" (48-50) son utilizados como ya dijimos, por el sistema, con el objeto de proteger las áreas de datos y para poder determinar el tipo de palabra con que se está trabajando, es decir, si se trata de un dato, un descriptor o alguna otra palabra de control.

Las áreas de datos que se exceden en tamaño a las 1024 palabras (1K word), se segmentan automáticamente en bloques de 256 palabras, para prevenir al sistema de demandas de grandes áreas de memoria contigua. Para este caso, el descriptor del arreglo tendrá el bit correspondiente prendido, indicando que el arreglo que describe está segmentado. El bit de segmentación indica que el descriptor describe a un "VECTOR DOPE".

Los descriptores de datos, no solamente describen áreas de datos, sino que también pueden referenciar a otros descriptores de datos, este es el caso de un arreglo multidimensional utilizado en Algol, para ello el descriptor del arreglo apunta a un área que contiene más descriptores de datos, cada uno de ellos representa un valor índice de la primera dimensión del arreglo. Tales áreas se les denomina "VECTOR DOPE", así, cada uno de los elementos de este vector describe a un renglón del arreglo; también puede describir a otro "Vector Dope".

Asignación del Espacio de Memoria.

Las áreas de memoria en uso pueden ser de dos tipos: SAVE y NON-SAVE. Las primeras (SAVE), deben estar residentes desde el momento de la asignación, (cuando las requiere el programa), hasta la desasignación (cuando ya no se ocupan).

Las áreas non-save, pueden ser transferidas de memoria a disco por el sistema operativo (Proceso de OVERLAY). Tales áreas se llevan automáticamente a memoria cuando son referenciadas.

El sistema operativo transfiere la información de memoria non-save a disco, siempre y cuando se trate de datos, o reestructura el descriptor del segmento del código, indicando que tal porción del programa la debe buscar en disco. En ambos casos, esas áreas se llevan automáticamente a memoria cuando son referenciadas.

Manejando la memoria de esta manera (áreas de memoria save y non-save mezcladas), se puede caer en problemas denominados "chequeo de límites". Los problemas de este tipo, pueden ser evitados de dos maneras:

- 1) Agrupando las áreas save en las partes bajas de la memoria, mediante el uso de dos listas de áreas de memoria: lista de áreas de memoria save y lista de áreas de memoria overlay. La primera lista lista todas las áreas disponibles que siguen inmediatamente a un área save, ordenadas por el tamaño del área. La segunda lista, lista todas las áreas disponibles que colindan con el área overlay, ordenadas de tal manera que la última en entrar es la primera en ocuparse (LIFO). Todas las áreas están referenciadas en una o en ambas listas.

2) Fragmentando la memoria en áreas pequeñas, junto con esto es necesario prohibir demasiadas solicitudes de segmentos pequeños de memoria. Cuando se recibe una solicitud de asignación de memoria, se examinan las listas de áreas disponibles hasta encontrar una del mismo tamaño ó una que no exceda a la solicitada en un factor delta ($\text{delta}=15$ palabras). El tamaño de delta se pone en el LINKB y el bit indicador de delta se prende en el LINKZ. Las palabras delta siempre siguen a las porciones en uso del área.

Solicitudes de Memoria.

Sucede una solicitud de memoria, cuando se encuentra ejecutando una tarea la cual referenciaba a un descriptor (puede ser de datos o de código), que tiene el PRESENCE BIT apagado. Tal referencia causa una interrupción al sistema. El encargado de atender esta señal es el hardware del sistema, el cual construye un "NON-PRESENT COPY DESCRIPTOR" con la dirección del MOM DESCRIPTOR, después de esto forza una entrada al sistema operativo para que la rutina encargada de esta tarea realice lo siguiente:

- Determine si la interrupción es de PRESENCE BIT.
- Llame a la rutina PRESENCEBIT.

Una vez que el código de la rutina PRESENCEBIT se está ejecutando, se garantiza que ningún otro stack de trabajo genere otra interrupción de este tipo (asíncrona). Esto lo logra cerrando el MOM DESCRIPTOR (es decir prende todos los

bits del campo de dirección del mom-descriptor ausente) desde luego, para poder realizar esta operación, se debe checar antes que no se encuentre en estado "cerrado". En caso de que se genere otra interrupción de PRESENCE BIT, la tarea que la generó se suspende hasta que la tarea que inició la primera interrupción termine.

La suspensión de la tarea va acompañada por una espera en el sistema operativo llamada "WAITING ON AN EVENT" , la cual es originada cuando la acción de PRESENCE BIT actual termina, así, la tarea que fue interrumpida checará, cuantas veces sea necesario, el descriptor MOM para asegurar que está ausente, si esto ocurre, genera nuevamente la interrupción PRESENCE BIT, y el proceso se repite nuevamente.

Después de que el descriptor MOM se hace presente, se realiza una búsqueda en el stack de la tarea, en ese momento todos los descriptors copia que corresponden al descriptor MOM se pasan a un estado presente, y su campo de dirección se cambia, conteniendo ahora la dirección del área de memoria que se ha hecho presente.

Es conveniente hacer notar, que la primera interrupción de este tipo en los arreglos de datos, origina que el área sea asignada y puesta en ceros, esta acción la realiza el sistema operativo.

En el caso de la comunicación entre programas (IPC), los descriptors copia que residen en los stacks de las otras tareas, se hacen presentes hasta que son referenciados por la tarea.

Proceso de Overlay.

En el punto anterior, se ha explicado la manera en que el sistema procede cuando recibe una solicitud de memoria; no hay que olvidar, que al trabajar en un ambiente de multiprogramación, normalmente se están ejecutando dos, tres o más tareas, por lo que es fácil suponer que un momento dado, la lista de memoria disponible ya no pueda satisfacer a una solicitud más, debido a que la cantidad solicitada excede a todas las áreas disponibles que en ese momento se tienen. Esta problemática la resuelve el sistema mediante una técnica denominada OVERLAY.

Puesto que los segmentos de un programa, pueden ser de código o de datos, habrá dos maneras diferentes de llevar a cabo el overlay. En primera instancia se hará referencia a los segmentos de código, y en seguida se mencionará como son tratados los segmentos de datos.

Los segmentos de código no necesitan ser escritos en disco, ya que la información que se tiene en memoria es una copia del código residente en él, y además el código no es modificado al pasar por memoria. Las acciones que se realizan para este caso son:

- 1) Marcar el segment descriptor en el bit correspondiente a "COPY BIT", como no presente.
- 2) Modificar el campo de dirección para que apunte al MOM DESCRIPTOR, este a su vez contiene la dirección en disco del segmento de código a eliminar.

Para efectuar, el proceso de overlay se examinan las áreas de memoria para determinar si son áreas de "OVERLAY" si este es el caso, el campo correspondiente para este control

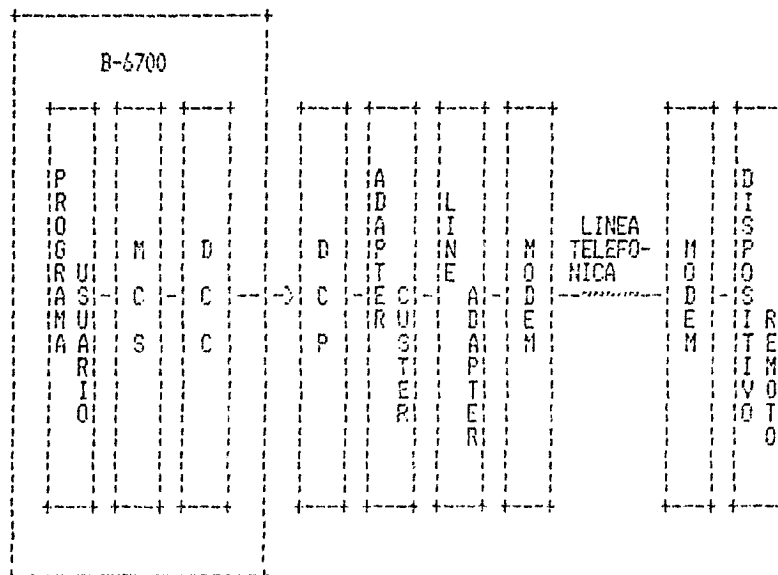
(en el memory link) se prende, ello indica que el área se encuentra en proceso de overlay. En este momento el sistema operativo detiene los otros procesos que están corriendo, y trata de encontrar otra área de memoria que satisfaga a la que se le aplicará el proceso de "OVERLAY" es decir, trata de hacer overlay de memoria a memoria. Si el área va a ser afectada por este tipo de proceso, el sistema operativo busca sus propios stacks de tareas y modifica los descriptores MOM y las copias de éste si es el caso, para que apunten a una nueva dirección del área de memoria. Cuando el área no puede ser relocalizada en memoria, se realiza una búsqueda en el stack de todos los descriptores que hacen referencia a esta área. La copia de los descriptores MOM se ponen como no presentes y se cambia su campo de dirección para que apunten al descriptor MOM, éste es "cerrado" y marcado como no presente. Las áreas de memoria que fueron hechas disponibles, se escriben en sus propios archivos de overlay para tareas (si se trata de datos) y finalmente se hacen disponibles al solicitante original.

Cada tarea tiene un archivo de overlay único, asignado al momento de iniciar la ejecución, sobre el cual se escriben las áreas de memoria que contienen información que puede ser relocalizada. El espacio ocupado por el archivo de overlay, permanece asociado al área de memoria hasta el momento en que el descriptor MOM, contenido en un bloque de código, se elimina, es decir, hasta que el área se desocupa, ó también, si el tamaño del área de memoria es modificado por el usuario (arreglos dinámicos de asol). Cuando el archivo de overlay ya no va a ser usado por más tiempo, se hace disponible para que sea utilizado por una nueva tarea. Asociado con cada archivo de overlay, se tiene un arreglo que se denomina OLAYINTD, éste tiene una máscara con un bit relacionado con un segmento de disco disponible, dentro del archivo de overlay. Este arreglo se utiliza para controlar la asignación y desasignación del espacio en disco.

3.3.3 Software de Comunicación de Datos.

Los elementos de software involucrados en la comunicación de datos son los siguientes:

- i) Programa que define la red de teleproceso el cual esta escrito en el lenguaje NDL (Network Definition Language).
- ii) El compilador de NDL.
- iii) El controlador de comunicaciones DCC (Data Communications Controller).
- iv) Programa de control del DCP.
- v) Sistema Controlador de Mensajes MCS (Message Control System).
- vi) Compilador de DCALGOL (Data Communications Algol).



SUBSISTEMA DE COMUNICACION DE DATOS.

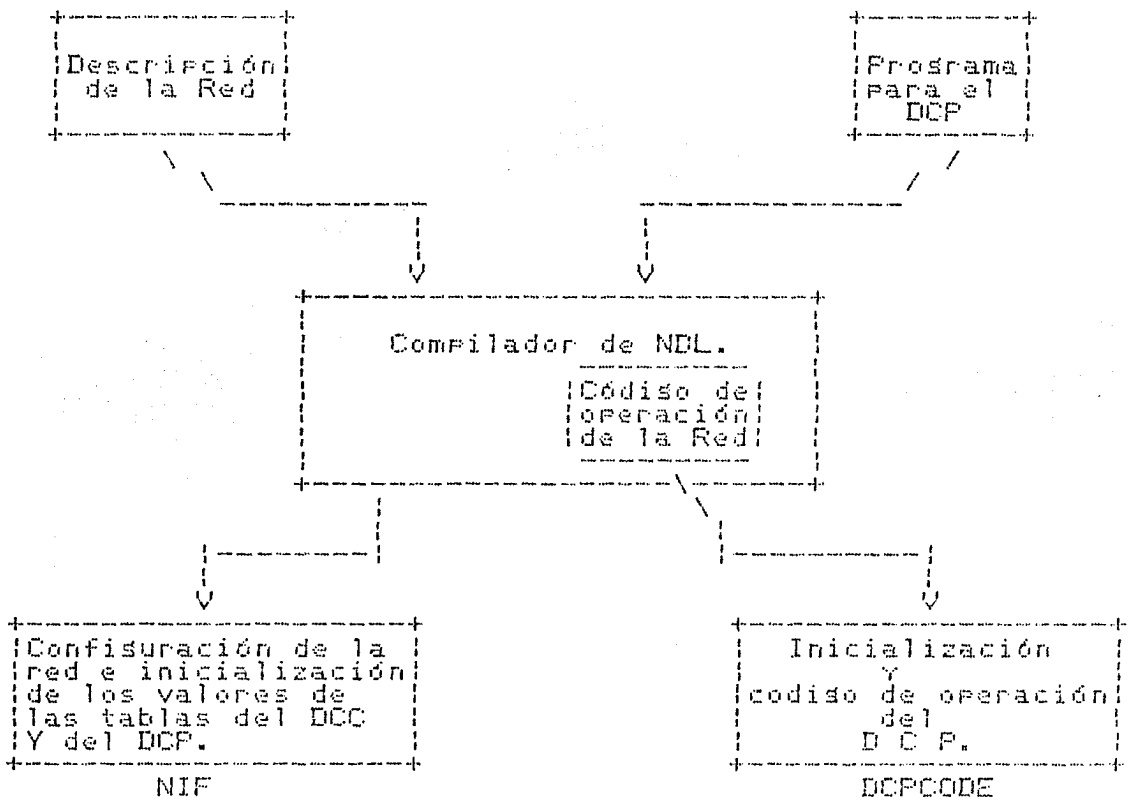
Lenguaje de Definición de la Red.

El programa fuente del lenguaje de definición de la red (NDL), es el que describe la red de comunicación de datos (DATACOM), física, lógica y funcionalmente. Este programa esta escrito en lenguaje NDL y es usado por el compilador de NDL, para crear las tablas de datos, que contienen las características físicas y lógicas del sistema de DATACOM, además de un conjunto de instrucciones que envuelven procedimientos para el manejo eficiente del sistema de comunicación de datos. El programa en NDL deberá describir el número y tipos de dispositivos remotos que son conectados a la red, el número y tipos de líneas de transmisión involucradas y el equipo usado para la interface de la red con el procesador central. Además se describen las prioridades de los dispositivos remotos, así como también la descripción del tipo de comunicación (dentro del diálogo) con el sistema central, y que acciones son tomadas en caso de ocurrir alguna circunstancia anormal. A continuación se presenta una tabla con los elementos de un sistema de comunicación de datos (DATACOM):

Elementos de la Red	Características Físicas	Características Lógicas
DCP's	Tamaño de la memoria capacidades de reconfiguración.	Grupo de terminales controladas por cada DCP.
LINES	Localización física Velocidad de transmisión tipo de línea y conexión.	Asignación de líneas a la estación capacidad de respuesta automática.
MODEM'S	Retardos físicos tipo de transmisión y velocidad.	Nombre simbolico
STATION'S	Características terminales.	Nombre simbolico Atributos lógicos MCS asociado.
TERMINAL'S	Código de transmisión Velocidad y tipo de paridad.	Numero de transmisiones caracteres especiales.

El compilador de NDL es un programa que compila el código fuente de NDL y como resultado de ésta compilación produce lo siguiente:

- a) El código objeto para el DCP (DCPCODE).
- b) Tablas usadas para el Sistema Central y el DCP NIF (Network Information File).



DESCRIPCIÓN DE UNA RED DE COMUNICACIÓN DE DATOS.

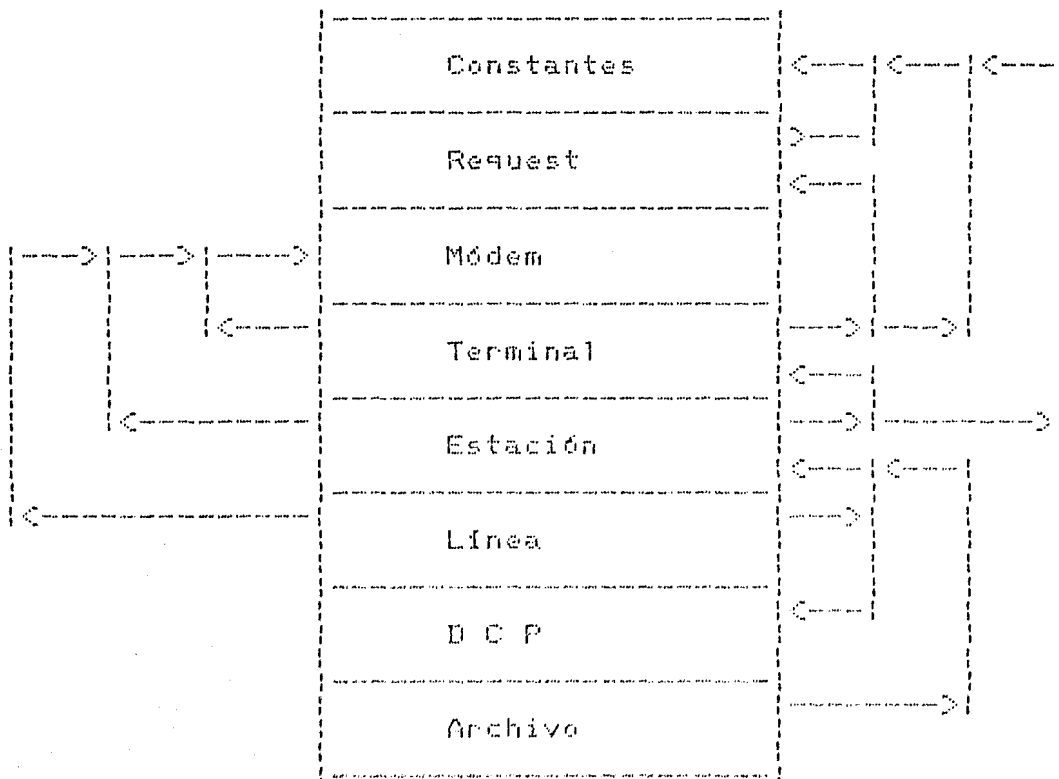
El programa escrito en NDL consta de varias secciones para describir las características físicas y lógicas de la red. La información provista en estas secciones es utilizada en parte, para proveer al DCC con las tablas y el código del DCP que es usado para operar las tablas a continuación se enumeran dichas secciones:

- a) Constantes.
- b) Request.
- c) Módem.
- d) Terminal.
- e) Estación.
- f) Línea.
- g) DCP.
- h) Archivo.

El compilador de NDL hace un chequeo a través de las varias secciones y definiciones para estar seguro que la red esta estructurada lógicamente. Por ejemplo una línea no puede estar asociada con un Módem en particular si el rango definido de velocidad y tipo de transmisión de el Módem no permiten una interface propia con la línea. Similarmente, una terminal definida para operar en forma asíncrona no puede estar asociada con una línea que usa un adaptador síncrono.

Toda la información provista por las definiciones del programa fuente de NDL son grabadas dentro de un archivo NIF. Esto permite a algun MCS ó a un programa de usuario tener acceso a muchas de las características lógicas de la red y tambien permiten una reconfiguración dinámica de la red por un MCS.

Existen lizas lógicas entre las diferentes secciones de la red; estas relaciones son:



Data Communication Controller.

El DCC es la parte del sistema operativo que se encarga de la comunicación de datos. Esta es la interface básica entre el DCP y el sistema principal. Existe como un subconjunto básico del MCP (Master Control Program), y opera independientemente en proceso o en stack. En un proceso por cada DCP activo.

Cada estación está controlada por un sistema controlador de mensajes (MCS) a la vez. Varias estaciones pueden ser controladas por diferentes MCS's. Todos los movimientos de información entre el sistema central y el DCP son controlados por el DCC. El DCC actúa como intermediario entre los MCS's y los programas de aplicación.

El DCC también es el encargado de checar la validez de las instrucciones pasadas por los MCS's a los DCP's.

Código objeto del DCP.

El código objeto del DCP es un sistema operativo para el DCP, las funciones de éste dependen del programa fuente NDL; algunas de estas funciones son:

- a) Operaciones de multiplexeo de líneas.
- b) Multiplexeo de las peticiones del sistema.
- c) Control de líneas.
- d) Recolección y distribución de caracteres.
- e) Detección de errores.

Las funciones anteriores llamadas definiciones de control y definiciones de pedido, son procedimientos que son ejecutados por el DCF cuando requiere efectuar alguna disciplina de línea. Cada línea debe tener asociado una definición de control, y cada terminal puede tener asociada una ó más definiciones de pedido.

Las definiciones de control y pedido consisten en instrucciones (programa fuente NDL) que el compilador transforma en instrucciones al DCF para ser ejecutadas cuando se requiera. Un 'Receive request' es invocado cuando una entrada desde una terminal es procesada y una definición de 'Transmit request' es ejecutada cuando una salida hacia una terminal ha sido requerida por un MCS o un programa de usuario. La definición de control de línea es utilizada para determinar cuando y para cual de las estaciones en la línea de una definición de 'Request' va a ser ejecutada.

Control de Línea.

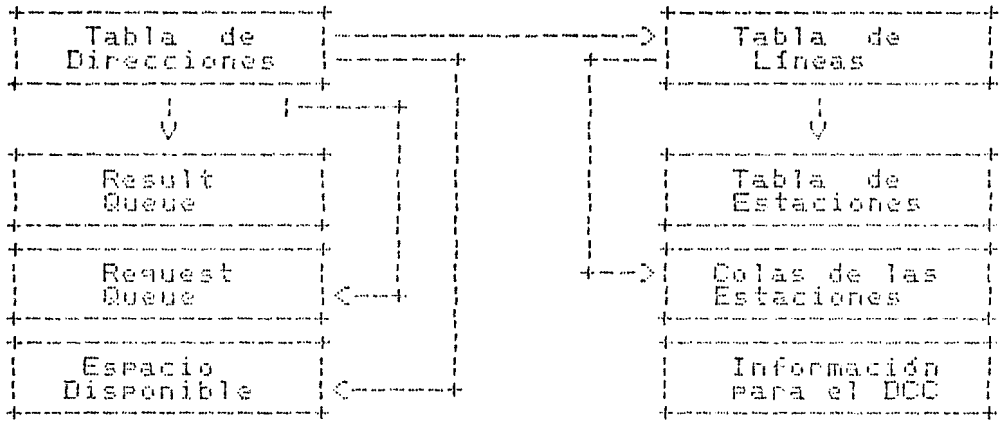
Cada definición de control o procedimiento de control de línea, debe realizar dos funciones: primero debe seleccionar cual estación en la línea debe recibir atención, y segundo debe decidir que función en particular sera efectuada para esa estación. Si la función a realizar es una petición de salida, el control es transferido a el 'Transmit Request' para dicha estación, si la función es una operación de entrada el 'Receive Request' de la estación es ejecutado las funciones de la red que no envuelven la recepción ó transmisión de mensajes, tales como cambios en el estado de la red son efectuados por medio de llamados a un subprograma común ó macro, dentro del sistema operativo del DCF mismo.

Tablas del DCP.

El compilador de NDL construye una serie de tablas que contienen las características físicas y lógicas de la red tal como fueron definidas en el programa fuente de NDL. El DCP utiliza estas tablas para guardar el estado y la información pertinentes para determinar que tipo de funciones serán ejecutadas por cada uno de las diferentes líneas y estaciones que el DCP controla.

El compilador pone una imagen de disco de estas tablas dentro del archivo del DCPCODE. Cuando el DCP es iniciado, las tablas son cargadas dentro de la memoria principal por el DCC el cual también provee al DCP las referencias para las tablas. Si existen varios DCP'S que comparten recursos entonces estos pueden utilizar el mismo grupo de tablas si la descripción de la red indica este modo de operación.

Cada grupo de tablas puede ser dividido en dos grupos de información. Cada línea tiene una tabla y cada estación también. El DCP utiliza un descriptor de línea para referenciar cada tabla de línea. Los descriptores para todas las líneas controladas por el DCP, son guardadas dentro de un vector el cual es indexado por las direcciones físicas de los adaptadores de línea. Cada descriptor de línea contiene información concerniente a el estado de la línea (Not ready, Connected, Busy, etc.) y una referencia para la definición de control que es usada para la línea contiene la dirección de memoria para la actual tabla de la línea. También existe un vector similar para los descriptores de estación.



Tablas y colas que se generan al comenzar el proceso de comunicación de datos.

Message Controller System.

Un MCS es un programa de propósito especial escrito en DCALGOL (Data Communications ALGOL), que puede ser proporcionado por el fabricante (por ejemplo: SYSTEM/CANDE, SYSTEM/RJE Y SYSTEM/DIAGNOSTICSMCS), ó bien puede ser escrito por los usuarios. Un MCS, como su nombre lo implica, controla el flujo de mensajes de comunicación de datos entre la terminal y el sistema principal.

La información de el DCP, tal como la entrada de una terminal e información del estado, es dirigida al MCS a través del DCC. Los mensajes del MCS al DCP, tales como salida de una terminal de la red de telerproceso, o cambios de red, son realizados por el MCS invocando una función intrínseca de la máquina llamada "DCWRITE". Cada estación que está definida por el programa objeto en NDL, deberá tener uno y sólo uno de los MCS que lo controle.

Un MCS puede controlar líneas a las cuales está asignado, en las cuales hay al menos una estación que controla el MCS particular.

Este control consiste entre otras cosas, en poner la estación lista para funcionar, y asignar a la estación un archivo de trabajo.

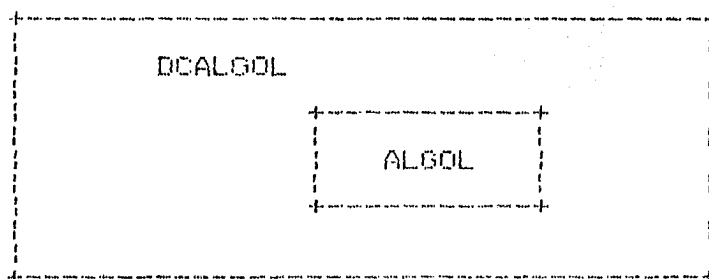
El lenguaje DCALGOL incluye todas las características del ALGOL del equipo B-6700 además de contener declaraciones adicionales para el manejo de colas y mensajes, estas son:

- Message.
- Message Array.
- Queue.
- Queue Array.

Además de poder referenciar intrínsecos del sistema operativo para el manejo de las colas y los mensajes y para poderse comunicar con el subsistema de comunicaciones.

Los intrínsecos del DCALGOL son los siguientes:

- | | |
|------------|--------------|
| - ALLOCATE | - NULL |
| - INSERT | - SIZE |
| - REMOVE | - QUEUE INFO |
| - HOLD | - FLUSH |
| - ATTACH | - ON |
| - COMBINE | |

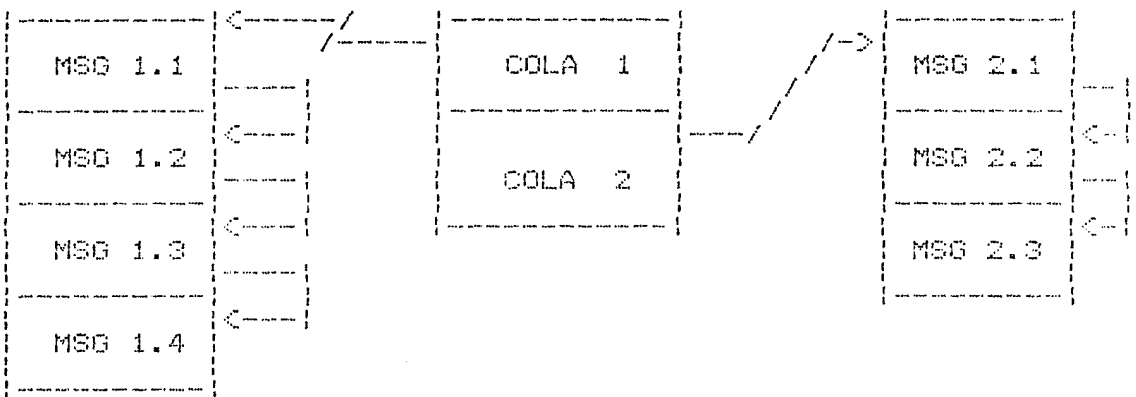


DCALGOL. VS. ALGOL.

Funcionamiento de la comunicación de datos del equipo B6700.

Los vehículos más importantes en la transferencia de información, entre los elementos que forman el sistema de comunicación de datos de la B6700, son los mensajes.

Los mensajes son áreas de memoria que se encuentran ligadas entre sí por apuntadores y forman en conjunto las colas.



ESTRUCTURA DE LOS MENSAJES EN B-6700.

Existen varias colas en el sistema. Cada DCP tiene un par de ellas que son la "Request Queue" y la "Result Queue".

En la "Request Queue" son puestos los mensajes que se mandan hacia los DCP's. En la "Result Queue" se obtienen los mensajes que llegan del DCP hacia el sistema central.

Cada estación tiene una cola que se llama "Station Queue", en donde se ponen todos los mensajes que van a esa estación. Cada MCS tiene una cola llamada "Primary Queue" y ninguna, o varias colas, llamadas "Current Queue".

Estas colas sirven para que el sistema de comunicación de datos se comuniquen con el MCS. De esta forma los mensajes forman la información que es transferida entre programas de aplicación y el MCS, programas de aplicación y el DCC, el DCC y el MCS, etcétera.

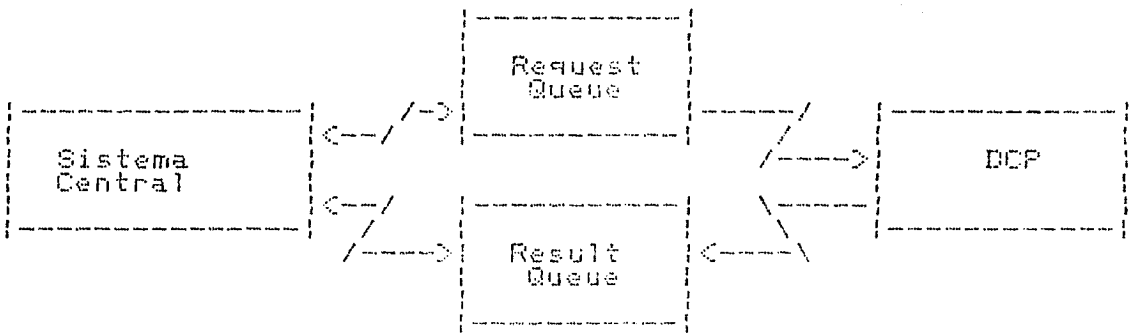


DIAGRAMA DE LA "RESULT QUEUE" Y LA "REQUEST QUEUE".

La transferencia de información no involucra necesariamente movimientos físicos, sino que al pasar un mensaje de un lugar a otro, solamente las cadenas o apuntadores de éste son cambiados.

Todos los mensajes constan de un número entero de palabras, de 6 caracteres y contienen un encabezado que en general es de seis (6) palabras, a continuación del encabezado está una parte de texto, que es el mensaje en sí.

Todas las transferencias de información en el sistema se

hacen a través de áreas de memoria, que en el caso de un programa en una escritura le pasa al MCP el dato en una área llamada "Program Buffer", lo mismo sucede al dar un "Read", el área es puesta en el archivo correspondiente por las llamadas de "I/O". Entonces el registro es puesto en el Program Buffer y el I/O intrínseco copia a éste dentro de la parte de texto de un mensaje y lo encadena a la cola apropiada. Si este es puesto en la "Request Queue" el "Buffer" permanece ocupado hasta que el DCP lo saca de la "Request Queue" para ponerlo en una terminal. En caso de que el MCS que controla la estación decida participar en las operaciones de I/O, este mensaje es pasado a una cola del MCS para que éste lo ponga en la estación correspondiente y pueda hacer trabajos de edición o toma de decisiones.

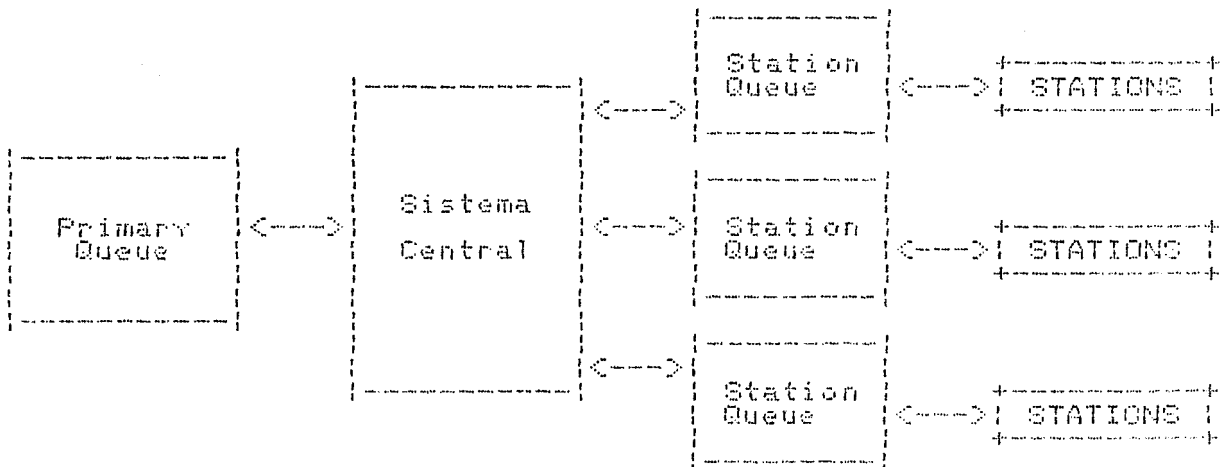


DIAGRAMA DE LA "PRIMARY QUEUE" Y LAS "STATIONS QUEUES".

El MCS, el DCP y los I/O intrínsecos se comunican entre sí a través de las colas.

Las denominadas "DCALGOOL Queue" y las colas de archivos remotos son de una misma estructura, mientras que las "Result

Queue", "Request Queue" y "Station Queue", son de otra estructura.

Las colas de archivos remotos y las de DCALGOL tienen un único descriptor de datos en el "Stack". Este descriptor de datos apunta a un conjunto de palabras contiguas llamadas "Hidden Messase", este mensaje contiene estadísticas sobre las cola tales como: número de mensajes en la cola y el número de palabras en memoria de todos los mensajes que contiene.

Cada DCP tiene su "Request Queue", de la cual extrae los mensajes y estas colas son el único medio por el cual los mensajes son transmitidos del sistema central a los DCP's y viceversa.

Cada mensaje debe ser enrutado a uno de los varios MCS's de la "Result Queue" siempre y cuando éste pertenezca a una estación bajo su control. Todos estos ruteos de mensajes los hace el DCC, que para simplificar sus funciones tiene un proceso asíncrono que sirve a cada DCP con un par de "Request Queue" para cada uno. La función más importante de este proceso es extraer información de la "Request Queue".

Algunas veces también inserta información en la "Request Queue", siendo éste más comúnmente hecho por los "I/O Intrinsic" del MCP y los "DCwrite Intrinsic" usados por el MCS para insertar mensajes en la "Request Queue".

La cola en la cual recibe un MCS información o comunicación del DCC es la "Primary Queue" y cada MCS al comenzar a funcionar tiene que iniciar esta "Primary Queue" para que el sistema tenga el medio de comunicarse con el MCS correspondiente. El MCS se comunicará a través de los "DCwrite Intrinsic", que son llamadas instrucciones de DCALGOL para ejecutar diferentes funciones de acuerdo a la información que se pase a estos intrínsecos.

Cuando se tecllea un carácter desde una terminal remota

del sistema, tienen lugar los siguientes eventos:

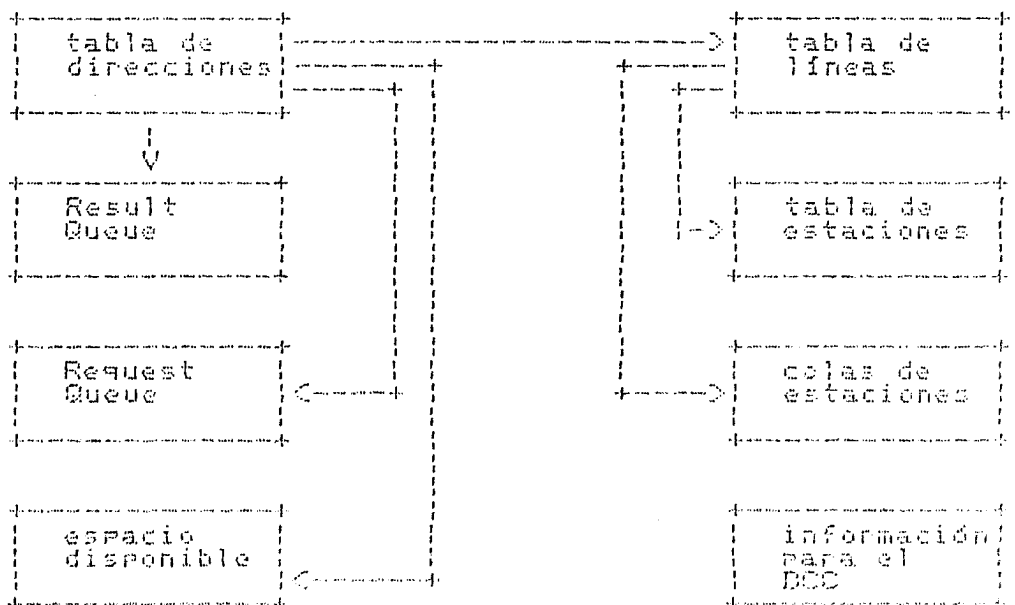
1. El teclado de la terminal remota envía una señal codificada al microprocesador de la terminal, indicándole que fue teclado un carácter y el código ASCII de dicho carácter. El CPU de la terminal (si lo tiene la terminal), envía este código en forma paralela al puerto encargado de transmitir este carácter en forma serial, y además convertirlo a niveles de voltaje lógicos de la interface RS-232C.
2. La señal serie se propaga por los conductores que forman un canal de comunicación entre la terminal y la computadora central.
3. El line-adapter detecta los niveles lógicos de tensión que se propagaron y llegaron a él en forma de señales RS-232C. Estos niveles de tensión son convertidos en bits.
4. El adapter-cluster recibe del line-adapter los bits que forman un carácter, y los acumula hasta que puede formar un carácter.
5. El DCP recibe los caracteres que le pasa el adapter-cluster, hasta acumular una palabra de 6 (seis) caracteres, a los cuales se les denomina mensaje.
6. Cuando se encuentra formado el mensaje, el software que se encarga de la supervisión y control de la red, previamente definido en NDL y que se encuentra corriendo en el DCP, "reconoce" el mensaje e investiga a que MCS está adscrita la terminal de donde proviene el mensaje, toda esta información es proporcionada al DCC.
7. El DCC checa la validez del mensaje y de las instrucciones que le pasa el DCP a el MCS respectivo. Si el mensaje es válido, lo pone en la "request queue" del MCS respectivo.
8. El MCS es el encargado de "reconocer" e interpretar la información proporcionada en el mensaje depositado en su "request queue".
9. Si se trata de algún comando para que el MCS lo realice, (como la asignación de un archivo en disco, la compilación de algún programa, etc.) ejecuta dicha instrucción si está capacitado para ella, si no, informará al usuario que se encuentra en la terminal remota, que no está capacitado para efectuar dicha operación.
10. Si se trata de un mensaje para algún programa que esté corriendo a través de este MCS, el mensaje es pasado al programa respectivo.

Para el caso en que el usuario que está operando la terminal remota, ejecute un programa que es interactivo, el proceso de comunicación de el programa con el usuario, será inverso al descrito en los incisos anteriores.

Proceso de Comunicación de Datos.

Al iniciarse la comunicación de datos en el equipo B-6700 se generan en memoria varias colas y tablas, que a continuación se listan:

- Bloque de direcciones para 'Request' y 'Result Queue', dirección de la tabla de líneas y de espacio disponible.
- Las 'Result' y 'Request Queue' para cada uno de los DCP's.
- Tabla de espacio disponible.
- Tabla de líneas.
- Tabla de estaciones.
- Colas de estación para cada una de las estaciones de la red.
- Tablas de información para el DCC.



TABLAS Y COLAS QUE SE GENERAN AL INICIAR EL PROCESO DE COMUNICACIÓN DE DATOS.

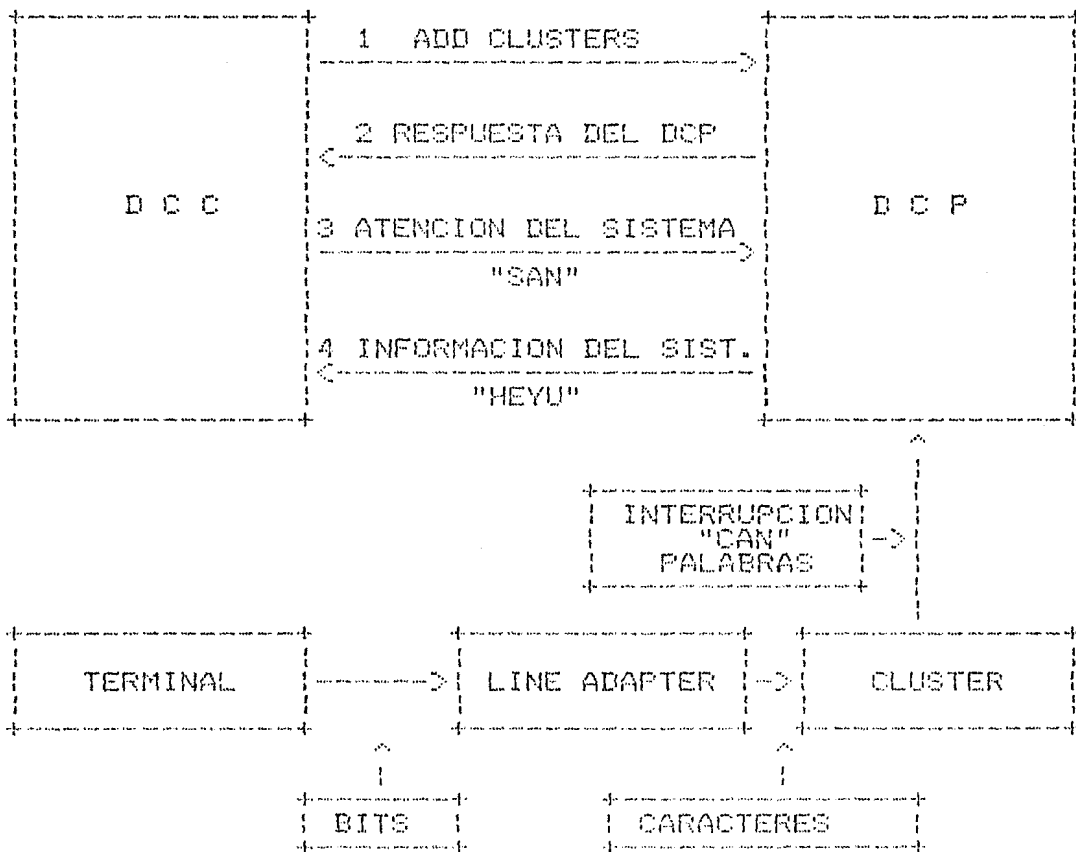
El DCC pone la petición en la 'Request Queue' de cada DCP 'ADD Clusters' y produce un 'Scan-out/Initialize' a cada DCP y espera por una respuesta de los DCP's, el DCP toma código de memoria principal y comienza la ejecución de dicho código 'Boot-Strap'. Pasando el control a una parte del código del DCP llamada 'Continuity Loop' en ese momento es cuando comienza a analizar las colas y se encuentra el 'ADD Clusters'.

El DCP prueba las interrupciones de SAN ('System Attention Needed') y el CAN ('Cluster Attention Needed') y termina la inicialización produciendo un mensaje de que la petición de 'ADD Clusters' ha sido servida, poniendo ese mensaje en la 'Result Queue'. En ese momento es cuando se comienza a recibir mensajes de las estaciones. La información proveniente de las estaciones llega por el Line Adapter bit a bit donde se mantiene hasta que pueda pasar al Adapter Cluster.

En el Adapter Cluster se reciben carácter por carácter formando palabras para cada uno de los Line Adapter en su 'Scratch-Pad Memory', recibe la palabra del Adapter Cluster en la localidad número seis de dicha memoria. Mientras tanto el DCP está esperando por alguna interrupción, ya sea de 'SAN' ó 'CAN', al tener el adaptador lista la información genera una interrupción, carga un bloque de control a su 'Scratch-Pad Memory' y recibe la palabra del Adapter Cluster en la palabra seis de dicha memoria. En el bloque de control se encuentra la dirección de memoria principal donde puede ser alojada la palabra que llegó en ese momento, el DCP lisa esa palabra al mensaje en memoria principal, al terminar el mensaje lo lisa a la 'Result Queue' con los apuntadores respectivos.

Ya que esta cola se encontraba vacía, provoca una interrupción al sistema 'Heyu'. Esta es recibida por el DCC y comienza a procesar la información de la 'Result Queue', en ese momento el DCC encuentra el mensaje y lo reconoce como la primera actividad de la estación, entonces accesa su tabla del NIF ('Network Information File') y busca cual es el MCS que controla a esa estación, e invoca a dicho MCS formulando un mensaje de 'New Station Activity' ligándolo a la 'Primary Queue' de ese MCS y deslisa el mensaje de la 'Result Queue' y lo lisa a la 'Primary Queue' del MCS. En ese momento el MCS recibe dichos mensajes y comienza así a funcionar.

Cuando otra estación intenta comunicarse con el sistema se sigue el mismo proceso, pero si el MCS ya está activo, el mensaje se pasa a la 'Primary Queue' sin activar al MCS otra vez.



PROCESO DE COMUNICACION DE DATOS EN LA B6700.

Cuando el sistema quiere enviar un mensaje a una estación, éste inserta un mensaje a la 'Request Queue' del DCP y produce un SAN ('System Attention Needed') así el DCP se da cuenta del SAN y deslisa el mensaje de su 'Request Queue' y lo lisa a la 'Station Queue' y lo manda palabra por palabra al 'Adapter Cluster Queue', quien manda caracter por caracter al 'Line Adapter' y éste a su vez transmite bit a bit por la línea de la estación, cerrandose así el ciclo.

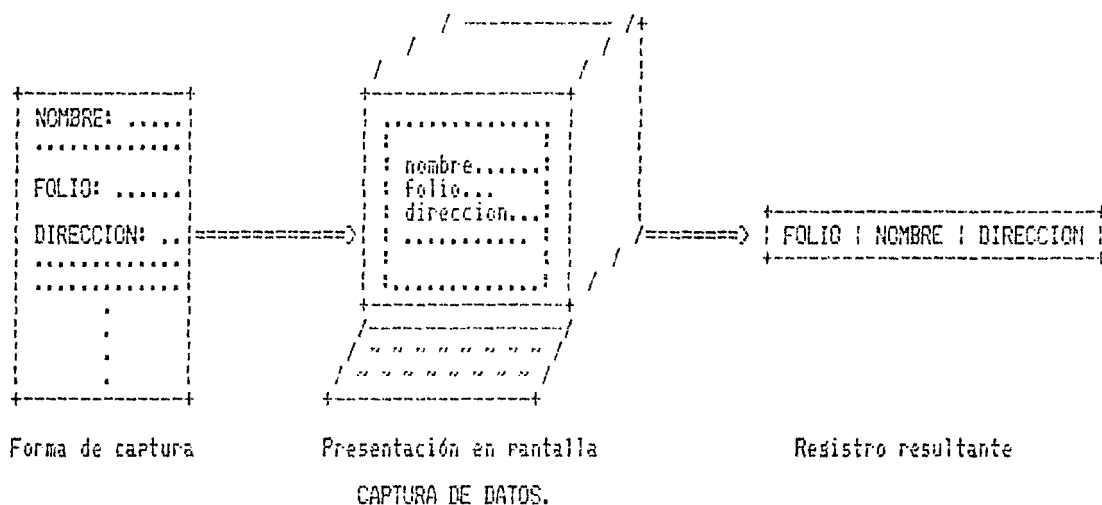
CANDE es un ejemplo de un MCS proporcionado por Burroughs, está escrito en DCALGOL; el propósito específico de este MCS es el comunicar a un usuario situado en una terminal remota, con la computadora central.

CAPITULO IV

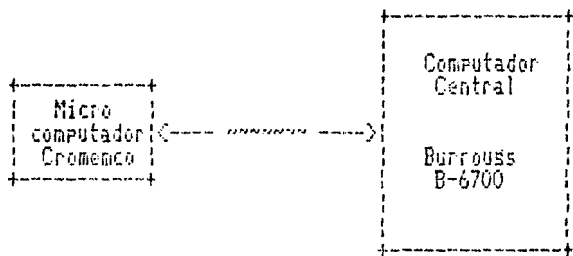
4 Diseño del Sistema

4.1 Diseño General del Sistema.

Como se a visto a través de los capitulos anteriores la realización de la tesis abarca ó comprende tres aspectos o fases principales, el primero de ellos ó primera fase del sistema es el desarrollo de un sistema de captura de datos general basado en una microcomputadora.

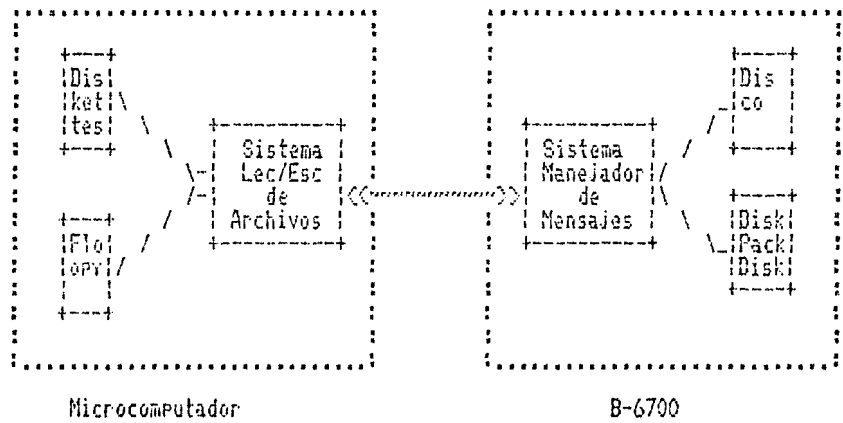


El segundo involucra la comunicación de los equipos, es decir el poder enviar mensajes del microcomputador al computador central y viceversa.



COMUNICACION.

La tercera fase del sistema total es la de tener un sistema en el computador central que nos permita tanto el envio como la recepci3n de archivos asi como parte de ellos solamente.



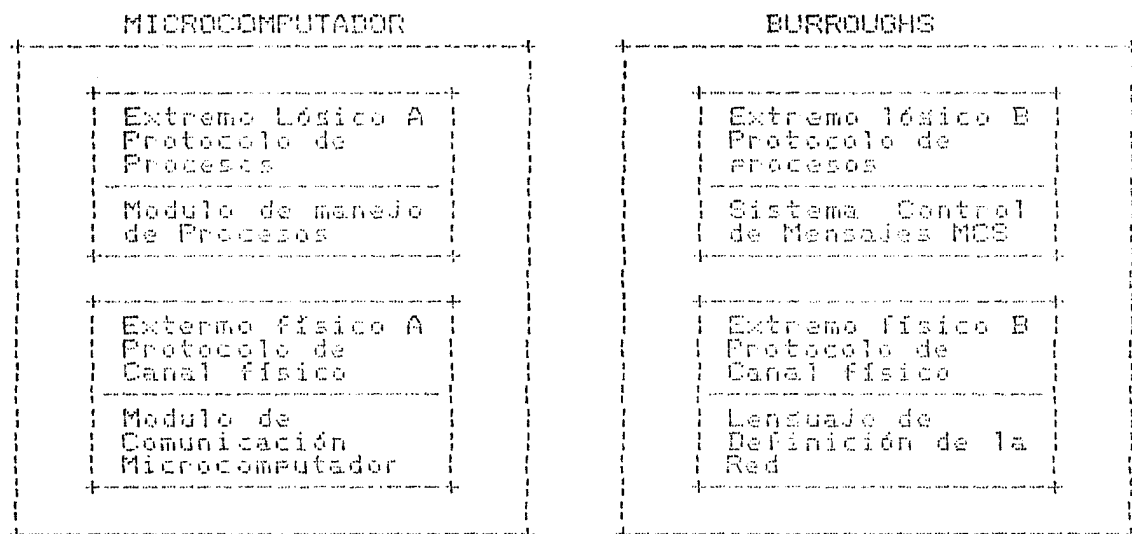
Ahora bien, en la siguiente parte del presente capitulo se abarcará la primera fase unicamente, que es el desarrollo

del sistema de captura de datos. La segunda y tercera fase del sistema se desarrollaran en un solo sistema ya que estan estrechamente ligadas, solo cabe aqui aclarar un punto en cuanto a la comunicaci3n de los equipos.

Dentro de cualquier comunicaci3n deben de existir una serie de reglas o convenciones para entablar un dialogo congruente, estas reglas que gobiernan la comunicaci3n entre dos o mas dispositivos se llaman protocolos de comunicaci3n, los cuales son esencialmente un conjunto de convenciones entre procesos de comunicaci3n sobre el formato y contexto de mensajes para ser intercambiados. Los protocolos de comunicaci3n juegan un papel importante en redes de computadoras y terminales, ellos definen como va hemos dicho las reglas de comunicaci3n entre todos los tipos de entidades dentro de una red; una entidad puede ser un usuario, un proceso, un programa de aplicaci3n 3 un sistema de archivos.

Los protocolos tambien definen el control de flujo de mensajes entre las distintas entidades que lo componen para evitar una saturaci3n de mensajes, una tercera funci3n es la de detectar y corregir los errores que pueden ocurrir durante la transmisi3n de mensajes.

En nuestro dise1o y dada la estructura de las computadoras que tratamos de intercomunicar, van a existir dos estratos o niveles de protocolo, el protocolo entre procesos y el protocolo de canal fisico.



PROTOCOLO DE DOS NIVELES O ESTRATOS.

El protocolo de canal físico estara implementado por parte del computador central en la definición de la red mas especificamente de la línea de comunicación, y sera un protocolo simple en el cual solo exista un acuse de buen recibo (ACK) y un acuse de mal recibo (NAK) y la contraparte en el microcomputador sera un programa que respetando estas convenciones reciba y transmita caracteres a través de su puerto de comunicación.

El protocolo anteriormente descrito es la forma más sencilla para la comunicación, sin embargo para un óptimo diseño de la comunicación de las líneas existen dos factores principales a tomar en cuenta:

- a) El 'Throughput' de la línea.
- b) Detección de errores y recuperación de errores en la línea.

Los factores que tienden a incrementar el 'Throughput' de la línea son:

- Reducción del número de cambios de la línea por mensaje pero no a expensas de la detección y corrección de los errores.
- Enrasetado de mensajes.
- Implementación de técnicas de compresión y expansión.

Los factores que involucran la detección y recuperación de errores son:

- Colisión en la línea. Por ejemplo, ambos lados de la línea intentan transmitir al mismo tiempo.
- Pérdida de mensajes, usualmente por falla de los módems para quedar en sincronía.
- Recepción parcial de mensajes.
- Mensajes con basura.

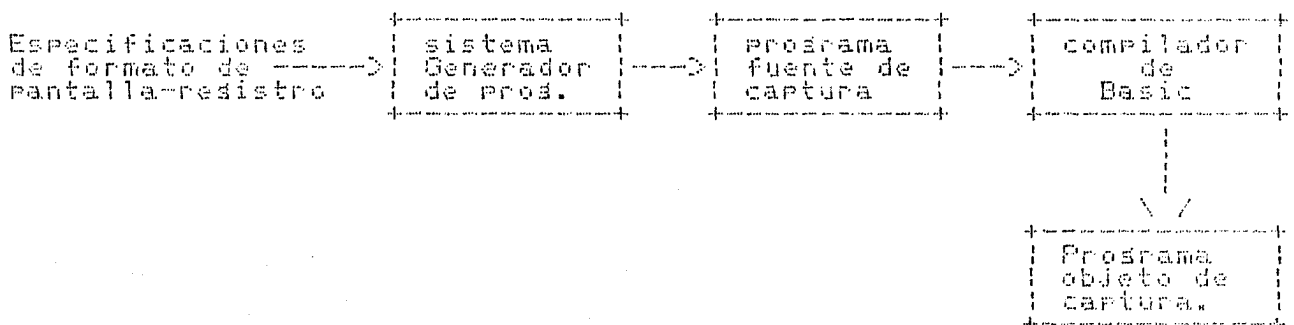
Un protocolo que cumple con los factores de protección, anteriormente descritos es el protocolo de línea RJE (B-771), el cual es el que será implementado en el futuro dentro del sistema, como se expuso en la tercer alternativa de comunicación entre los equipos. El protocolo utilizado en la primera versión del sistema es el protocolo 'TTY', comúnmente utilizado para terminales 'Hardcopy' y para algunas terminales sin módem.

El protocolo entre procesos, se encargara de controlar la transmisión y recepción de archivos, y este protocolo estara dado por las funciones que realice tanto el sistema manejador de mensajes como el sistema que este corriendo en el microcomputador (dichas funciones estan descritas en el segundo capitulo y su implementación en el siguiente capitulo). Esto quiere decir que, un comando ya específico dado en el microcomputador, debiera existir necesariamente un procedimiento que lo realice ó ejecute en el sistema central y viceversa.

4.2 Diseño del Sistema de Captura de Datos. Alternativas de Diseño.

De acuerdo a las necesidades y funciones que debe de cumplir el sistema se plantearon dos alternativas de diseño, las cuales se exponen a continuación:

La primer alternativa de diseño se muestra en la figura y es la de hacer un sistema que dadas las especificaciones de un formato tanto de pantalla como de validación de la información, genere un programa en lenguaje basic, el que al ser compilado produzca un sistema particular dependiendo de la especificaciones de entrada del formato. Esta alternativa tiene la ventaja de que es más eficiente el código generado pues, es particular a la aplicación, pero el sistema generador de programas, debe ser mucho más complejo para poder generar un código eficiente.



La segunda alternativa es la de generar una mascara, esto es una cuerda en la cual se defina cada caracter como numerico, alfabetico o alfanumerico y de acuerdo a esto hacer las validaciones de los registros ha teclear, comparandolos con la cuerda que se definio como mascara.

IAIAIAIAIAIAIAINININININIAIAIAIAIAIAIAIAIAIAIAININI

A --> Alfabético.
N --> Numérico.

La tercera opción es la de hacer un sistema general, el que se reconfigure de distinta manera de acuerdo a ciertos parámetros que se le especifiquen.

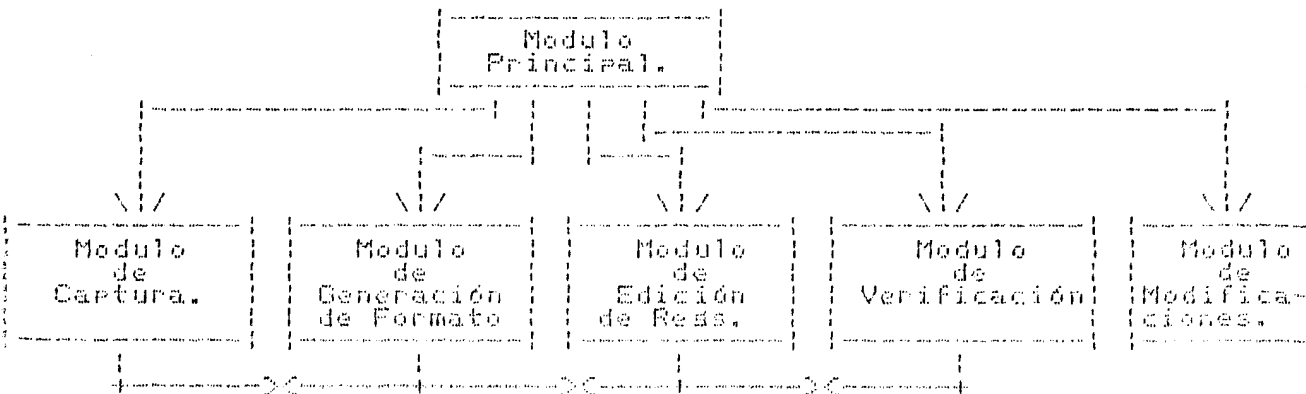
Alternativa Seleccionada

La alternativa escogida es la última expuesta en los párrafos anteriores, se decidió por esta alternativa porque era la que ofrecía más flexibilidad en cuanto a las validaciones de los datos, además, se podía hacer un diseño modular de una manera mas sencilla, y se podría trabajar con versiones del sistema, las que irían siendo cada vez más sofisticadas.

Definición de los Módulos del Sistema.

El sistema cuenta con cinco módulos los cuales tienen una área común de datos. Desde el módulo principal se llama a cualquiera de los otros cuatro módulos y al término del proceso dentro de cualesquiera de estos módulos se respesa el control a el módulo principal.

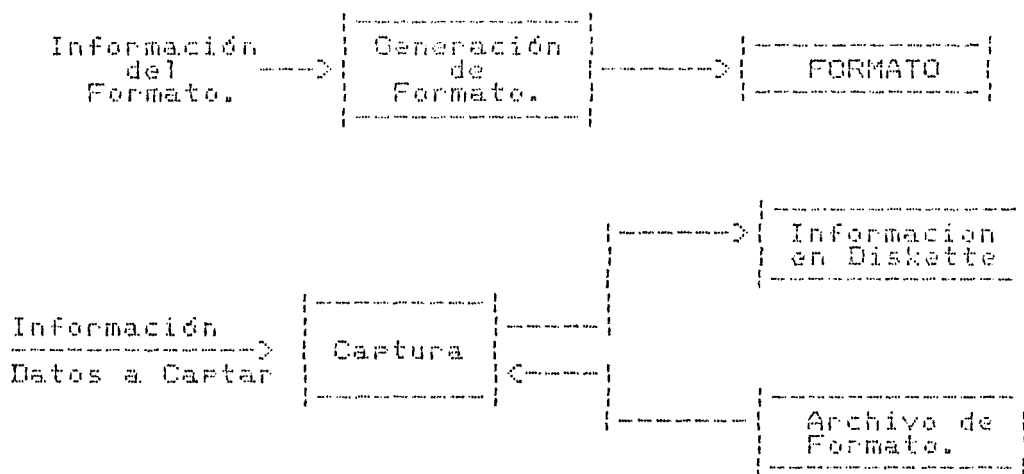
A continuación se presenta el diagrama modular del sistema:



Además de ésta descomposición modular del sistema, todos los módulos comparten una porción de código la cual contiene las subrutinas de validación que contiene el sistema y que están aseguradas en dicho código.

Descripción de Archivos del Sistema.

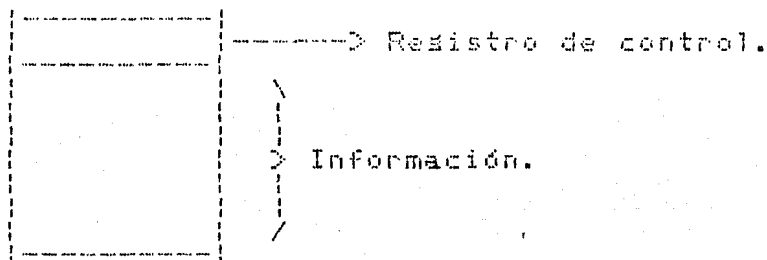
En el Sistema de Captura se manejan dos tipos de archivos: los archivos de formato que es donde se guardan el formato y los atributos de cada campo del formato, el cual es generado por medio del módulo de Generación de Formato, y los archivos de datos que es donde queda guardada la información que se captura mediante el módulo de captura.



Archivo de Datos.

Este tipo de archivo es donde se va a grabar la información, tienen en su extensión el calificativo de "SCD"

y están conformados de dos partes: la primera es un registro de control (el registro cero) el cual tiene información descriptiva del archivo y la segunda parte es comprende la información que se carte en si, la cual abarca el resto del archivo y depende del formato utilizado para su generación.



El registro de control contiene la siguiente información:

- 1er Campo --- Longitud del registro en caracteres.
- 2do Campo --- Clave del usuario en el equipo Burroughs.
- 3er Campo --- Nombre del archivo en el equipo Burroughs.
- 4to Campo --- Nombre del formato con el cual fue generado este archivo.
- 5to Campo --- Numero de registros en el archivo.
- 6to Campo --- Numero de registros verificados dentro del archivo.

Archivo de Formatos.

Este tipo de archivo contiene la información de un formato en especial y tiene en la extensión de su nombre el calificativo de "FMT". Este archivo esta dividido en cinco partes, las cuales se describen a continuación:

Reg. de Control
Formato imagen de pantalla.
Atributos de cada Campo.
Vectores de relación a las longitudes de claves.
Vectores de validación y de claves válidos.

i) La primer parte es un registro de control que contiene campos que describen el formato contenido en este archivo, estos campos son:

- 1er Campo ---- Número de ranglones en el formato.
- 2do Campo ---- Número de campos en el formato.
- 3er Campo ---- Longitud del registro.
- 4to Campo ---- Número de vectores de referencia a las longitudes de claves válidas dentro de algún campo (si es que existe este tipo de validación dentro del formato).
- 5to Campo ---- Número de cuerdas de validación donde están las claves válidas a las cuales hacen referencia los vectores descritos en el campo anterior, y las tablas de caracteres válidos para cada uno de los campos contenidos en el formato.

ii) La segunda parte contiene una imagen del formato, tal y como aparece en la pantalla de la micro.

iii) La tercera parte contiene el nombre de cada campo (un registro por campo) y sus atributos asociados, que están codificados en un vector los cuales se describen a continuación:

ATRC	< 1 >	---->	Rensión	}	Nombre del Dato.
"	< 2 >	---->	Columna		
"	< 3 >	---->	Longitud	}	Dato.
"	< 4 >	---->	Rensión		
"	< 5 >	---->	Columna	}	Registro de Salida.
"	< 6 >	---->	Longitud		
"	< 7 >	---->	Col-Inicio	}	
"	< 8 >	---->	Col-Termino		
"	< 9 >	---->	Lugar que ocupa dentro del registro	/	
"	< 10 >	---->	Tipo de Dato		
"	< 11 >	---->	Constante		
"	< 12 >	---->	Verificación		
"	< 13 >	---->	Obligatorio		
"	< 14 >	---->	Referencia <Izq,Der>		
"	< 15 >	---->	Relacion con otro Campo		
"	< 16 >	---->	Referencia a directorio		
"	< 17 >	---->	Caracter de llenado.		
"	< 18 >	---->	Referencia a res. de validaciones		
"	< 19 >	---->	Referencia a res. de validaciones		
"	< 20 >	---->	Cota-Inf. (tipo 2)		
"	< 21 >	---->	Cota-Sup. (tipo 2)		
"	< 22 >	---->	LIBRE		

iv) La cuarta parte contiene un conjunto de vectores que contiene las longitudes de las claves válidas (en caso de que existan) las cuales se encuentran guardadas en la última parte del archivo.

v) La quinta parte del archivo contiene un conjunto de vectores (cuerdas) que pueden ser de dos tipos: vectores que tienen los caracteres válidos para algún campo en especial, estos vectores son de noventa y cinco elementos (uno por cada caracter válido numeros, letras y caracteres especiales) los cuales solo podran ser ceros o unos dependiendo de si es un caracter válido o no dentro de la definición de ese campo, y el segundo tipo de vectores contiene las claves válidas para un determinado campo.

Algoritmo de Validación.

La estructura de datos utilizada en el sistema es la siguiente: como ya se vió anteriormente se tienen dos tipos de archivos relacionados entre sí por un nombre de formato el cual contiene los elementos necesarios para poder realizar la captura de la información así como las validaciones a los diferentes campos que lo componen. Para validar éstos campos se tienen dos opciones:

La primera es la de guardar los caracteres válidos y comparar estos con las cuerdas de entrada el cual tiene la ventaja de que cuando son pocos los elementos por validar es muy eficiente pero cuando son muchos sería muy tardado estar comparando éstos. Así se tendría la opción de invertir la

condición y presuntar por los elementos que no son válidos dentro de la cuerda, pero de todos modos habría que excursionar por la cuerda de entrada y una vez por cada carácter ya sea válido ó no válido, lo cual hace muy lenta esta alternativa.

La segunda alternativa que es la que se escogió, y es la de formar un vector de validación por cada campo del formato y que contiene unos y ceros dependiendo de si es un carácter válido o no para ese campo. De esta forma por cada carácter que se va leyendo en la pantalla para un determinado campo se toma su valor ASCII y se resta de 32 para eliminar los caracteres de control que están en la secuencia de caracteres ASCII y éste número nos da la posición dentro del vector de validación y la cual contendrá un uno o un cero. De esta forma con sólo un acceso a un vector, determinamos si es un carácter válido o no.

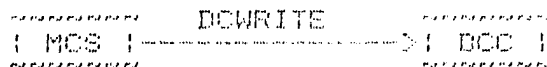
Para la validación por claves especiales se genera un cuerda con todas las claves sin ningún delimitador entre ellas estas cuerdas son de hasta 132 caracteres, asociada con cada una de estas cuerdas está un vector el cual contiene la longitud de cada clave (éste contiene una entrada por clave), de esta forma se pueden recuperar las claves de una manera rápida ahorrando el máximo de espacio, pues estas cuerdas están en memoria.

4.3 Diseño del Sistema Manejador de Mensajes 'MCS'.

En el segundo capítulo se dieron los lineamientos ó funciones que debería contener el 'MCS' propuesto, en esta parte del diseño se especificarán las estructuras necesarias que deberá manejar dicho sistema y los procedimientos que este deberá realizar para poder entablar una comunicación primeramente con el subsistema de E/S y posteriormente ya con el dispositivo conectado a la línea, que en nuestro caso es una microcomputadora.

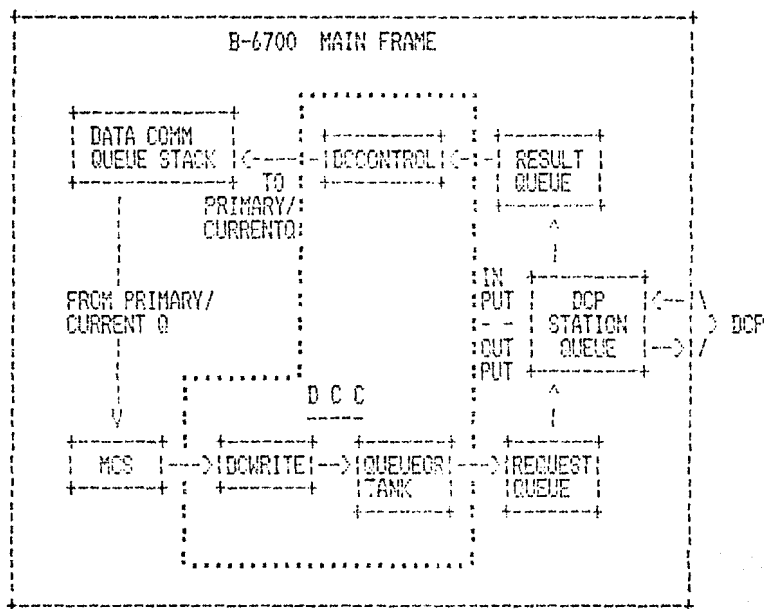
El primer punto a estudiar es el lenguaje de implementación del sistema; este será el 'DCALGOL' (Data Communications Algol) el cual contiene además del conjunto total de instrucciones del lenguaje Algol, otras instrucciones adicionales para el manejo de colas y mensajes y referencias a intrínsecos del 'MCP' los que permiten el manejo de las colas y mensajes para poder comunicarse con el subsistema de entrada salida.

De estos intrínsecos el que nos permite la comunicación con el 'DCC' (Data Communications Controller) es el 'DCWRITE', el cual tiene un parámetro que es un arreglo en el cual se especifica la acción que se desea ejecutar (tipo de 'DCWRITE'), éstas se explican más adelante.



Y para tener comunicación hacia el 'MCS' se realiza a

través de mensajes puestos por el 'DCC' en la cola asignada al sistema 'MCS'.



COMUNICACION DE DATOS.

Los 'DCWRITE' están divididos en tipos por su función hacia el 'DCC' y estos son:

ENVIRONMENT.- Del tipo 0 al 7, por medio de éstos 'DCWRITES' el 'MCS' puede inicializar la 'Primary Queue', enlazar una estación, interrogar otro 'MCS' y el medio ambiente de la terminal.

STATION.- Tipos del 32 al 52 con estos 'DCWRITES' podemos cambiar la cola de mensajes a otra distinta de la

'Primary Queue', habilitar o deshabilitar la entrada a una estación determinada, poner el número de aplicación y el número de transmisión, volver a pedir un mensaje dado, transferir el control de una estación, iniciar y terminar el modo archivo de 'DCWRITE'.

ATTRIBUTE.- tipos del 64 al 68, permiten asignar un archivo a una terminal, escribir hacia un programa en ejecución, añadir una estación a un archivo, cambiar los atributos de una terminal.

LINE.- Tipos del 96 al 105 con éstos se puede conectar o desconectar una línea (lógicamente), interrogar por el estado de una línea en especial.

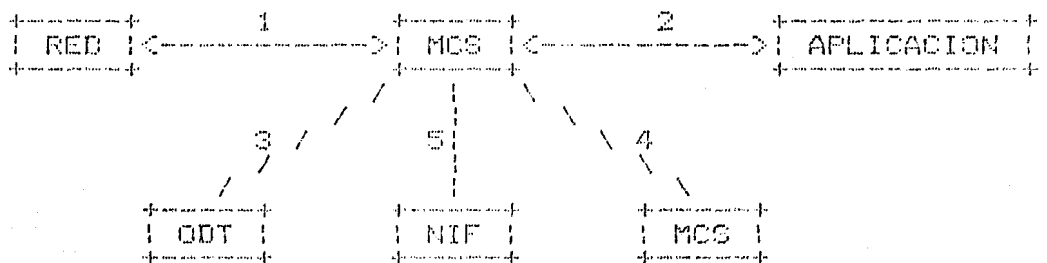
RECONFIGURATION.- tipos del 128 al 131, nos permiten intercambiar líneas, clusters, mover, añadir o quitar estaciones de una línea.

TEST.- tipos del 160 al 170, nos sirven para realizar determinadas pruebas sobre los distintos dispositivos involucrados en la comunicación (DCP, CLUSTER, ADAPTER).

DCWRITE-ERRORS.- Los tipos del 64 al 172 son mensajes de error.

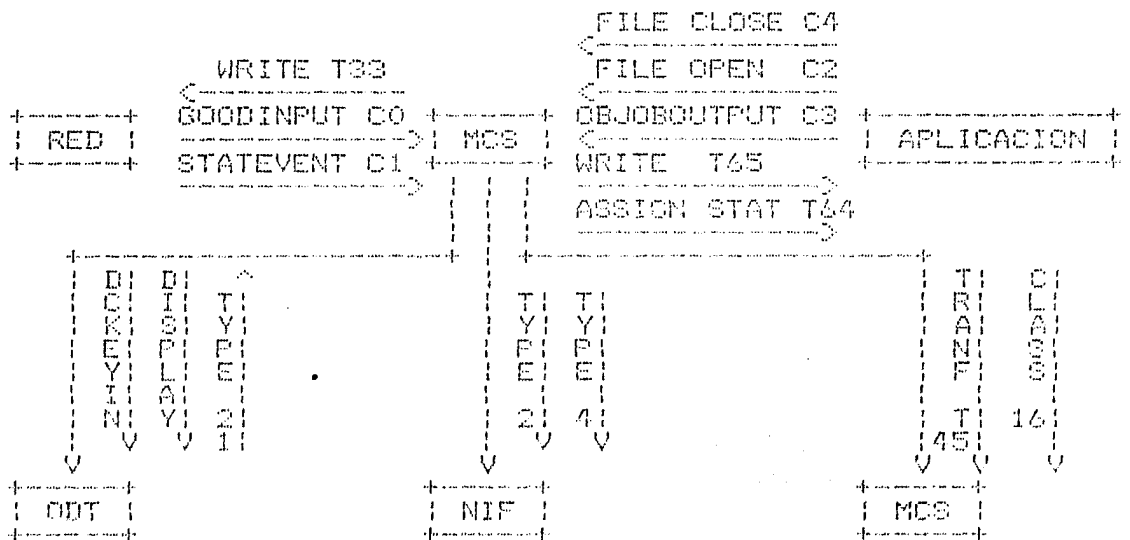
Los mensajes de entrada al 'MCS' están divididos por clases y son del 1 al 25; el 99 es un mensaje de error.

En resumen un 'MCS' tiene que controlar el envío y recepción de mensajes a varias entidades del sistema en general, a continuación se muestra los diferentes fuentes ó receptores de información:

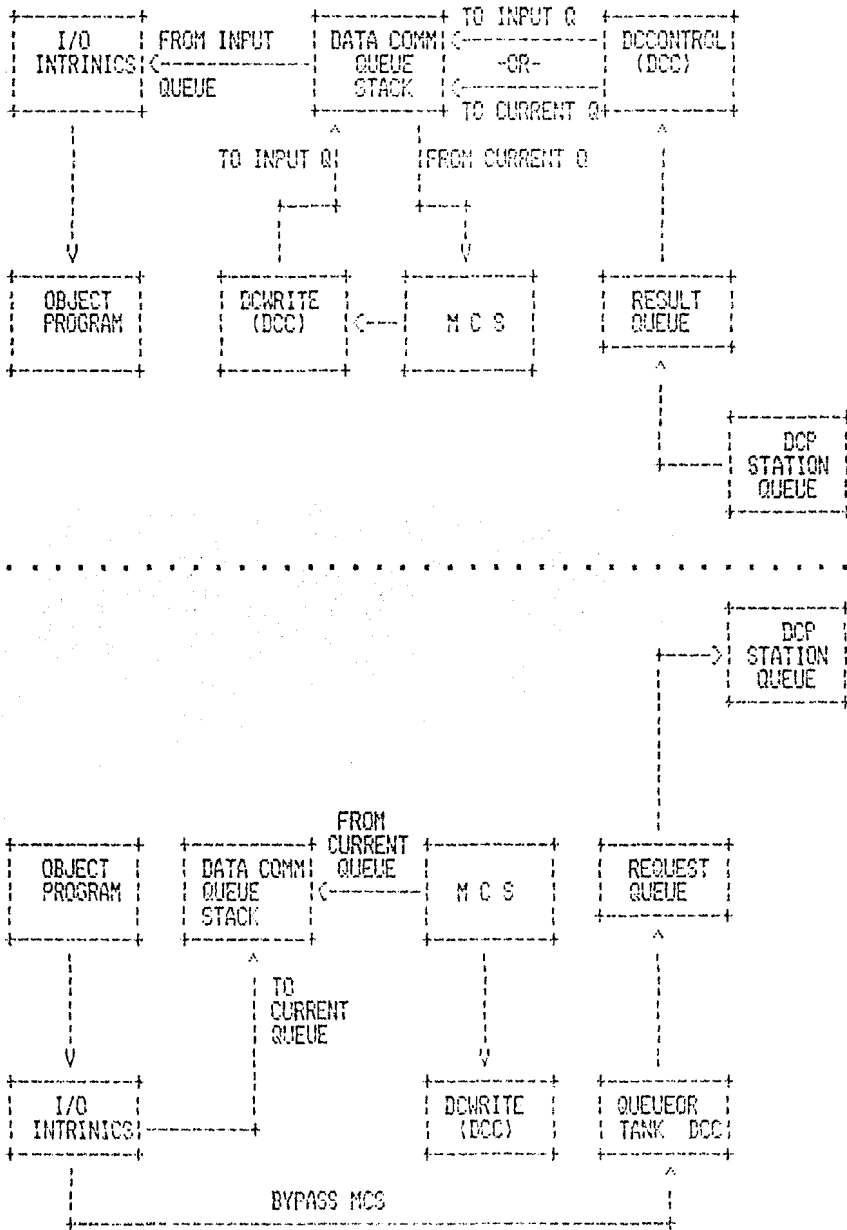


- 1 Manejo de los dispositivos conectados a la red.
 - 1.1 Transmisión y Recepción de mensajes.
 - 1.2 Cambios a la Red.
 - 1.3 Interrogar el estado de las estaciones.
- 2 Administración de la comunicación entre las aplicaciones y la red.
 - 2.1 Autorizar la apertura de archivos remotos.
 - 2.2 Participación en el I/O.
- 3 Recepción y transmisión de mensajes.
- 4 Comunicación y acciones de paso y recepción de estaciones.
- 5 Presuntar por el medio ambiente del teleproceso (líneas, estaciones, etc)

Los 'Tipos' y 'Clases' de mensajes que se utilizan para la administración de un 'MCS', como se describio anteriormente, son los siguientes:



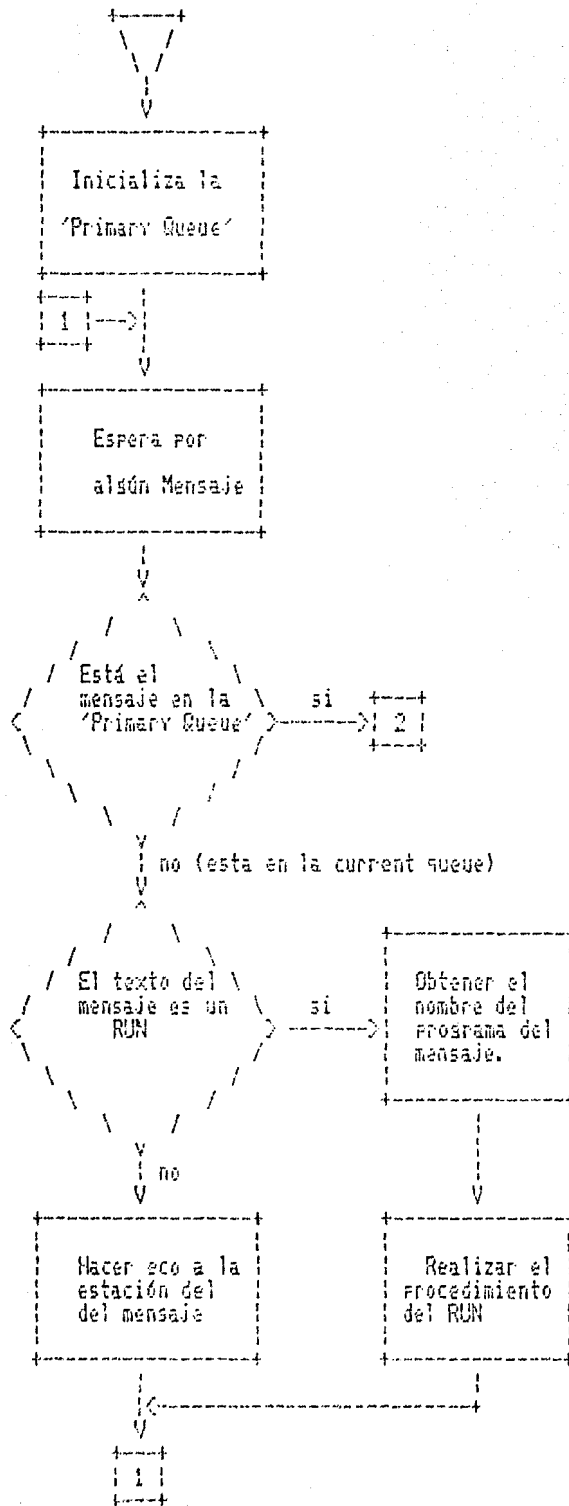
Más específicamente el flujo de información entre un programa ejecutándose en el equipo Burroughs y una estación de trabajo (en nuestro caso una microcomputadora) es el descrito en las siguientes figuras:

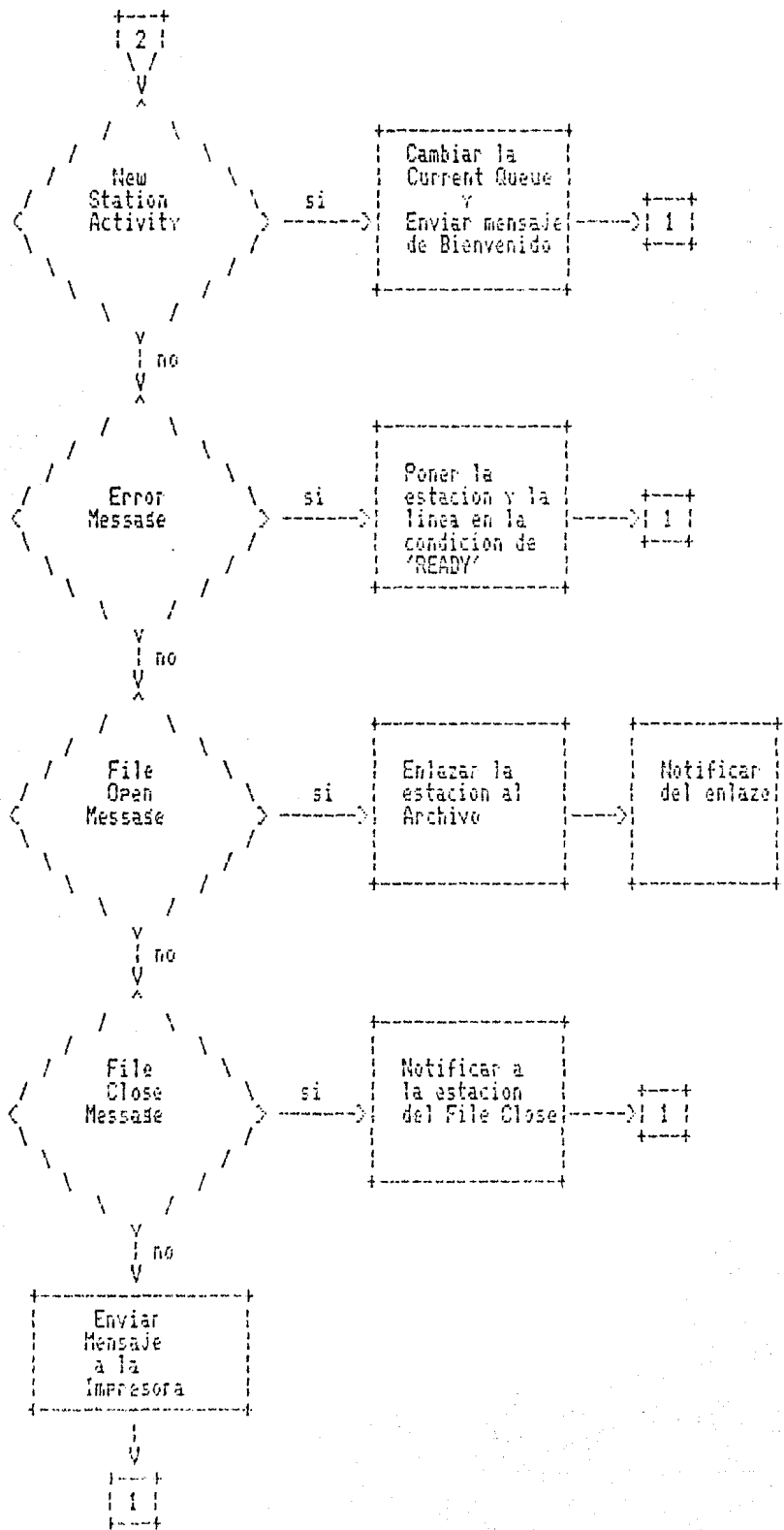


FLUJO DE INFORMACION DE/HACIA UN PROGRAMA.

Existen dos formas de ejecutar o iniciar un 'MCS', antes que nada debe de estar declarado en la definición de la red como un 'MCS'. La primera forma de ejecutarlo es que en la declaración de la red se le asignen las terminales que estarán trabajando bajo este 'MCS', o sea las líneas a las cuales estarán conectadas las microcomputadoras, y al encender cualquiera de estas estaciones automáticamente el 'DCC' ejecutará el 'MCS' asociado y enviará un mensaje a éste de 'New Station Activity'. La segunda forma es a través de CANDE, pasarle el control de la estación con el comando 'MCS SYSTEM/TESIS', el cual le pasará el control de la estación al 'SYSTEM/TESIS', si éste no está activo, es activado y se manda el mensaje de 'Transfer Station Control' en la 'Primary Queue' del 'MCS'. A continuación se describe el diagrama de flujo del Sistema 'MCS', con las acciones básicas, para poder funcionar y posteriormente se irá expandiendo éste con las funciones descritas en el capítulo dos.

El 'MCS' tiene tres funciones inicialmente, la primera es recibir estaciones para ponerlas bajo su control, asignar archivos remotos a las estaciones en caso de que algún programa así lo requiera y procesar un sólo comando que es el 'RUN', el cual solo ejecuta el programa especificado a continuación de dicho comando.





La primera acción que debe ejecutar el 'MCS' es inicializar la 'Primary Queue' la cual quedara asignada para recibir mensajes de control y la 'Current Queue' donde recibira mensajes destinados las estaciones y después deberá ponerse en un estado de espera por algún mensaje en cualquiera de sus dos colas en uso. Una vez que a llesado algún mensaje por la 'Current Queue' se analiza si fue la instrucción Run si es así se obtiene el nombre del programa y se ejecuta volviendose al estado inicial que es el de esperar más mensajes, si no fue ese comando, se hace eco a la estación, si el mensaje llesó por la 'Primary Queue' se analiza que tipo de mensaje llesó ('New Station Activity', 'Error Messase', 'File Open', 'File Close') si no fue ninguno de estos mensajes se manda a la impresora y se regresa al estado de espera por mensajes, si fue alguno de los mensajes anteriores, se procesa de acuerdo al mensaje. En este punto ya no se entrara al detalle de los campos que debere contener cada uno de los distintos mensajes que se envian o reciben, porque dsto agrandaria enormemente el presente trabajo, y tampoco se llesara al detalle de las instrucciones especificas a ejecutar para cada función.

Para cada línea que es activada se mantendrá una tabla general la cual contendrá información de:

- Número de línea asociado.
- Número de MCS al cual será regresada la línea.
- Si tiene algún proceso.
- Si ha abierto algún archivo remoto.
- Tipo de línea.

El número de línea nos identifica la línea en sí, y será el número de la estación. El tipo de línea no indicará si se trata de una terminal o una microcomputadora.

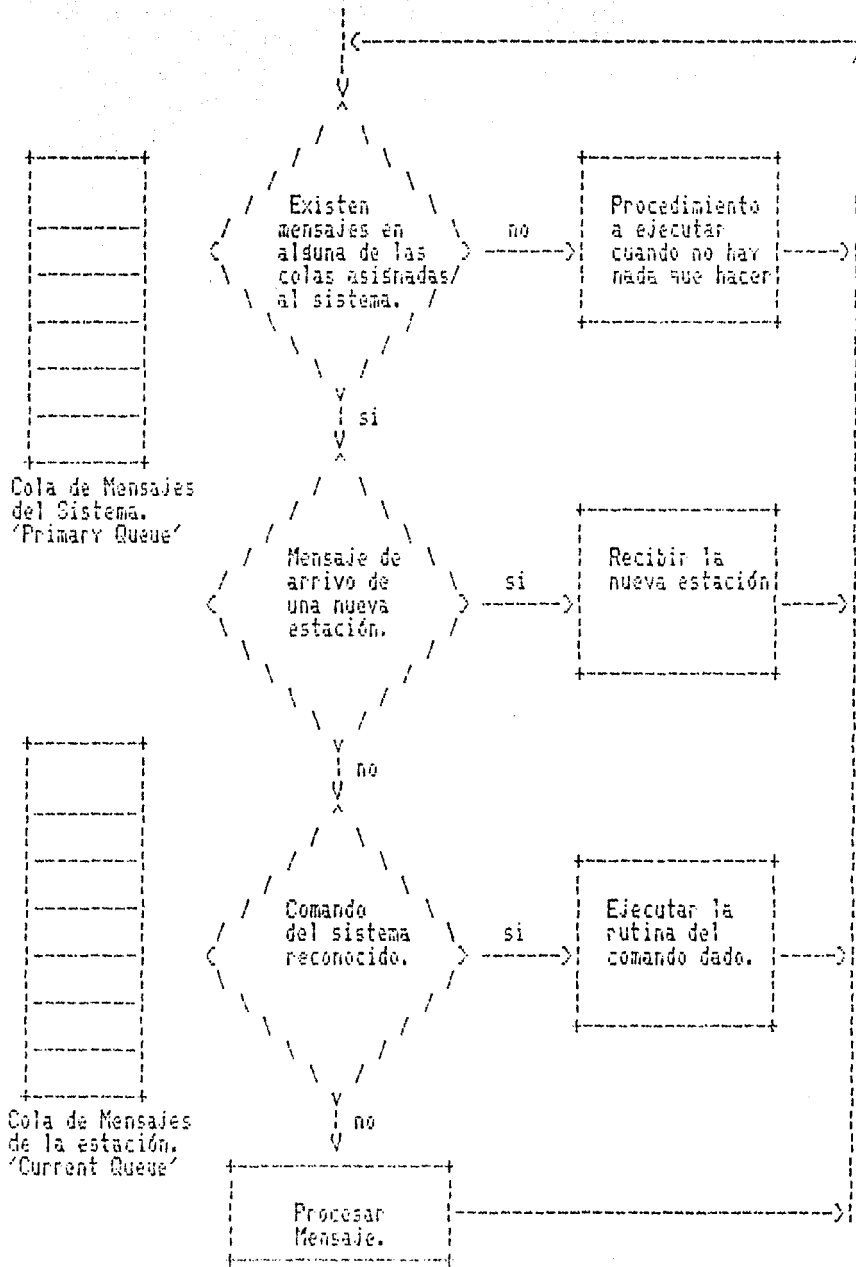


DIAGRAMA BASICO DEL 'MCS'.

Los comandos que básicamente componen al 'MCS', además de los ya vistos como son la apertura o cierre de un archivo remoto la asignación de un terminal etcétera son los que se describen a continuación:

- Recibe <Nombre de Archivo> : <Rango de Registros>
- Envía <Nombre del Archivo> : <Rango de Registros>
- Alta C# de LEND
- MSD C# de LEND " Texto del Mensaje"
- Run <Nombre del Programa>
- Quit
- Mcs <Nombre del MCS>

CAPITULO V

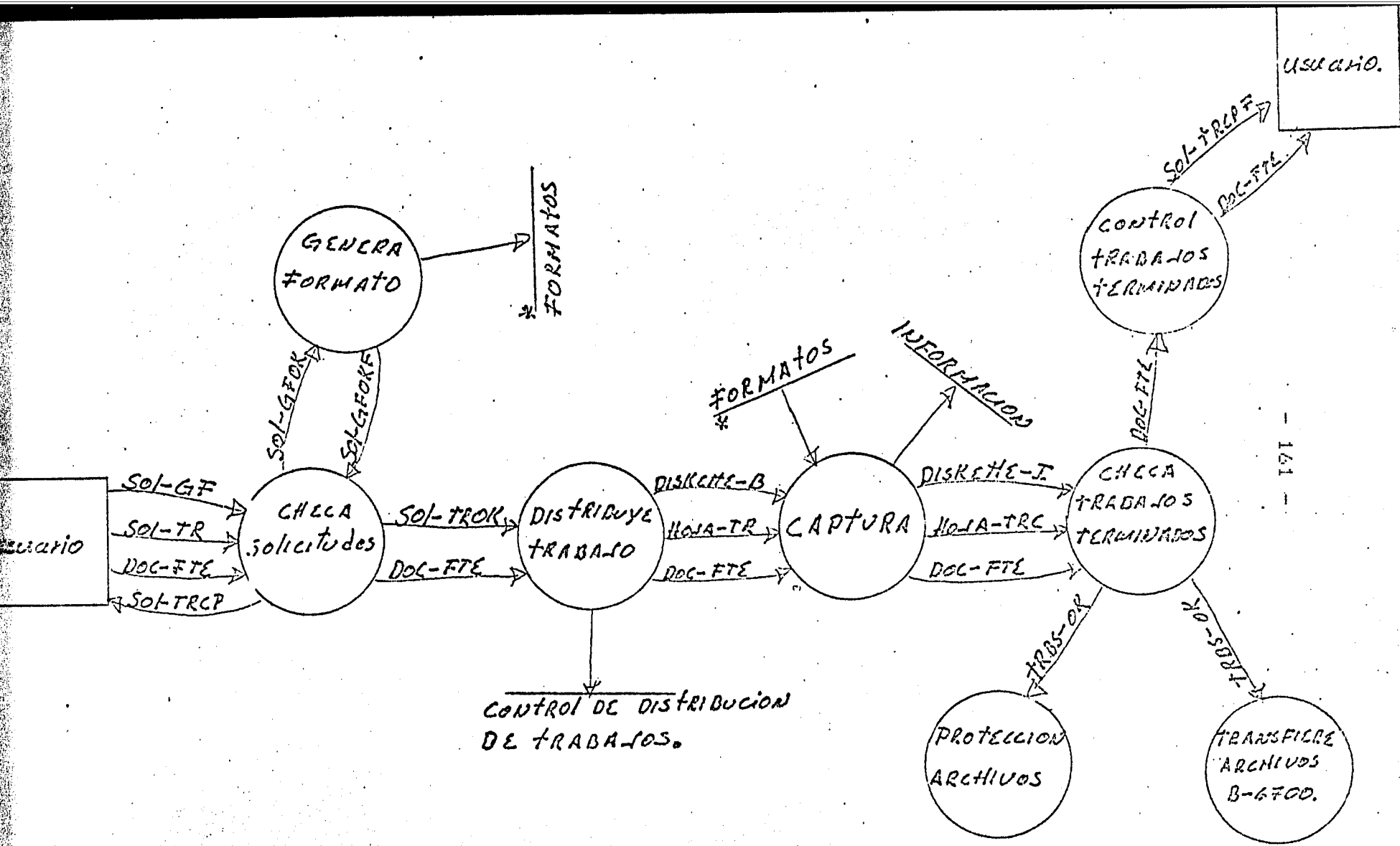
5 Implementación del Sistema

Para la implementación del sistema de captura se definió una estructura y una organización para una operación más eficiente del proceso de captura automatizado, por medio de microcomputadores. El presente capítulo da una descripción de la organización propuesta para la operación del sistema de captura de datos.

5.1 Organización del proceso de Captura.

A continuación se presenta la organización del Proceso de captura, que va desde la entrega de los documentos fuente por parte del usuario, el procedimiento interno de captura, hasta la creación del archivo con la información captada en la clave del usuario del equipo Burroughs.

La descripción del flujo de información dentro del proceso se dará en base al diagrama de la siguiente figura. (especificación del sistema) en el cual se presenta el flujo de datos. (diagrama lógico). Cada círculo representa una acción o función dentro del proceso y las etiquetas de cada flecha representan el flujo de dato o documento, los cuales están descritos en la sección de especificación del sistema. En la última sección se da una descripción funcional de las formas que se utilizan para el proceso de captura de un trabajo.



DCC-FTE = * Documentos Fuente a captar *

SOL-TR = * Solicitud de trabajo *

SOL-OF = * Solicitud de generación de formato *

SOL-OFOK = * Solicitud de formato verificado *

SOL-OFOKF = * Solicitud de formato, con el nombre del formato
generado *

SOL-TROK = * Solicitud de trabajo verificada *

HOJA-TR = * Hoja de trabajo *

DISKETTE-B = * Diskette en blanco *

DISKETTE-I = * Diskette con información *

HOJA-TRC = * Hoja de trabajo posterior a las anotaciones de la
carturista *

TRBS-OK = * Trabajo terminado; comprende diskette con información
y hoja de trabajo *

SOL-TROPF = * Copia de la solicitud de trabajo con la firma de
conformidad y entrega de material *

SOL-TROP = * Copia de la solicitud de trabajo con fecha de
recepción de material *

5.1.1 Descripción del Sistema.

Quando un usuario requiera de la captura de datos, deberá presentarse al área de captura, con el siguiente material:

- i) Documentos Fuente
- ii) Solicitud de trabajo
- iii) Solicitud de generación de formato, solamente en caso de que sea la primera vez que requiera de la captura de este tipo de documentos Fuente

Una vez hecho esto, en la mesa de control se verificará que los datos en las solicitudes estén correctas, y se indicará la fecha de entrega de la información captada en la copia de su solicitud de trabajo, entregándole dicha copia al usuario.

En caso de ser necesario generar un nuevo formato, se deberá ir con la responsable de la mesa de control. Una vez hecho esto se procederá a distribuir el trabajo tomando en cuenta:

-La carga de trabajo existente

-El volumen de los documentos fuente del solicitante.

De acuerdo a esto, se fijará el número de partes en que se dividirá el trabajo, a que capturistas se les va a asignar y en que equipo se va a realizar el trabajo.

A continuación se le asignará los diskettes correspondientes para el trabajo.

Con esta información se llenará la orden de trabajo que será entregada a cada capturista junto con un diskette para la información, un diskette de formatos y los documentos fuente.

Una vez cumplidos estos requisitos se procederá a iniciar la captura.

Esta información también quedará asentada en la forma de "Control de Distribución de Trabajo" la cual será llevada por la mesa de control, y así tener un control de los trabajos que se estén procesando.

Posteriormente se realizará el proceso de captura, grabando la información contenida en los documentos fuente en el diskette que se le proporcionó a la capturista; en esta fase se anotarán las anomalías que se presentan (problemas con los documentos fuente, con la Microcomputadora con el diskette, etc.).

Al término de la captura o del día de trabajo cada capturista entregará a la mesa de control el diskette de datos, los documentos fuente y la hoja de trabajo, con las anotaciones correspondientes a la captura, tales como número de registros captados y observaciones en cuanto al trabajo.

La responsable de la mesa de control recibirá el trabajo que entrega la capturista y lo verificará anotando en la forma de control de distribución de trabajo: la fecha de entrega, el número de registros captados y una anotación de si se terminó completamente el trabajo o si se quedó pendiente, en este último caso indicando la causa.

Al finalizar la recepción de todos los trabajos realizados durante el día, en la mesa de control se verificará cuales están ya completamente terminados, se les marcará de listos para su transmisión. Una vez verificados los trabajos, se procederá a la transmisión de los archivos al equipo B6700.

Proceso de Protección de Archivos.

El proceso de Protección de Archivos se realizará de la siguiente manera: al momento de transmitir un archivo al equipo Burroughs en la clave del usuario, se creará una copia de ese archivo en la clave del área de captura, estos archivos se conservarán en el rack, hasta el fin de semana, llegando ese momento se hará una protección a cinta se dejará un job para correr el viernes por la noche y se eliminarán los archivos del rack al finalizar la protección de cinta, dejándolo listo para la semana siguiente. En la cinta se puede tener la protección de aproximadamente dos fines de semana y llevar un ciclo de protección de dos meses, por lo tanto es necesario contar con 4 cintas para protección, mas 2 cintas de respaldo en caso de que se requiera segundo carrete en alguna protección.

Se hará también una protección de archivos a nivel de diskettes, el cual será de la siguiente manera: cada capturista recibirá un diskette en blanco para procesar los trabajos, por cada día de la semana, los cuales volverán a ser utilizados hasta la siguiente semana en su respectivo día de trabajo. Esto es, se tendrá un diskette para el lunes, otro para el martes y así sucesivamente hasta el viernes y a la siguiente semana se volverán a utilizar los mismos diskettes correspondientes al día de la semana. Hay que notar que si no se termina la captura de un día de trabajo, al siguiente día deberá continuarse con el mismo diskette, hasta terminar la captura de dicho trabajo, momento en el cual se respesa el diskette a la mesa de control y se solicitará el diskette del día correspondiente, en caso que exista trabajo.

Especificación de Documentos.

1) Solicitud de trabajo

Mediante esta forma el usuario podrá solicitar un trabajo de captura. Esta forma se hace por duplicado, quedando el original en el área de captura y la copia para el usuario.

2) Solicitud de generación de formato

Esta solicitud la entregará el usuario cuando sea la primera vez que requiere de la captura de un tipo determinado de trabajo (en reticciones posteriores de este mismo trabajo, no será necesario entregar esta solicitud). En esta solicitud se definen los campos que contiene el documento a captar.

3) Hoja de trabajo

Esta forma es llenada por la responsable de la mesa de control y después la entregara a la capturista, para que se tenga los datos necesarios, referentes al trabajo que se va a realizar.

4) Control de distribución de trabajo

Mediante esta forma se llevará el control de los trabajos que se estan procesando en el área de captura.

5.2 Operación del Sistema de Captura.

Ya en el proceso en si de captura, se deberá proceder de la siguiente manera: Cada capturista debe tener un diskette de formatos esto es para realizar un trabajo de captura se debe contar con un diskette, que contenga todos los formatos, hasta el momento generados, el cual debe ser una copia de diskette de formatos maestro. En este mismo diskette debe de existir también una copia del sistema de captura, mediante el cual se realizarán los trabajos de captura. (Este diskette siempre deberá colocarse en el "DRIVE A").

Cuando se requiera de un trabajo de captura se debe asignar a cada capturista el siguiente material:

i) Documentos fuente.

Estos son los documentos de donde se va a tomar la información a captar.

ii) Una hoja de trabajo.

En la cual aparecerá la información necesaria para iniciar la captura y en la cual la capturista anotara al finalizar la captura el número de registros que se capturaron, así como las observaciones en relación al trabajo realizado. (Problemas en cuanto a los documentos fuente, fallas de la Microcomputadora, causas por retraso en el trabajo de captura o simplemente la anotación de que se termino el trabajo sin problema alguno).

iii) Un diskette para datos

En el cual se grabará la información contenida en los documentos fuente y el cual deberá contener espacio suficiente para la captura. (Este diskette siempre deberá colocarse en el "DRIVE B").

iv) Un diskette de formatos

El cual debe de contener el sistema de captura de datos y el formato necesario.

Una vez que se tienen los documentos fuente, la hoja de trabajo y los diskettes la capturista puede ya comenzar a procesar el trabajo, en el microcomputador, para lo cual deberá seguir los siguientes pasos:

i) Inicialización del sistema

Para inicializar el sistema, lo primero que hay que hacer es encender el equipo y cargar el sistema operativo. Una vez encendida la maquina el sistema nos pondra en la pantalla:

CROMEMCO RDOS:

Inmediatamente despues tendremos que introducir nuestro diskette de formatos en el "DRIVE A" y teclear en el equipo la letra "B", hecho esto se cargara el sistema operativo del equipo y ya se puede continuar con los siguientes pasos para la captura (este paso de inicialización solo debere realizarse cuando la maquina esté apagada).

ii) Se introduce el diskette de datos en el "DRIVE B".

iii) Teclear:

Sistema captura.

Este paso es automatico cuando se inicializa la maquina.

Hecho esto, ya se encuentra en el sistema de captura de datos y ya se puede realizar cualquiera de las operaciones que permite el sistema (ver manual de operación del sistema).

Una vez concluido el trabajo la capturista debere regresar a la mesa de control el siguiente material:

i) La hoja de trabajo con datos y observaciones de la captura.

ii) Los documentos fuente.

iii) El diskette con la información captada.

iv) El diskette de formatos, en caso de que sea fin del dia de trabajo, en caso contrario se quedara con el diskette de formatos para realizar otros trabajos.

Hecho esto, la mesa de control se encargara de transmitir los archivos generados, hacia el equipo Burroughs.

Observaciones:

El sistema verificará si existe espacio suficiente en el diskette para grabar, y nos desplegara en la pantalla el espacio ocupado y el espacio disponible en el diskette, si este es suficiente podemos continuar, si no existe espacio suficiente tendremos que regresar el diskette a la mesa de control y solicitar otro diskette con espacio suficiente.

CAPITULO VI

6 Documentación del Sistema

6.1 Manual de Operación del Sistema de Captura

El sistema de Captura de Datos fué diseñado de forma tal que se pudieran definir en la pantalla del microcomputador las formas de captura, de manera semejante al diseño de los formatos para codificación y además que contemple diversos criterios de validación para los datos. Todo esto de una manera generalizada, es decir, que tuviera el menor número de restricciones para la generación del formato de captura en la microcomputadora, y en la definición de los datos a captar. Entre las características más importantes del sistema, están las siguientes:

- Poder definir un orden de captura de los datos en la pantalla y otro en el registro físico del archivo.
- Validaciones de los datos por carácter(es), palabras, RFC y rangos en el caso de datos numéricos.
- Detección y corrección de errores en los datos al momento de la captura.
- Edición de registros mediante su número de secuencia.
- Definición de datos constantes dentro de la captura.
- Definición de datos obligatorios.
- Posibilidad de verificación de ciertos datos.

6.1.1 Descripción del Sistema.

El sistema de captura de datos cuenta actualmente con cinco módulos, los cuales se describen brevemente a continuación:

Módulo Principal.- Permite el acceso al sistema y a cualquiera de los módulos descritos a continuación.

Módulo de Generación.- Mediante el cual se puede generar un formato nuevo, para captura de información.

Módulo de Captura.- Por medio de este módulo se puede captar la información de un determinado trabajo, en un formato previamente diseñado.

Módulo de Verificación.- Con este módulo se puede verificar un trabajo que ya ha sido teclado con anterioridad. (Sólo se permitirá la verificación de los campos que hayan sido declarados como tales).

Módulo de Edición.- Este módulo nos permite el acceso a cualquier registro de un archivo de datos, y dentro de cada registro la corrección del campo que se desee. También existe la opción de corregir por rango de registros.

Una vez iniciada una sesión a través del sistema, se presentará en la primera línea de la pantalla (parte alta) el menú de los comandos que se pueden ejecutar en éste, y en la última línea de la pantalla la forma de ejecutar estos comandos, que puede ser: oprimiendo sólo la tecla de la primera letra del comando o la tecla de "control" al mismo tiempo que la primera letra del comando a ejecutar.

En el sistema de captura de datos se utilizan dos tipos de archivos, los archivos tipo ".FMT" que es donde se guarda una imagen del formato de pantalla, junto con los atributos de cada campo que lo componen, y los archivos tipo ".SCD" que contienen la información captada mediante el sistema. Este último archivo contiene en el primer registro información, que es utilizada por el sistema de transferencia de archivos CROMEMCO-BURROUGHS. Para este tipo de archivos la transferencia es automática.

6.1.2 Operación del Sistema.

Este sistema actualmente cuenta con cinco módulos, cuya definición y modo de operación se daran a continuación. Para poder operar el sistema se requieren de los elementos mencionados en el capítulo cinco inciso dos.

Modulo Principal.

Por medio de este módulo podemos acceder a cualquiera de los otros módulos que contiene el sistema. Al término de un proceso en cualquiera de los otros módulos siempre se retorna a este módulo y de aquí podemos concluir la sesión de captura o acceder otro módulo.

Dentro de este módulo se presentará en la primera línea de la pantalla el menú de operaciones que se pueden ejecutar dentro del sistema, estas corresponden a los distintos módulos que contiene el sistema y que son explicados en detalle en las siguientes páginas.

EL sistema nos presentará "Que operación quieres realizar" a lo cual le podemos indicar la operación que deseamos realizar de acuerdo al menú del sistema y esto con sólo oprimir la tecla de la primera letra de la operación que se eligió.

El menú que el sistema nos presenta en la primera línea de la pantalla es el siguiente:

Operaciones: Captura Genera Edita Verifica Fin

Que Operación quieres realizar:

Al indicarle la operación que deseamos realizar (oprimiendo la tecla de la primera letra de la operación a realizar) se pondrá en un tono mas claro en el menú el nombre de la operación que se eligió. A continuación el sistema nos hará las preguntas necesarias, de acuerdo a la operación que se ha escosido. Estas preguntas se explican a continuación dentro de cada módulo.

Módulo de Generación.

Con este módulo podemos generar un formato nuevo de captura. Este módulo al invocarse nos pedirá el nombre del archivo-formato, esto es, como queremos que se llame nuestro formato para su posterior utilización. Si le damos un nombre de formato que ya fue definido, el sistema nos contestará que ese formato ya existe, por lo tanto tendremos que darle un nuevo nombre. Una vez, que se ha comprobado que ese formato no existe, la máquina nos responderá, " TECLEA FORMATO PANTALLA ", que es la forma en la cual nosotros queremos que aparezcan los campos en la pantalla. Para esto tenemos dos tipos de campos que son:

- i) Identificador.- Es el nombre del campo que vamos a captar y debe de ir encerrado entre paréntesis rectangulares " [] "
- ii) Dato.- Es la información que queremos captar y debe ir encerrado entre paréntesis triangulares " < > ", éste campo es de longitud igual a la longitud del dato a captar y puede contener cualquier caracter (excepto números que se utilizan para definir su longitud), para llenar este campo.

En una línea pueden ir tantos identificadores o datos como quepan, el sistema pone un marco del cual no se puede salir al formato, también existe la opción de que en vez de llenar el campo del dato con puntos o cualquier otro caracter hasta completar su longitud, se pone la longitud del dato con número, y el sistema automáticamente lo expandera a la longitud deseada, esto sera útil cuando los datos son muy grandes. Ejemplo:

Si deseamos generar el formato de un registro con dos campos los cuales identificamos como "CAMPO-1" y "CAMPO-2" , cuya longitud es de 5 y 10 caracteres , respectivamente podríamos definir el formato con las siguientes opciones :

```
[CAMPO - 1] <.....> [CAMPO - 2] <.....>
```

donde el caracter utilizado para llenar el campo es el punto "." o bien,

```
[CAMPO - 1] <5> [CAMPO - 2] <10>
```

donde se define la longitud del campo lo cual expandirá como:

```
[CAMPO - 1] <.....> [CAMPO - 2] <.....>
```

No es necesario que después de cada identificador sigan el campo del dato en la misma línea, se tiene la opción de teclear en una línea los campos de identificadores y en otra línea los campos de datos. Ejemplo:

```
[CAMPO - 1]           [CAMPO - 2]  
<.....>           <.....>
```

En esta forma se asocian al primer campo de identificador encontrado con el primer campo de datos encontrado y así sucesivamente. Para indicar que ya hemos finalizado nuestra definición del formato de pantalla tecleamos el comando "dF" que indica al sistema que se ha terminado el proceso.

Una vez que se ha definido el formato de captura, el sistema emite una serie de preguntas relacionadas con los atributos de cada campo. Para mayor sencillez y claridad se ha hecho un estandar en las preguntas del sistema, que es el siguiente: dentro del mismo texto de la pregunta, el sistema nos indica cuales son las respuestas válidas para cada pregunta, indicando además entre paréntesis rectangulares la respuesta que toma por omisión, esto es, si la respuesta a la pregunta del sistema es la que está entre paréntesis, sólo tendremos que oprimir la tecla de RETURN y automáticamente se

asume la opción entre paréntesis rectangulares, y pasa al sistema a la siguiente presunta; en caso contrario, se tendrá que teclear la opción que satisfaga las necesidades.

Antes de definir los atributos de cada campo del formato, el sistema nos hace dos preguntas referentes al formato de pantalla, y son las opciones de quitar o no los delimitadores de los campos de los datos, o de los campos de los identificadores. En seguida de esto, el sistema nos presentará el formato en la pantalla, y nos preguntará si el formato está bien o si se quiere volver a teclear por alguna razón, si está correcto el formato nos presentará en la parte alta de la pantalla los tipos de validaciones que se pueden tener en los datos, los cuales son 6 y se explican a continuación.

- i) Alfabético.- Este tipo de validación sólo acepta en el campo, las letras del alfabeto. (sin incluir el blanco).
- ii) Numérico.- Este tipo de validación sólo acepta los dígitos (0...9), (sin incluir el punto decimal ni el blanco). Además el sistema nos pedirá la cota inferior y la cota superior para el caso de que sea validación por un rango, si el caso es al contrario, que no se tenga un rango por validar sólo se tendrá que dar como rango a las cotas un cero.
- iii) Alfanumérico.- Este tipo de validación acepta cualquier carácter dentro del campo. (sin restricciones).
- iv) Definición.- Este tipo de validación permite definir cualquier validación por caracteres, si nosotros estamos por este tipo de validación el sistema primero nos preguntará si el campo contiene el alfabeto, después nos pregunta si contiene a los números, pudiendo contestar, o cualquiera de estas preguntas, afirmativo o negativamente, a continuación nos pedirá que le tecleemos los caracteres válidos dentro de nuestra definición. Esto nos permite definir, por ejemplo, un solo carácter válido dentro de nuestra definición.
- v) Definición de Claves.- Este tipo de validación permite definir claves, las cuales nos pedirá el sistema, para almacenar y después comparar contra cada una de ellas. Estas claves pueden ser de distintos tamaños y contener los caracteres que uno desee.
- vi) R.F.C.- Este tipo de validación permite verificar que un R.F.C. esté correcto chequeando que los cuatro primeros caracteres sean letras, a continuación puede seguir un blanco, un guión, una diagonal o inmediatamente puede venir la parte numérica, la cual se va a validar que los rangos sean 20-85 para el año, 1-12 para el mes y 1-31 para el día del mes.

A continuación se presentan los atributos de cada campo, que el sistema nos pedirá interactivamente, para cada uno de los campos que le definimos en el formato de pantalla:

- + **Posición en el Registro de Salida.**- (dado en forma de col-inicio - col-fin), esto es el lugar que nosotros queremos que ocupe el dato en nuestro registro de salida, la longitud dada por las columnas de inicio y fin debe ser la misma que se definió en el formato de pantalla, de otra forma el sistema nos lo indicará y nos pedirá nuevamente las posiciones del dato en el registro de salida.
- + **Tipo de Dato.**- Es el tipo de validación que se efectuará sobre el dato. (como ya fue explicado con anterioridad).
- + **Referenciado.**- Esto es hacia que lado va a quedar justificada la información de este campo. (izquierda o derecha, por omisión es derecha).
- + **Tipo de Llenado.**- Esto es con que caracter se va a completar el espacio faltante en algún campo. (puede ser blanco o cero o algún caracter que el usuario defina, por omisión es el blanco).
- + **Dato Constante.**- Permite definir si el campo va a ser constante durante la captura o no. Si obtenemos por campo constante el sistema inmediatamente nos pedirá el dato que va a ser constante, en todos los registros, a menos que durante la captura se le especifique lo contrario (esto se explicará con mayor detalle en el módulo de captura).
- + **Verificación.**- Con esta opción podemos definir que un campo pueda ser o no, verificado posteriormente.
- + **Obligatorio.**- Esto nos permite definir si un campo debe ser llenado o pueda no contener información.

Modulo de Captura.

Dentro de este módulo, nosotros podemos captar información, de acuerdo al formato definido con anterioridad. El sistema nos preguntará el nombre del formato que se va a utilizar, si no existe, el sistema nos lo indicará; en caso contrario (que si exista), el sistema nos pedirá el nombre del archivo de datos sobre el cual va a quedar grabada la información, si este archivo no existe se manda el mensaje de:

archivo nuevo y a continuación el sistema nos pedirá la clave del usuario y el nombre del archivo en Burroughs, donde va a quedar grabada la información, una vez hecho esto se procede a desplegar en la pantalla el formato para iniciar la captura.

Si el archivo de datos sí existe, se verifica que éste haya sido generado con el formato que le fue especificado, si no fue así el sistema nos manda un mensaje de que el archivo de datos no pueda ser procesado con este formato, si por el contrario el archivo de datos sí fue generado con el formato que le fue especificado, el sistema nos dará el número de registros que contiene el archivo, a continuación se desplegará en la pantalla el formato y en el margen superior derecho el número de registro que corresponde a captar.

Ya dentro de la captura de datos existen varios comandos que se pueden dar dentro de cualquier campo y en cualquier momento, estos comandos constan de dos caracteres el primero que es un "CNTRL-" que indica al sistema que es un comando y una letra que designa el tipo de comando de que se trata y que se explican a continuación.

CNTRL-F o Tecla de ESC: FIN.- Con este comando se termina el proceso de captura y devuelve el sistema al módulo principal.

CNTRL-C : CORRISE.- Con este comando podemos corregir cualquier campo dentro del registro presente, esto es, modificar algún campo que hayamos teclado incorrectamente y que este dentro del registro presente. Como respuesta a este comando el sistema nos presentará el nombre del campo que queremos corregir o bien podemos dar el número de campo que le corresponde secuencialmente, en lugar de teclear el nombre del campo, si no existe el campo el sistema nos lo indicará y tendremos que teclear el nombre del campo nuevamente, si ésta ya está correcto el cursor se posicionará en el campo deseado, y al terminar de teclear el campo a corregir el cursor se regresará al campo donde dimos el comando. Si en el momento en que el sistema nos pide el nombre del campo a corregir no queremos corregir nada, sólo tenemos que darle el comando "CNTRL-F" y el cursor se posicionará nuevamente en el campo donde nos quedamos.

- CNTRL-E : EDITA.- Con este comando nos podemos ir al módulo de edición que será explicado más adelante.
- CNTRL-D : DUPLICA.- Con este comando podemos poner en pantalla el registro anterior y modificar los campos que queremos, saltándonos los campos que son iguales con solo oprimir la tecla de "RETURN". (esto es útil cuando tenemos registros semejantes que sólo varían en uno o dos campos).
- CNTRL-K : CONSTANTE.- Por medio de este comando, podemos cambiar el contenido de un campo constante, o podemos definir otro campo constante aunque no haya sido definido en el formato al momento de ser generado, o podemos hacer que un campo constante deje de serlo, todo esto, sólo tiene efecto durante nuestra sesión, al finalizar, el formato queda como cuando fue definido. Al invocar este comando, el sistema nos presentará el nombre del campo que queremos modificar, (también podemos darle sólo el número de campo que le corresponde), si éste es correcto, a continuación nos pedirá el dato constante nuevo, esto es, cuando queremos modificar un campo constante o queremos definir un campo constante nuevo, pero si lo que queremos hacer es que el campo constante deje de serlo, en este momento le tecleamos "CNTRL-N" que significa no constante, y el campo pasa a ser variable.

Modulo de Edición.

Existen dos formas para acceder éste módulo :

La primera es por medio del módulo principal, el cual nos pedirá el nombre del archivo-formato y el nombre del archivo-datos los cuales deberán de existir y además deberán ser compatibles, si esto no ocurre el sistema nos lo indicará.

La segunda forma es acceder este módulo por medio del de captura al cual se le dará el comando "CNTRL-E", hecho lo cual estaremos automáticamente en el módulo de edición, trabajando con los archivos que se definieron en el de captura. Estando en el módulo de edición también podemos trasladarnos al de captura o viceversa.

Por medio de este módulo, podemos corregir cualquier campo dentro de cualquier registro o rango de registros. Para poder acceder un registro utilizamos el número secuencial del registro como llave de acceso a tal registro, hecho lo cual el sistema nos desplazará en la pantalla dicho

registro, también podemos adelantarnos o retroceder cuantos registros se requieran, esto se realiza de la siguiente manera: el sistema nos presentará Número de Registro, a lo cual le podemos contestar de cinco formas que son:

- i) Dar un número válido de registro, por ejemplo "27" y el sistema nos desplegará en la pantalla el registro número "27".
- ii) Si lo que queremos es irnos 5 registros más adelante por ejemplo, al registro "32", se tecleará "+5" y automáticamente el sistema nos desplegará el registro "32".
- iii) Si lo que queremos es regresarnos podemos teclrear "-10" y se ira diez registros hacia atrás (al "22") y nos desplegara dicho registro.
- iv) Si lo que queremos es acceder registro por registro no tenemos mas que teclrear el registro desde donde queremos iniciar, y para pasar al siguiente registro solo tenemos que oprimir la tecla de "RETURN", automáticamente pasara al siguiente registro y desplezandolo en la pantalla.
- v) Si queremos hacer una o mas correcciones por un rango de registros, lo que tenemos que hacer es teclrear el rango de registros de la forma "N1-N2" donde el primer número es el primer registro que queremos corregir y el segundo el ultimo a corregir, hecho esto el sistema nos presentará en la pantalla el primer registro del rango sobre el cual queremos hacer las modificaciones o correcciones (que se efectuarán sobre todos los registros del rango definido).

Despues de que el sistema nos presenta el número del registro y que nosotros le respondamos de alguna de las formas antes mencionadas, éste nos preguntará por el campo que queremos corregir a lo cual le podemos responder con el nombre del campo o el número del campo a corregir, si éste está correcto el cursor se posicionará en el campo deseado, al terminar de teclrear el campo el sistema nos vuelve a preguntar el nombre del campo a corregir si ya no queremos corregir ningun campo solo oprimimos la tecla de "RETURN", y el sistema respesa a preguntarnos por el número de registro, si ya no queremos corregir mas registros la podemos teclrear "CNTRL-F" u oprimir la tecla de "ESC" con lo cual finalizamos la edición y regresamos al módulo principal, o "CNTRL-C" que nos transfiere al módulo de captura.

Modulo de Verificación.

Dentro de este módulo podemos verificar un archivo de información que haya sido captado con anterioridad, este módulo hará una verificación de los archivos de formato y de datos de igual forma que lo hace para el módulo de captura.

Al entrar a este módulo nos informa de cuantos registros consta el archivo de datos y cuantos de ellos han sido verificados, y nos despliega en la pantalla el registro a turno a ser verificado, los campos que hayan sido definidos como verificación (dentro del módulo de generación de formato), aparecerán en la pantalla en color mas brillante y el cursor se posicionará en el primer campo a verificar y a continuación podemos teclear la información para verificar si está correcta o no, a lo cual puede ocurrir alguna de las siguientes situaciones :

- i) Que la información del registro concuerde con lo que nosotros tecleamos, por lo cual no existe ningún problema y el sistema pasa automáticamente al siguiente campo o al siguiente registro a verificar.
- ii) Que la información que nosotros tecleamos no sea igual a la del campo que estamos tecleando lo cual el sistema nos lo informará y nos pondrá en la primera línea de la pantalla el contenido inicial del campo, así nosotros podemos decidir cual de las dos informaciones está correcta .
- iii) Si la información correcta es la que contenía con anterioridad el campo solo tendremos que oprimir la tecla de "RETURN" y el sistema pasara al siguiente campo o registro , dejando el contenido inicial del campo.
- iv) Si la información correcta es la que nosotros tecleamos tendremos que dar el comando "CNTRL-A" y el sistema nos pondrá en dicho campo la información que nosotros le havamos tecleado al momento de verificación borrando la anterior información .
- v) Si ninguna de las dos informaciones está correcta (la que contenía en un principio el campo ni la que nosotros le tecleamos), solo tendremos que volver a teclear la información que sea la correcta.

Finalmente para salir de este módulo solo se oprime la tecla de "ESC" y pasamos al módulo principal del sistema. No es necesario terminar de verificar el archivo completamente, se puede cortar una sesión y continuarla posteriormente sin ningún problema, pues el sistema se encarga de iniciar la verificación en el lugar correcto.

6.1.3 Lista resumida de los comandos del sistema.

La siguiente es una lista de los comandos que posee el sistema de captura de datos, dividida por módulos:

Módulo Principal.

Dentro de este módulo solo bastara con oprimir la primer letra de cada comando para su ejecución.

- <C>aptura --- Transfiere el control al módulo de captura.
- <G>enera ---- Transfiere el control al módulo de generación de formatos.
- <V>erifica -- Transfiere el control al módulo de verificación.
- <A>yuda ----- Se obtiene información del sistema ya sea por pantalla o por impresora.
- <E>dita ----- Transfiere el control al módulo de edición de registros.
- <F>in ó <ESC> Termina la sesión de captura.

Para ejecutar los comandos de los siguientes módulos bastara con oprimir la tecla de control simultaneamente con la primer letra del comando escogido.

Módulo de Captura.

- CNTRL-F ó <ESC> --- Termina el proceso de captura y resresa al módulo principal.
- CNTRL-C ----- <Corrige> Para corregir cualquier campo dentro del registro actual de trabajo.
- CNTRL-E ----- <Edita> Transfiere el control al módulo de edición.
- CNTRL-D ----- <Duplica> Nos copia el registro anterior en el registro presente.

CNTRL-K ----- <constante> Nos permite definir cualquier campo como constante. Para definir que un campo ya no es constante se tecldea CNTRL-N en el momento que se vide dato constante nuevo.

Módulo de Edición.

CNTRL-F <ESC> -- Termina el proceso de edición y regresa el control al módulo principal.

CNTRL-C ----- Transfiere el control al módulo de captura.

C####> ----- Nos despliega el #### registro para su corrección, siempre que este sea válido.

C####-####> ---- Nos despliega el primer registro del rango que se le dio, para la corrección en todos los registros, siempre que este rango sea válido.

<+>#### ----- Nos despliega el registro que esta #### registros mas adelante del que tenemos en pantalla.

<->#### ----- Nos despliega el registro que esta #### registros mas atras del que tenemos en pantalla.

RETURN ----- Para pasar de un registro a su inmediato sin tener que teclear el número de registro siguiente. También nos sirve para indicarle al sistema que no se va a hacer ninguna corrección a un determinado registro (esto funciona cuando el sistema nos pregunta el nombre del campo que queremos corregir).

Módulo de Verificación.

CNTRL-F <ESC> -- Termina el proceso de verificación y redresa al control al módulo principal.

CNTRL-A ----- Si existió algún error en la verificación, con este comando le indicamos al sistema que el dato que tecldeamos en ese momento es el correcto.

RETURN ----- Si existió algún error en la verificación, con esta tecla le indicamos al sistema que el dato que contenia el campo (anterior a que tecldearamos) es el correcto.

Módulo de Generación de Formato.

Para definir un formato se especifican dos tipos de campos:

- Identificador.- Que va entre paréntesis rectangulares y sirve para dar nombre a una campo.
- Dato.- Es donde vamos a captar la información, y para su definición va entre paréntesis triangulares.

Después de teclear el formato de pantalla el sistema nos presentará por los atributos de cada campo.

Posición Col-Inicio Col-Fin

Este es el lugar que se desea que ocure el dato en el registro de salida, la longitud dada entre las columnas de inicio y fin, debe coincidir con la que se definió en el formato de pantalla.

Tipo de Campo

Esto es el tipo de validación que se debiera aplicar al campo, los cuales pueden ser una de seis y que corresponden a:

- | | |
|-----------------|-------------------------------------|
| 1. Alfabético | 4. Definición de Caracteres válidos |
| 2. Numérico | 5. Definición de Claves |
| 3. Alfanumérico | 6. R.F.C. |

Referencia Izq/Der

Esto es hacia que lado va a quedar cargada la información de este campo. (por omisión es Derecha)

Llenado B--> b , O--> O , CAR. ESP.

Esto es con que caracter se va a completar el espacio faltante, en este campo (en caso de que no se haya llenado completamente), y este caracter puede ser: Blanco en cuyo caso se pondrá el número "1" o cercos "O" o algún caracter especial que el usuario desee para lo cual se debiera teclear el número "2". (por omisión es Blancos)

Constante Si/No

Aquí el usuario podrá definir si este dato es el mismo en varios registros (aunque esto no significa que este dato no se pueda cambiar durante la captura de los documentos fuentes).

Esto es para ahorrar el trabajo de repetir el mismo dato en varios registros, si éste no es el caso simplemente se tecleara la letra N o No. (por omisión es No)

Obligatorio Si/No

Si un campo debe de contener siempre información debe indicarse afirmativamente, la ausencia del dato en el documento fuente da como consecuencia el rechazo de este registro, en la captura.

CAPITULO VII

B Conclusiones

El desarrollo del trabajo de tesis fue satisfactorio, debido a que se alcanzaron en gran parte las metas propuestas para este trabajo; el sistema de captura de datos fue terminado e implantado y del tiempo que ha estado en producción se han desarrollado nuevas versiones de éste y se ha depurado el sistema, en la parte de transmisión también se implantó la primera versión mencionada en el capítulo dos y se ha continuado trabajando en ella y en la implantación de nuevas adiciones. Realmente creo que cualquiera de los tres subsistemas realizados hubieran bastado para el desarrollo de una tesis aceptable, y tal vez se hubieran podido realizar mas eficientemente por separado. Por esta razón pienso que la tesis fue muy ambiciosa, y sentó las bases para futuros desarrollos.

En resumen, el primer provecho que se obtuvo del desarrollo de la presente tesis fue el estudio que se tuvo que realizar sobre los equipos, lo cual llevo mas de la mitad del tiempo de trabajo sobre la tesis, pero esta experiencia adquirida servirán para terminar de complementar el desarrollo iniciado con esta tesis, además de servir también para otros desarrollos en los mencionados equipos. De lo anterior creo que la mayor satisfacción es que el trabajo realizado (en la fase de captura) ha servido para la Universidad, se ha utilizado y se sigue utilizando en varias dependencias de esta y yo creo que es el mejor aliciente para seguir trabajando en ello.

7.1 Bibliografía.

- (1) Information Processing Systems, Reference Manual, BURROUGHS B6700
- (2) System Miscellanea relative to mark 2.6 release, BURROUGHS B7000/B6000 SERIES
- (3) System Software Operational Guide (SSOG) Vol. II, BURROUGHS B6700
- (4) Data Communications Functional Description, Burroughs B6700
- (5) DCALsol, Reference Manual, Burroughs Corporation.
- (6) Training Course Data Communications, Student Material (EP-4043), Burroughs Corporation.
- (7) NSL, Reference Manual, Burroughs Corporation.
- (8) Large Systems MCP Manual, Release 3.3, Febrero 1980, Burroughs Corporation.
- (9) Basic System Support Large Systems, Student Material, Burroughs Corporation.
- (10) CDSOS (Cromemco Disk Operating System.) Instruction Manual, Cromemco, inc. Enero 1981. Part No. 023-0034.
- (11) Cromemco ZPU, Z-80 Central Processing Unit, Instruction Manual, Enero 1981.
- (12) Cromemco TU-ART, Digital Interface Instruction Manual, Enero 1980. System Zero, Instruction Manual Febrero 1981, CROMEMCO, Inc.
- (13) Macro Assembler, Instruction Manual. CROMEMCO, inc. octubre 1979. Part No. 023-0037.
- (14) Link and Lib. Reference Manual. CROMEMCO, inc. septiembre 1980. Part No. 023-4024.
- (15) SBasic 32K Structured Basic. CROMEMCO, inc. septiembre 1979. Part No. 023-0030.
- (16) Fortran IV. Reference and User's Manual. CROMEMCO, inc. marzo 1979. Part No. 023-0038.
- (17) Cobol. Instruction Manual and User's Guide. CROMEMCO, inc. abril 1979. Part No. 023-0049.
- (18) CBasic. Reference Manual. Software Systems. agosto 1978. Version 2.
- (19) Basic 16K Extended Basic. Instruction Manual. CROMEMCO, inc. diciembre 1979. Part No. 023-0037.

- (20) Screen Editor. Instructio Manual. CROMEMCO, inc. enero 1981. Part No. 020-0081.
- (21) Interconexión entre dos computadoras B-6700, Tesis 1979, Boris Bobin Rosenthal.
- (22) UNAMNET Red de Universitaria de Telemproceso, Tesis 1979, Marco Antonio Bendimez Serrano.
- (23) Enlace de una Microcomputadora con una Burroughs B6700. Tesis 1982, Mario Alberto Rivera Ponce.
- (24) Estudio de Viabilidad para la implantación de Microprocesadores de apoyo a la Universidad, Abril 1981, Colmenares Baron Raul.
- (25) Aspectos Generales de Hardware y Software del equipo B6700, Febrero 1983, Ma. Eusebia Mora Diaz Barrisa, Jose Luis Olmedo Talavera.