

27
24



UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

**SISTEMA DE CAPTURA Y EDICION DE ARCHIVOS,
UNA HERRAMIENTA DE PRODUCTIVIDAD MAIN FRAME-PC**

T E S I S

**QUE PARA OBTENER EL TITULO DE
INGENIERO EN COMPUTACION**

P R E S E N T A N

**MARTHA GUTIERREZ GONZALEZ
ORALIA CARRILLO ROMERO**

DIRECTOR DE TESIS

ING. HECTOR CAMPOS ESTRADA

FALLA DE ORIGEN

MEXICO, D.F.

1990

FALLA DE ORIGEN





UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

INDICE TEMATICO

Capítulo 1 Introducción	1
1.1 Definición del Problema	2
1.2 Objetivo del Sistema	2
Capítulo 2 Análisis del Sistema	3
2.1 Metodología de Sistemas	3
2.2 Metodología Utilizada para el Análisis	6
2.3 Etapas para el Desarrollo del Análisis del Sistema	8
2.3.1 Análisis del Sistema Actual	8
2.3.1.1 Descripción del Sistema Actual	8
2.3.1.2 Restricciones del Sistema	12
2.3.1.3 Requerimientos del Sistema	13
2.3.2 Justificar una Estrategia de Solución para el Problema	15
2.3.2.1 Esbozar Varias Estrategias de Solución Considerando las Restricciones	15
2.3.2.2 Selección de la Estrategia de Solución Óptima	17
Capítulo 3 Descripción del Equipo	19
3.1 Equipo PC	20
3.1.1 Software del Equipo PC	20
3.1.2 Hardware del Equipo PC	28
3.2 Lenguaje de Programación C	35
3.2.1 Descripción del Lenguaje C	36
3.3 Equipo A9	40
3.3.1 Hardware del Equipo A9	40
3.3.2 Software del Equipo A9	46

Capítulo 4 Diseño del Sistema	55
4.1 Metodología del Diseño del Sistema	55
4.2 Etapas a Desarrollar en el Diseño del Sistema	58
4.2.1 Diseño Externo del Sistema	58
4.2.1.1 Consideraciones para el Diseño	58
4.2.1.2 Características Funcionales del Sistema	60
4.2.1.3 Estructura General del Sistema	61
4.2.2 Diseño Interno del Sistema	63
4.2.2.1 Descripción Conceptual de los Archivos del Sistema	63
4.2.2.2 Descripción Funcional de los Módulos del Sistema	69
Capítulo 5 Instrumentación del Sistema	76
5.1 Formato y Documentación Interna	76
5.2 Estructuras Utilizadas en el Sistema	78
Capítulo 6 Implantación del Sistema	91
6.1 Primera Etapa: Introducción al Sistema	92
6.2 Segunda Etapa: Instalación Parcial del Sistema	93
6.3 Tercera Etapa: Instalación Total del Sistema	93
Capítulo 7 Documentación del Sistema	94
7.1 Documentación Técnica	94
7.2 Documentación de Operación	95
7.3 Documentación de Referencia	95
Capítulo 8 Diseño del Sistema de Comunicación	97
8.1 Conceptos Generales de Comunicación	97
8.2 Características del Equipo PC para Comunicaciones	99
8.3 Características del Equipo A9 para Comunicaciones	107
8.4 Diseño General del Sistema	112
Conclusiones	124

CAPITULO 1

INTRODUCCION

Desde los inicios de la civilización, la humanidad ha necesitado información como una ayuda en la lucha por la supervivencia, así como en los intentos para administrar las organizaciones. La creciente complejidad de la sociedad, sobre todo en la forma en que se manifiesta en las organizaciones sociales, políticas y económicas, ha aumentado en gran medida la necesidad de tener información más conveniente y oportuna.

La computadora, con los programas apropiados, constituye de manera especial una herramienta adecuada al nivel de estas complejidades y cambios. La combinación de las computadoras con los medios de telecomunicación ha suministrado herramientas y técnicas muy poderosas para implantar sistemas de respuesta rápida, y de esta manera le ha dado una nueva dimensión al concepto de sistemas de información.

La Universidad Nacional Autónoma de México, siendo una de las instituciones educativas más grandes del país, no está exenta de la necesidad de la información en sus escuelas, facultades, institutos y otras dependencias. El crecimiento de la institución, así como las diversas actividades que realiza, requieren en diferentes sentidos del manejo de grandes volúmenes de información.

Uno de los problemas más grandes, en cuanto al manejo de información en la UNAM es el referente a la administración, el cual es solucionado en parte en los lugares donde se origina dicha información, pasando después, si es necesario a otras instituciones dedicadas a dar apoyo a los procesos administrativos. La Dirección General de Servicios de Cómputo para la Administración (D.G.S.C.A) es una institución creada principalmente para satisfacer los requerimientos de cómputo de las dependencias que demandan sistemas automatizados en el control y seguimiento de procesos administrativos, es por ello que una de sus funciones es la de apoyar la integración de la información que se genera en distintas dependencias, la cual mediante procesos adecuados, permite proporcionar información confiable y oportuna para una óptima planeación y toma de decisiones.

El presente trabajo es un sistema que se desarrolló para la D.G.S.C.A. y tiene como objetivo, por una parte, el mejorar los procesos de captura de la información proveniente de diferentes fuentes en la Universidad para los procesos que se requieran, por la otra

facilitar la edición de este tipo de archivos de tal manera que se optimice el uso de los recursos tanto humanos como de cómputo.

1.1 Definición del Problema

Por una parte, dentro del desarrollo de sistemas el analista siempre enfrenta los problemas de la captura, verificación y corrección de la información que se va a manejar independientemente de la aplicación que vaya a realizar el sistema. El desarrollo de los programas para la solución de dichos problemas involucra tiempo que va desde una semana hasta un mes, si se contara con una herramienta sencilla y eficiente para capturar, verificar y corregir la información se reduciría bastante el tiempo de desarrollo de los sistemas de aplicación que realizan los usuarios por un lado y por otro los programadores tendrían más tiempo para dedicarlo a los algoritmos fundamentales de la aplicación.

Por otra parte, una de las cargas más fuertes en un sistema "main frame", son los programas de captura y corrección en línea de grandes archivos de datos. Por lo tanto, una solución que se plantea para este problema es el desarrollo de un sistema, mediante el cual de manera sencilla y rápida se pudiera definir, el formato del archivo y la pantalla de captura y/o corrección en un microcomputador PC¹, se procesará la información en este y posteriormente se enviarán los archivos al sistema central para los diversos procesos de aplicación. Si posteriormente se requirieran, podrían transferirse porciones del archivo del "main frame" al microcomputador para modificaciones de la información, y al terminar regresarse al "main frame" los registros modificados para actualizar el archivo original.

1.2 Objetivo del Sistema

El objetivo de este proyecto es la realización de un sistema que ayude a las tareas tanto de desarrollo como de mantenimiento de sistemas, mediante el uso de un microcomputador PC.

Este sistema estará enfocado al mantenimiento de archivos, que permitirá la creación y/o edición de archivos de manera sencilla e inmediata, de tal forma que el usuario final pueda utilizarlos sin necesidad de programación; además contemplará la comunicación entre el equipo PC y el "main frame" con el objetivo de poder transferir información entre los dos equipos.

1 Microcomputador tipo PC compatible con IBM.

CAPITULO 2

ANALISIS DEL SISTEMA

El análisis del sistema es la primera fase de toda una metodología para el desarrollo de un sistema, en la cual se basa el presente trabajo. Es por esto, que primero se hará una breve descripción de esta metodología, para continuar con lo que es el análisis del sistema.

2.1 Metodología de Sistemas

Para desarrollar un sistema, las necesidades y limitaciones de los usuarios deben estar determinadas y explícitamente establecidas, el sistema debe diseñarse considerando tanto a los instrumentadores como a los usuarios, y a quienes les den mantenimiento. Las tareas de mantenimiento de "software" muchas veces requieren rediseño y modificación del código, actualización de documentación que muestren los cambios, etc. Es por ello, la necesidad de un enfoque sistemático para el desarrollo y mantenimiento de "software".

Aunque muchos sistemas han sido desarrollados adecuadamente, gran parte de estos sistemas quedan sujetos a sobrecosto, entrega tardía, poca confiabilidad, ineficiencia y poca aceptación por parte de los usuarios.

Además, conforme la tecnología ha ido evolucionando, haciendo los equipos de cómputo más complejos, la demanda por los productos de "software" ha crecido en mayor cantidad que la capacidad de producir y mantener dichos sistemas. Es por todo esto que surge una disciplina conocida como **Ingeniería de Software** la cual se define como el enfoque sistemático para el desarrollo, operación y mantenimiento de "software".

Para el desarrollo y mantenimiento de sistemas de información, se requiere el uso de técnicas de resolución de problemas, técnicas que sientan las bases de la planeación y administración de sistemas, análisis de sistemas, diseño metódico, instrumentación y mantenimiento continuo del sistema.

Existen diversas consideraciones sobre lo que es la planeación de un sistema de información, los pasos requeridos para planear un sistema de información, así como diversos modelos del ciclo de vida del desarrollo de un sistema.

A continuación se presentan las fases de un modelo del ciclo de vida del desarrollo de un sistema¹:

- **Análisis del Sistema**
 - Planeación
 - Definición de Requisitos
- **Diseño del Sistema**
 - Diseño Externo
 - Diseño Interno
- **Instrumentación del Sistema**
 - Codificación
 - Depuración
 - Prueba de Módulos
- **Prueba del Sistema**
 - Integración
 - Aceptación
- **Mantenimiento del Sistema**
 - Mejoras
 - Adaptación
 - Ajustes

Análisis del Sistema

- Planeación:**
- Desarrollar un enunciado definitivo del problema.
 - Descripción de la situación actual.
 - Descripción de las restricciones del problema.

¹ Fairley, Richard, Ingeniería de software, p. 32.

-Justificar una estrategia de solución para el problema

- Requisitos:
- Identificar las funciones por realizar, las restricciones, el subsistema de equipo electrónico, el subsistema de información, y el subsistema de personal.
 - Determinar los objetivos y requisitos del sistema.

Diseño del Sistema

- Diseño Externo: -Concepción, planeación, especificación de las características del sistema. Estas características incluyen despliegues en pantalla, formato de reportes, definición de entradas y salidas, así como la estructura general del sistema.
- Diseño Interno: -Especifica la estructura interna, los algoritmos, la instrumentación de las funciones, las estructuras de datos, interacciones entre datos y funciones. Proporciona una guía para la programación.

Instrumentación

- Codificación: -Traducción de las especificaciones del diseño a código fuente. El código debe ser claro y sencillo como sea posible.
- Depuración: -La claridad del código fuente facilita la depuración, prueba y modificación, dado que estas actividades consumen una gran porción del presupuesto en la programación.
- Prueba de Módulos: -Cada unidad del programa debe ser lo suficientemente pequeña y modular, de tal manera que pueda ser probada totalmente en forma aislada por el programador que la desarrolla o modifica.

Prueba del Sistema

- Integración:** -Estrategia para integrar los componentes del "software" en un producto. La estrategia de integración dicta el orden en que los módulos deben estar disponibles, por lo que ejerce gran influencia en el orden en que se escriben, depuran y hacen las pruebas de unidad de los módulos.
- Aceptación:** -Las pruebas de aceptación implican la planeación y ejecución de pruebas funcionales y desempeño para verificar que el sistema realizado satisfaga sus requisitos.

Mantenimiento del Sistema

- Mejoras:** -Proporcionar nuevas capacidades funcionales, mejorar los despliegues al usuario y los modos de interacción.
- Adaptación:** -Cambios en el ambiente de procesamiento como puede ser el traslado de "software" a un equipo distinto.
- Ajustes:** -Corrección de fallas en el sistema. La corrección de problemas implica la modificación y revalidación del "software"

2.2 Metodología Utilizada para el Análisis

Generalmente el objetivo del análisis es desarrollar el modelo lógico de un sistema, para ello, el primer paso de todo análisis de sistemas consiste en entender la finalidad de dicho análisis. Algunas de las razones para emprender un análisis de sistemas son las siguientes:

- Puede suceder que el actual sistema no esté funcionando como se esperaba lo que implica corregir alguna anomalía.
- Otra razón podría consistir en que han surgido nuevas necesidades o disposiciones en la organización. Independientemente de la causa que de origen a la nueva necesidad, el análisis de sistemas identificará las modificaciones o adiciones que deben hacerse al sistema de información con el fin de que la empresa pueda satisfacer dicha necesidad.

Prueba del Sistema

- Integración:** -Estrategia para integrar los componentes del "software" en un producto. La estrategia de integración dicta el orden en que los módulos deben estar disponibles, por lo que ejerce gran influencia en el orden en que se escriben, depuran y hacen las pruebas de unidad de los módulos.
- Aceptación:** -Las pruebas de aceptación implican la planeación y ejecución de pruebas funcionales y desempeño para verificar que el sistema realizado satisfaga sus requisitos.

Mantenimiento del Sistema

- Mejoras:** -Proporcionar nuevas capacidades funcionales, mejorar los despliegues al usuario y los modos de interacción.
- Adaptación:** -Cambios en el ambiente de procesamiento como puede ser el traslado de "software" a un equipo distinto.
- Ajustes:** -Corrección de fallas en el sistema. La corrección de problemas implica la modificación y revalidación del "software"

2.2 Metodología Utilizada para el Análisis

Generalmente el objetivo del análisis es desarrollar el modelo lógico de un sistema, para ello, el primer paso de todo análisis de sistemas consiste en entender la finalidad de dicho análisis. Algunas de las razones para emprender un análisis de sistemas son las siguientes:

- Puede suceder que el actual sistema no esté funcionando como se esperaba lo que implica corregir alguna anomalía.
- Otra razón podría consistir en que han surgido nuevas necesidades o disposiciones en la organización. Independientemente de la causa que de origen a la nueva necesidad, el análisis de sistemas identificará las modificaciones o adiciones que deben hacerse al sistema de información con el fin de que la empresa pueda satisfacer dicha necesidad.

-Una tercera razón puede ser el caso de implantar una nueva idea o una técnica diferente.

-Por último, el análisis puede emprenderse simplemente con la intención de encontrar la forma de hacer mejor lo que ya se está haciendo, muchos de los sistemas de procesamiento de datos e información actualmente en uso fueron diseñados e instalados hace muchos años. En numerosos casos, han dejado de ser válidas las razones que indujeron a diseñar los sistemas en una forma determinada. Los objetivos generales del mejoramiento de sistemas pueden definirse como reducción de costos, mejor servicio a los usuarios e información más rápida.

Cualquiera que sea la razón por la que se realiza el análisis del sistema, esta fase comprende básicamente dos puntos a desarrollar: la planeación del sistema y la definición de requisitos.

El primer paso en la planeación de un sistema es preparar, en la terminología del usuario, un enunciado breve del problema que se solucionará y de las restricciones que existen en su resolución. El enunciado debe incluir una descripción de la situación actual y de las metas que debe lograr el nuevo sistema.

La definición del problema requiere un entendimiento claro del problema y del entorno de éste. Las técnicas para obtener este conocimiento, por parte del analista, son entrevistas con el usuario, cuestionarios y observación especial de las tareas que presentan problemas.

El segundo paso en la planeación de un sistema es determinar la solución apropiada. Un sistema además de ser eficaz en términos de costo, debe aceptarse social y políticamente. Para ser eficiente en costo, un sistema, debe proporcionar las mismas facilidades, usando menos tiempo y personal, o proporcionar facilidades e información que antes eran inaccesibles.

Para determinar la solución más apropiada, es recomendable desarrollar primero una estrategia de solución, esta no es un plan detallado sino un enunciado general sobre la naturaleza de las posibles soluciones.

Se deben considerar varias estrategias de solución antes de elegir alguna. La estrategia seleccionada proporciona un marco de referencia para el diseño y la instrumentación del sistema. La factibilidad de cada estrategia de solución propuesta, se puede establecer por el análisis de restricciones de la solución. Una estrategia de solución es factible si las metas y requisitos del sistema se pueden satisfacer dentro de las restricciones de tiempo disponible, recursos y tecnología por medio de esta estrategia.

En cuanto a la definición de los requisitos, se refiere a la especificación total de los requerimientos técnicos del sistema de una manera concisa y sin ambigüedades.

2.3 Etapas para el Desarrollo del Análisis del Sistema

Para llevar a cabo el análisis del sistema es necesario efectuar una serie de estudios; la selección de la forma de realizar dichos estudios es importante, puesto que los resultados obtenidos permitirán conocer con exactitud los problemas y requerimientos del sistema de procesamiento actual, lo que llevará posteriormente a proponer la solución más adecuada. Es por ello que se presentan a continuación los puntos a desarrollar para cumplir con el objetivo del presente capítulo.

- Análisis del sistema actual
 - Descripción del sistema actual
 - Restricciones del sistema
 - Requerimientos del sistema
- Justificar una estrategia de solución para el problema
 - Esbozar varias estrategias de solución considerando las restricciones
 - Seleccionar la estrategia de solución más óptima

2.3.1 Análisis del Sistema Actual

2.3.1.1 Descripción del Sistema Actual

Como ya se mencionó anteriormente, la Dirección General de Servicios de Cómputo para la Administración cuenta con un departamento de captura de datos, en el cual se captura información de las diversas dependencias, instituciones, facultades etc. de la U.N.A.M. que solicitan este servicio.

El sistema de captura se encuentra instalado en equipos microcomputadores CROMEMCO y fué desarrollado en lenguaje ensamblador Z-80.

A continuación se describirá la organización del sistema de captura (figura 2.1), el cual incluye, principalmente la organización administrativa que se tiene desde la entrega de

ORGANIZACION DEL SISTEMA DE CAPTURAS

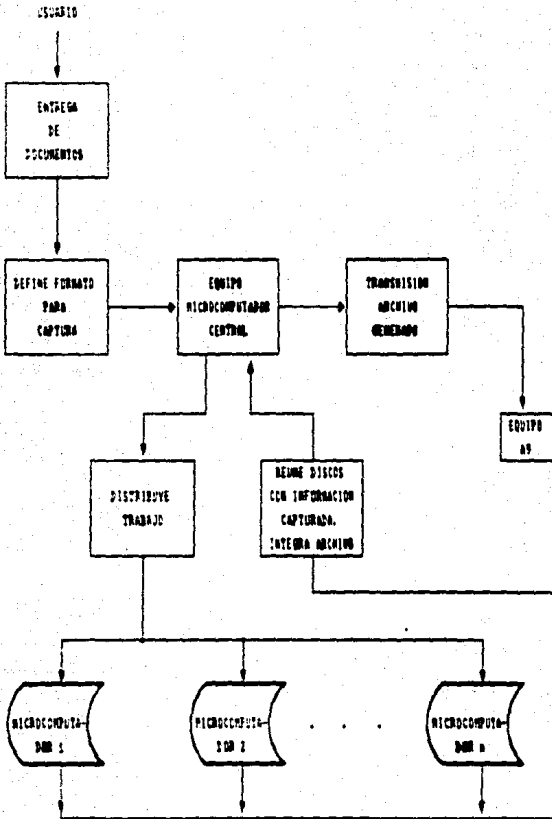


FIGURA 2.1

documentos y distribución de ellos entre diversas capturistas, hasta la entrega del archivo con los datos captados, listo para usarse por parte del usuario.

- El usuario entrega sus documentos fuente donde viene especificado el formato a seguir en la captura, y que en cierto modo corresponderá al formato que se deberá contemplar en el archivo de salida.
- Existe un equipo central en el cual se definen el formato, tipos de validación y características propias de los datos que se deberán seguir en la captura.
- Una vez que se definió el archivo de formato, este se distribuye entre las distintas capturistas que realizarán este trabajo.
- Ya que se han generado los distintos archivos, la persona que realizó la distribución de trabajos, centralizará todos los archivos generados a un equipo central, donde se generará un solo archivo.
- Por último, esta misma persona realiza la transferencia del archivo generado al equipo AS, en la clave del usuario, listo para ser usado.

Descripción del Sistema de Captura

El sistema consta de dos partes; sistema de captura de la información en el microcomputador CROMEMCO y el sistema de transferencia de archivos al equipo central.

Este sistema involucra a dos tipos de usuarios; los que capturan la información en grandes cantidades en el microcomputador y los que hacen uso de esta información ya en el equipo "main frame", desarrollando programas para aplicaciones diversas.

A continuación se describen brevemente cada uno de las partes del sistema de captura.

Sistema de Captura de Información en el Equipo Microcomputador CROMEMCO

Módulo Principal

Contiene el menú, el cual nos permite acceder cualquier módulo del sistema o bien terminar el proceso.

Módulo de Generación de Formato

Proporciona la facilidad de definir la pantalla de captura, es decir la distribución de los campos definiendo sus respectivos atributos, como son tipo de dato, lugar que ocupa en el registro de salida, etc.

Módulo de Captura

Permite capturar información con un formato previamente definido, así como también corregir cualquier campo en el registro presente, para esto se proporciona el número de campo o bien el nombre, con lo cual el cursor se posiciona al inicio del campo a corregir para poder teclear el dato correcto. Otras operaciones con las que cuenta este módulo son duplicar un registro, reiniciar la captura de un registro, definir constante un campo con lo cual en los registros futuros el campo definido como constante será llenado automáticamente.

Para poder realizar alguna corrección o modificación a un registro anteriormente capturado, se tiene que terminar el proceso o bien interrumpirlo para hacer uso del editor del microcomputador CROMEMCO en el que por supuesto no se hará ningún tipo de validación sobre la información modificada, lo que no garantiza que los datos sean correctos.

Módulo de Verificación

Con este módulo se puede verificar la información que haya sido captada con anterioridad; los campos que se verifican solo serán aquellos que hayan sido definidos con verificación, en el módulo de generación de formato. Este tipo de verificación se lleva a cabo volviendo a teclear todos los campos con dicho atributo, lo que significa realizar nuevamente la captura.

La edición en el microcomputador CROMEMCO solo se puede realizar con la información que no ha sido enviada al equipo central. Los archivos que se transfieren al equipo central no pueden ser editados nuevamente en el microcomputador; por lo tanto si se requiere de alguna modificación o corrección a un registro, los usuarios tienen que acceder sus archivos a través de una terminal del equipo AS para modificar sus registros con el editor de CANDE. Y debido a la manera en que la máquina almacena la información, las correcciones que tengan que hacerse no puede realizarse inmediatamente, teniendo entonces que desarrollar programas con este propósito.

Sistema de Transferencia de Archivos al Equipo Central

Para llevar a cabo el proceso de transferencia de archivos se tiene un equipo CROMEMCO conectado al equipo A9. Una vez que se han terminado los diversos archivos de un trabajo de captura, se procede a la centralización de estos, es entonces cuando la persona encargada de este equipo ejecuta el programa de transferencia y únicamente es necesario proporcionarle el nombre del archivo en el "diskette" del microcomputador.

Este programa se encarga de correr otro programa en el equipo A9, el cual lleva a cabo la instalación de los archivos de datos en el equipo central.

2.3.1.2 Restricciones del Sistema

En base a la información obtenida y observación del funcionamiento del sistema actual podemos citar los siguientes problemas:

-La restricción principal es el equipo en el cual se encuentra funcionando el sistema. El equipo microcomputador está limitado únicamente a las tareas de captura, ya que, aunque cuenta con recursos para desarrollo de aplicaciones, esto no es factible debido a que como equipo de cómputo es obsoleto.

Con la tecnología moderna, los cambios son más rápidos, y los equipos que permanecen mayor tiempo en producción son los que son compatibles con los nuevos productos que se van desarrollando. En el caso del equipo CROMEMCO, las fallas se presentan con mayor frecuencia, lo que ocasiona una disminución en la producción ya que los equipos permanecen mayor tiempo en mantenimiento. Además debido principalmente a que su fabricación ya no se realiza su mantenimiento es cada vez más difícil y de mayor costo.

-Otro punto importante a considerar en el sistema actual es que no se cuenta con un editor que permita modificar información en cualquier momento, lo que ocasiona tanto una pérdida de tiempo al usar otros programas para ello, así como que la información no sea validada.

-Una vez que la información capturada en el equipo microcomputador ha sido enviada para su proceso al equipo central, si se deseara modificarla sólo sería con herramientas que funcionan en ese equipo, tal como lo es el editor de CANDE, pero debido a la forma en que se encuentra dispuesta la información, muchas veces esto no es posible, teniendo entonces que crear programas que permitan modificar los archivos. Lo anterior ocasiona pérdidas de tiempo en el desarrollo de dichos programas así como el no poder disponer de manera inmediata de la información.

2.3.1.3 Requerimientos del Sistema

Como resultado del análisis llevado a cabo, se presentan a continuación los puntos más importantes a considerar en el nuevo sistema, los cuales comprenden los problemas, tanto del usuario de la información como del personal que la captura.

- Considerar el equipo en el cual se implementará el nuevo sistema, ya que como se vió, en el que funciona actualmente es ya inadecuado, principalmente por todos los problemas de mantenimiento que presenta.
- Contar con un nuevo sistema de creación y edición de archivos que agilice por un lado las operaciones de captura, contemplando la facilidad de creación de formatos y criterios de validación óptimos, de tal manera que se garantice la validéz de los datos de acuerdo a como hayan sido definidos por el usuario, evitando realizar verificaciones, lo que representaría mayor tiempo en la captura. Por lo tanto el sistema debe contemplar:

- La facilidad de generación de formatos, de acuerdo al documento original que permita definir la posición de los campos en la pantalla de captura, para que esta se lleve a cabo en forma más rápida.

- El uso de diferentes formatos en la captura de un trabajo, estableciendo el número de registros a capturar con cada uno de estos. Un ejemplo del uso de diferentes formatos es el siguiente:

En la captura de información bibliográfica se tienen diferentes campos, como son: nombre del autor, título de la obra, clasificación, edición, editorial así como la descripción en forma breve del contenido. La mayoría de los datos son de longitud constante, sin embargo, en este caso, la descripción del contenido es de longitud variable por lo que el trabajo se podría capturar con dos formatos; uno que contenga los campos de longitud constante y que se capture una solo una vez, y otro con lo que se capture la descripción del contenido del libro, el cual se repetirá tantas veces como sea necesario, pudiendo seleccionar nuevamente el primer formato para iniciar la captura de la información de otra obra.

- La definición del lugar que ocupará el campo en el registro de salida, independientemente de la ubicación que tenga en la pantalla de captura.
- De acuerdo al tipo de información que se captura se requieren validaciones de los siguientes tipos:

.Alfabética

.Númerica (considerando rangos)

.Alfanumérica

.Validaciones Especiales

Como es el RFC, (registro federal de causante), caracteres especiales y cadenas de caracteres definidas por el usuario

.Numérico UNIVAC y Alfanumérico UNIVAC

Debido a que inicialmente la captura se realizaba en equipos univac en donde las teclas numéricas tenían una ubicación diferente a al que se tiene en un equipo microcomputador, y debido a que la mayoría del personal está acostumbrado a este teclado por el tiempo que ha estado trabajando con el, no se puede imponer el uso de un teclado diferente, ya que esto repercutiría en la productividad del personal. Por lo tanto se tendrá la opción de este tipo de teclado además del que dispone el equipo.

. Así también, se observó que en la información que se captura, con frecuencia algunos datos se repiten, ya sea en algunos registros o en la totalidad del trabajo, datos como pudieran ser el número de un folio, la clave de una dependencia, etc.; por lo que además de las funciones básicas de edición como son borrar caracteres, campos y registros, insertar y sobrescribir caracteres, se hace necesario contar con las siguientes facilidades:

.Duplicar un registro del formato que se especifique

.Duplicar campos del formato que se especifique

.Definir campos constantes.

Esto es una forma de duplicar información, pero difiere de la anterior ya que en la duplicación de campos y registros, esta operación se realiza cuando lo indica el usuario, en cambio, en la definición de campos constantes la duplicación es automática. Esta característica se define en la generación del formato, indicando de que formato se obtendrá la información

del campo, o bien se tendrá la alternativa de definirlo durante la captura.

- Por lo que se refiere a la edición de los archivos se deberá tener una herramienta con amplias facilidades especialmente dirigidas al usuario final, que le permitan corregir o modificar su información, es decir, borrar e insertar tanto caracteres como registros, además de las operaciones propias de la captura. La recuperación y acceso a los registros se hará a través de una llave, la cual será definida por el usuario, esto permitirá el acceso aleatorio del archivo de datos determinando el criterio de selección de los registros, todo esto desde el equipo microcomputador.
- El sistema microcomputador deberá encargarse de realizar todas las tareas concernientes al mantenimiento de archivos, con lo cual los recursos del "main frame" serían destinados a los procesos de aplicación.

2.3.2 Justificar una Estrategia de Solución para el Problema

2.3.2.1 Esbozar Varias Estrategias de Solución Considerando las Restricciones

Actualmente existen en el mercado diversos programas con gran variedad de facilidades para la edición de archivos, sin embargo, estos editores están integrados a paquetes con propósitos más amplios.

Ejemplos de estos paquetes son el manejador de bases de datos dBASE y FRAME-WORK, los cuales analizaremos a continuación de acuerdo a los requerimientos ya identificados anteriormente.

dBASE

dBASE es un manejador de bases de datos que nos permite crear, actualizar y obtener información de una base de datos, a través de un conjunto de comandos. Así también, nos proporciona la facilidad de generación de formatos (pantallas de captura) de forma tal que la información a capturar, editar o desplegar se haga de un forma más clara.

Cuando se genera una base de datos, se especifican los diferentes tipos de campos que la conformarán, en particular dBASE permite cinco tipos de campos:

Campos de caracter/texto: almacenan letras, números, algunos símbolos especiales o espacios en blanco.

Campos de fecha: permite introducir una fecha con el formato mes/día/año.

Campos numéricos: almacena números con o sin decimales.

Campos lógicos: consisten en una simple letra representando un valor de verdadero o falso.

Campos memo: almacenan grandes bloques de texto por registro.

Aún cuando dBASE cuenta con estos diferentes tipos de campos, no satisfacen los criterios de validación que demandan los trabajos que se realizan en el departamento de captura.

Otro punto relevante a mencionar es, que el personal dedicado a la generación de pantallas y captura de datos necesitaría tener un mayor conocimiento, es decir, deberá conocer y manejar aquellos comandos de dBASE que le permitan generar las pantallas y capturar la información, para lo cual se requeriría capacitar al personal.

Además de que este paquete carece de algunas otras facilidades que permiten agilizar el trabajo, tales como el duplicar un registro, tener campos constantes, etc., lo cual podría solucionarse con el desarrollo de un programa en dBASE, pero esto implicaría no sólo capacitar al personal sino darle una formación como programador, para realizar cualquier tipo de trabajo que se presente.

Con respecto a la transferencia de archivos del microcomputador al equipo central no es posible realizarla con este programa, por lo que se requeriría el uso de otro programa para este propósito.

El archivo de datos en este caso, tiene además de los registros capturados, información referente a la estructura que se definió en DBASE, esta información deberá ser eliminada de dicho archivo para poder ser transferida. De manera inversa, si se quiere transferir un archivo del equipo A9 al microcomputador para su edición, no se podrá llevar a cabo si el archivo carece de la información acerca de la estructura que guarda este.

FRAMEWORK

FRAMEWORK es un paquete de "software" integrado que proporciona las funciones de uso más común en el ambiente administrativo, tales como: procesador de palabra, hoja de cálculo, bases de datos, gráficas y comunicaciones.

La base de datos es uno de los puntos de análisis en este paquete. Este paquete permite la definición de campos de un archivo con sus respectivos atributos. Los tipos de campo que maneja son sólo tres, tipo carácter, numérico y fórmula; este último es un campo que es calculado es decir, se calcula usando otros campos de la misma base de

datos o aún de diferentes bases de datos. Por otra parte, es posible realizar las operaciones básicas sobre las bases de datos (altas, bajas y cambios) con las validaciones correspondientes.

Con respecto al módulo de comunicaciones este paquete permite la comunicación del microcomputador con otro computador a través de la línea telefónica, además de permitir la conexión hacia otros servicios tales como "Dow Jones", "CompuServe", "Telemail". A través del menú de comunicaciones se pueden dar todos los parámetros para iniciar una llamada, incluyendo el número de teléfono y los parámetros para la comunicación.

Este módulo permite el envío y recepción de archivos, así como también la emulación de diferentes terminales. Cuenta además con importación y exportación de datos, es decir permite importar datos desde otros paquetes, así como también exportar hacia otros paquetes tales como Lotus 1,2,3, dBase II y dBase III.

Por último, podemos decir que si bien FRAMEWORK nos proporciona amplias facilidades en cuanto a la comunicación se refiere, está notablemente limitado en el aspecto de los criterios de validación que permite.

El hecho de que se hayan analizado las herramientas anteriores no quiere decir que sean las únicas en el mercado, sino que son de fácil acceso ya que se tienen actualmente en la universidad; también existen otros equipos especializados para captura, pero debido a su costo requieren mayores recursos haciendo más difícil su adquisición.

2.3.2.2 Selección de la Estrategia de Solución Óptima

Generalmente, el "software" disponible en el mercado ha sido diseñado para resolver problemas estándar que frecuentemente se presentan en las empresas, debido a ello no dan solución a problemas específicos que puedan existir.

Con frecuencia es necesario preferir el desarrollo de un sistema que proporcione la solución adecuada.

En este caso se tomó la determinación del desarrollo de un nuevo sistema, ya que si bien existe "software" altamente eficiente para proporcionar una ayuda parcial, no existe un producto que englobe total y eficientemente la solución a los diferentes aspectos que se han planteado.

Por otra parte se decidió implementar el nuevo sistema en un equipo microcomputador PC, ya que como se expuso anteriormente la obsolescencia del sistema radica también en el equipo en el cual está funcionando actualmente.

Las ventajas que proporciona el equipo PC son:

- Una mayor velocidad en los procesos
- Mayor capacidad que permita realizar más procesos disminuyendo de esta manera la carga de trabajo al equipo central.
- Proporcionará además características de "software" y "hardware" que faciliten en gran medida la comunicación entre los equipos.
- Compatibilidad, lo que facilitará que en caso de falla del equipo el sistema sea transportable a equipo semejante que evitará bajas en la productividad.
- Facilidad de expansión en "software" y "hardware" que permita satisfacer requerimientos futuros.
- Su mantenimiento no presenta mayor dificultad ya que se trata de equipo con gran disponibilidad actualmente en el mercado.

CAPITULO 3

DESCRIPCION DEL EQUIPO

Cuando se va a realizar un diseño es necesario conocer todos los elementos con que se cuenta tanto de "software" como de "hardware", sus características, capacidades, etc.. Generalmente esto redundará en el mejor aprovechamiento de los recursos. Es por ello que en el presente capítulo se describen las características del equipo "main frame" así como las del microcomputador.

También se hablará acerca del lenguaje utilizado en la programación. Generalmente la selección de este, está en función del tipo de problema de que se trate. El conocimiento de un lenguaje lleva a saber si tiene o no las características y herramientas que se requieren para la programación del sistema.

El uso del lenguaje ensamblador proporciona un medio muy flexible para casi todos los problemas de programación, mientras que los lenguajes de más alto nivel son más orientados a problemas específicos. La selección del lenguaje adecuado se basa en la compatibilidad del lenguaje con el problema en consideración.

Otros puntos que contribuyen a la selección del lenguaje son: su popularidad, la posibilidad de interfase con otros lenguajes, la claridad con la que el lenguaje puede autodocumentarse, además por supuesto de la facilidad en la programación.

3.1 Equipo PC

3.1.1 Software del Equipo PC

Sistema Operativo: MS-DOS (Microsoft Disk Operating System)/IBM-PC

El Sistema Operativo MS-DOS desarrollado por Microsoft Corporation, es un Sistema Operativo de disco para un sólo usuario y computadoras basadas en los microprocesadores 8086/8088.

El Sistema Operativo es el programa más importante en un sistema de computación, siendo usualmente el más complejo. El propósito de un Sistema Operativo es administrar los recursos del sistema:

- Procesadores
- Memoria
- Periféricos
- Información (Programas y Datos)

El Sistema Operativo es el sistema de "software" más íntimamente relacionado con el "hardware" y su diseño es únicamente para un procesador en específico.

Estructura del Sistema Operativo

Un disco con formato compatible con el Sistema Operativo MSDOS está dividido en cuatro regiones de datos, definidas como sigue:

- 1.- Región de carga del Sistema ("BOOT").
- 2.- Región de tablas de asignación de archivos (FAT).
- 3.- Información del directorio.
- 4.- Archivos del Sistema, aplicaciones y datos.

El programa de carga ('BOOT'), llamado 'BOOTSTRAP' es el responsable de cargar los archivos del Sistema Operativo en la memoria principal de la máquina en alguna localidad determinada y se encuentra siempre en el primer sector del disco que contiene al Sistema Operativo.

El Sistema de Tablas de Asignación de Archivos (FAT), es simplemente una tabla que utiliza el Sistema Operativo para registrar que sectores del disco están siendo utilizados para almacenar archivos, que sectores están disponibles para asignarlos a archivos, y por último que sectores se encuentran dañados y no son utilizables. El FAT es una lista ligada que mantiene un mapa de todos los sectores ocupados por el archivo.

Las entradas en el directorio son de 32 "bytes" de longitud. La información contenida en ellos se refiere a la localización del archivo en el disco, la fecha de la última modificación que se le hizo, atributos del archivo, etc.

Por último se tiene el área disponible para archivos, ya sean del sistema, de aplicación o datos.

Los archivos correspondientes al MS-DOS son los siguientes:

- IO.SYS
- MSDOS.SYS
- COMMAND.COM

El archivo IO.SYS, es lo que se conoce como el sistema básico de entrada/salida (BIOS). Este archivo proporciona las interfaces con el "hardware" requeridas por el Sistema Operativo para efectuar el "BOOTSTRAP". Además contiene el vector de interrupciones de los manejadores para soportar el video, el teclado, la impresora, discos, etc.

El archivo MS-DOS.SYS será el encargado de efectuar las tareas del Sistema Operativo en sí. Esta es la parte que se encarga de hacer llamadas a rutinas, accesos a archivos y manejo de recursos.

Por último, el archivo COMMAND.COM contiene el procesador de comandos o "SHELL", el cual está formado de una parte residente y otra transitoria, y es el encargado de interpretar los comandos que proporcionamos al computador y que pueden ser de uno de los siguientes cuatro tipos:

- Comandos Internos

- Comandos Externos
- Comandos "Batch"
- Comandos de configuración

Los comandos internos algunas veces llamados intrínsecos, son aquellos que pueden ser ejecutados inmediatamente ya que el código se encuentra en el mismo COMMAND.COM. Para reconocer los comandos internos COMMAND.COM tiene una tabla de nombres de comandos.

Los comandos externos también llamados extrínsecos, programas transitorios o utilerías, son los nombres de programas almacenados en disco y deben ser cargados cada vez que se ejecutan. Entre estos se encuentran todos los de extensión .EXE, .COM y .BAT.

Los comandos "batch" son archivos de texto que contienen listas de comandos intrínsecos o extrínsecos.

Los comandos de configuración definen ciertas características de operación del Sistema Operativo cuando es cargado. Estos comandos solo pueden estar contenidos en un archivo de configuración, al cual el sistema conoce como CONFIG.SYS.

El resultado final del proceso de carga del Sistema Operativo en memoria se muestra en el diagrama 3.1.

Organización de la Memoria en el Equipo PC

En el equipo PC se tienen localidades de memoria de 8 "bits" o un "byte" en tamaño; el procesador 8086 tiene un espacio de direccionamiento de 1024 "kbytes", en donde cada dirección de memoria tiene una longitud de 20 "bits".

La mayoría de la aritmética que el 8086 puede realizar está limitada para manipular palabras de 16 "bits" y dado que las direcciones completas son de 20 "bits" se tiene que llevar a cabo un direccionamiento segmentado.

MAPA DE MEMORIA

0000-0000

VECTOS DE INTERRUPCIONES
AREA DE DATOS USADOS POR DOS
AREA DE DATOS USADOS POR DOS
IO.SYS (INTERFASE CON ROUTINAS DE L/S DE DOS)
USOSYS (NOMESADRES DE INTERRUPCIONES, ROUTINAS DE SER- VICIO).
"BUFFERS" DE DOS, AREAS DE CONTROL, "DRIVERS" INSTALADOS.
PARTE RESIDENTE DE COMMAND.COM (NOMESADRES DE INTERRUPCIONES PARA TERMINACION, ABRENO, ERROR CRITICO, Y CONTROL PARA RECARGAR LA PARTE TRANSITORIA).
COMANDO EXTERNO O UTILERIA (ARCHIVOS .EXE O .COM)
"STACK" DEL USUARIO PARA ARCHIVOS .COM
PARTE TRANSITORIA DE COMMAND.COM INTERPRETE DE COMANDOS, COMANDOS INTERNS, PROCESADOR "BATCA", Y CARGADOR DE COMANDOS EXTERNOS.

FIGURA 3.1

La memoria se encuentra distribuida de la siguiente manera:

00000-40000	-Memoria residente de la tarjeta del procesador. Contiene información fundamental de operación para la computadora. Vectores de interrupción. Rutinas de Control del BIOS. Sistema Operativo.
40000-A0000	-Memoria de expansión.
A4000-B0000	-Memoria reservada sin ningún propósito en especial.
B0000-B4000	-Bloque de memoria del video monocromático.
B8000-C4000	-Bloque de memoria del video de color. Ninguno de los adaptadores de video necesitan o utilizan totalmente los 32 "kbytes" que tienen destinados.
C8000-CC000	-Reservada para usos futuros del BIOS (controladores de disco duro).
CC000-EC000	-Reservada para usos futuros de la memoria ROM.
F0000-FFFFF	-Area destinada para la ROM-BIOS. La ROM-BIOS contiene todas las rutinas de servicios fundamentales que son construidas para la arquitectura específica del equipo.

Registros

Además de una memoria como lugar de almacenamiento de operaciones y cálculos, el microprocesador 8086/8088 utiliza registros para aumentar la velocidad en su trabajo, los cuales se mencionan a continuación:

- Cuenta con 4 registros generales de 16 "bits" conocidos como AX, BX, CX y DX. Cada registro puede ser dividido en dos registros de 8 "bits", usando la parte alta (H) o la parte baja (L) del registro completo.
- Se cuenta también con 4 registros de apuntadores y de índice SP, BP, SI y DI. SP es el apuntador al "stack", BP es el apuntador de base y SI y DI indican los registros índices fuente y destino respectivamente.
- Un registro de indicadores de 16 "bits" que contiene varios "bits" de estado para el procesador. Estos incluyen:

ZF- indicador de cero

REGISTROS PROGRAMABLES

AX	AH	AL	REGISTROS
BX	BH	BL	DE
CX	CH	CL	
DX	DH	DL	DATOS
PUNTERO DE PILA SP			REGISTROS
PUNTERO DE INSTRUCCION IP			
PUNTERO DE BASE BP			DE
INDICE FUENTE SI			
INDICE DESTINO DI			DIRECCIONES
SEGMETO DE CODIGO CS			REGISTROS
SEGMETO DE DATOS DS			DE
SEGMETO DE PILA SS			
SEGMETO EXTRA ES			SEGMENTOS
INDICADORES			REGISTRO DE ESTADO

FIGURA 3.2

SF- indicador de signo
PF- indicador de paridad
CF- indicador de acarreo
AF- indicador auxiliar
DF- indicador de dirección
IF- indicador de interrupción
OF- indicador de desbordamiento ("overflow")
TF- indicador de desvío ("trap")

-También estan los 4 registros de direcciones de segmentos, CS para el segmento de código , DS para el segmento de datos , SS para el segmento de "stack" y ES para el segmento extra suplementario.

Interrupciones

Una interrupción es una señal de un programa o dispositivo físico para el procesador. Las interrupciones suspenden momentáneamente la tarea que este desempeñando la unidad central de procesamiento para que pueda ejecutar otro requerimiento. Cada interrupción tiene una sección de código residente en memoria que ejecuta el procesador cada vez que tiene lugar la interrupción. Este código se llama gestor de interrupciones o rutinas de servicio de interrupciones.

El procesador localiza las rutinas de servicio de interrupciones por medio del vector de interrupciones, en este vector hay un conjunto de 4 "kbytes" que especifican el segmento de memoria y el desplazamiento sobre el segmento para localizar la rutina de atención de una interrupción. Este conjunto de 4 "kbytes" existe para cada interrupción disponible. Para el caso específico de las interrupciones a MS-DOS su vector de interrupciones se encuentra localizado en las direcciones 80H y FCH.

Las llamadas a rutinas del Sistema MS-DOS son hechas a través de la aplicación del "software" de interrupciones.

Las rutinas que MS-DOS utiliza para manejar la operación del sistema y sus recursos, pueden ser llamadas por cualquier programa de aplicación, facilitando la escritura de programas independientes de la máquina, incrementando la posibilidad de que un programa sea compatible con futuras versiones de MS-DOS. Las llamadas a MS-DOS pueden clasificarse de la siguiente manera:

- E/S de dispositivos estándar

- Manejo de Memoria
- Manejo de Procesador
- Manejo de Archivos y Directorios
- Llamadas a rutinas de Redes de Microsoft
- Misceláneas

El rango actual de interrupciones que maneja MS-DOS comprende desde la 10H a la 67H. La interrupción 21H, es la que da servicio a solicitudes de funciones, permitiendo el acceso a una amplia variedad de servicios del MS-DOS.

La interrupción 21H de MS-DOS, contiene una serie de funciones que también pueden ser llamadas por programas de usuario. Desde estas funciones podemos hacer el manejo de recursos del sistema. El procedimiento para su llamado es como sigue:

1. Mover la información requerida a los registros necesarios
2. Mover el número de función a AH
3. Mover el código de acción a AL si es requerido
4. Llamar a la interrupción 21H

Cuando MS-DOS toma el control, los registros que estaban siendo usados por el usuario son salvados en un "stack" interno, (excepto AX), hasta que la rutina termina y el control pasa nuevamente al programa del usuario.

EDLINE (Editor de Línea)

EDLINE es el procesador de líneas del Sistema Operativo, el cual nos permite crear o modificar archivos de texto.

Con EDLINE se puede:

- Crear archivos de texto
- Examinar el contenido de un archivo de texto
- Borrar, editar, insertar y desplegar líneas de un archivo
- Búsquedas para borrar, o reemplazar textos en una ó más líneas en un archivo

EDLINE divide todo el texto en líneas, y cada línea puede tener hasta 253 caracteres.

Secuencias de Escape ANSI

En versiones anteriores del Sistema Operativo, la portabilidad de programas de un computador a otro era un aspecto importante para tomarlo en consideración. Con la adición de diseños de "hardware" más elaborados, particularmente los referentes a gráficas, surgió la dificultad para los programas de aplicación de utilizar herramientas dependientes del "hardware" y mantener sin embargo, la compatibilidad entre sistemas. Para resolver esta situación, Microsoft provee la descripción de una terminal ANSI estándar con su distribución en el Sistema Operativo. Los programas pueden seguir o no dicha descripción.

Para tener acceso a los manejadores ANSI de consola y teclado, debe existir en el archivo CONFIG.SYS una línea como la siguiente:

```
DEVICE = \ANSI.SYS
```

Una secuencia de escape ANSI es una serie de caracteres (empezando con un caracter de escape) que pueden usarse para definir funciones a MS-DOS. Especialmente pueden reasignarse teclas, cambiar funciones gráficas, y afectar el movimiento del cursor.

3.1.2 Hardware del Equipo PC

Arquitectura del Microprocesador 8086/8088

El 8086/8088 son procesadores de propósito general de Intel. El 8086 es un microprocesador de 16 "bits", tanto en lo que se refiere a sus estructura como a sus conexiones externas, mientras que el 8088 es un procesador de 8 "bits" que internamente es idéntico al procesador de 16 "bits" 8086.

El 8086 y el 8088 son prácticamente idénticos excepto por el tamaño de "bus" externo de datos. Intel trata esta igualdad interna y desigualdad externa, dividiendo cada procesador 8086/8088 en dos subprocesadores. O sea, cada uno consta de una unidad de ejecución (EU: "Execution Unit") y una unidad interfase de "bus" (BIU: "Bus Interface Unit"). La unidad de ejecución está encargada de realizar todas las operaciones mientras que la unidad de interfase de "bus" está encargada de acceder a datos e instrucciones del mundo exterior. Las unidades de ejecución son idénticas en ambos procesadores, pero las unidades de interfase de "bus" son diferentes, la BIU del 8088 es más compleja, ya que

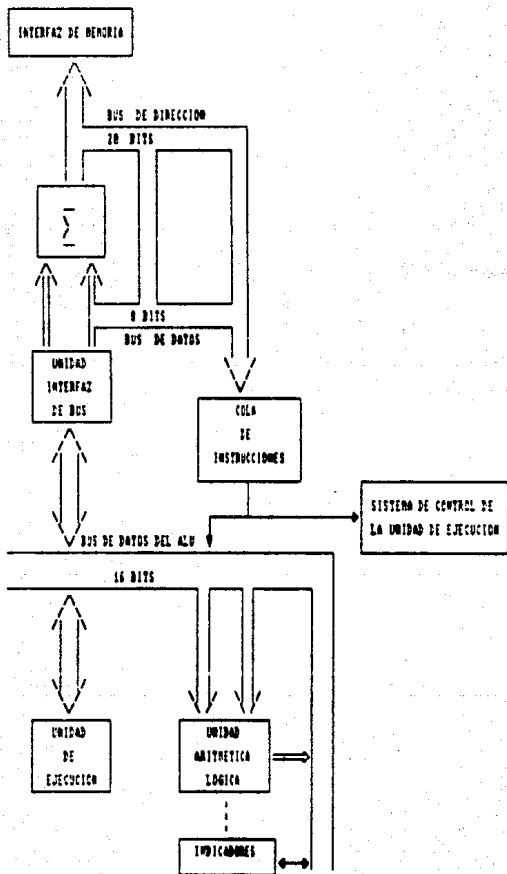


FIGURA 3.3

debe transferir datos entre el "bus" de datos interno de 16 "bits" y el "bus" de datos externo de 8 "bits".

Arquitectura "Pipeline"

Intel diseñó el procesador 8086/8088 para realizar al mismo tiempo las principales funciones internas de transferencia de datos y búsqueda de instrucciones. Para conseguirlo, tanto el 8086 como el 8088 constan de dos procesadores interconectados en la misma pieza de silicio.

Además la unidad encargada de buscar instrucciones utiliza un método llamado de estructura tubular "pipeline" o por cola para almacenar nuevas instrucciones hasta que se necesiten.

Principales Características del CPU (Central Processor Unit) 8086/8088

El microprocesador 8086/8088 puede conectarse al circuito de dos formas distintas: el modo máximo y el modo mínimo. El modo queda determinado al colocar una determinada terminal (llamada MN/MX) a tierra o a la tensión de alimentación.

El 8086/8088 debe estar en modo máximo si quiere trabajar en colaboración con el procesador de datos numérico 8087 y el procesador de entrada-salida 8089. En el modo máximo, el 8086/8088 depende de otros "chips" adicionales como el controlador de "bus" 8288 para generar el conjunto completo de señales de control de "bus". El modo mínimo permite al 8086/8088 trabajar de una forma más autónoma.

En ambos modos, las señales del 8086/8088 se pueden agrupar de la siguiente manera:

- Alimentación
- Reloj
- Control y Estado
- Direcciones
- Datos

Señales de reloj. El 8086/8088 requiere una única señal de reloj. Estos procesadores no generan su propia señal de reloj, dependiendo del generador de reloj 8284. Intel tiene una versión de 5 MHz y otra de 8 MHz para el 8086 y el 8088. Estas versiones representan las velocidades más altas, recomendadas para estos "chips". No se recomienda bajar de 2 MHz para ninguno de los dos.

las velocidades más altas, recomendadas para estos "chips". No se recomienda bajar de 2 MHz para ninguno de los dos.

Para un rendimiento óptimo, el microprocesador 8086/8088 requiere una señal de reloj que se mantenga a tensión alta una tercera parte del tiempo total del ciclo.

Alimentación. Hay tres terminales para la alimentación: tierra (GND) en las terminales 1 y 20, y una tensión de 5 volts (Vcc) en la terminal 40. La tierra es tierra a la vez para la alimentación de las señales.

Tanto para el 8086 como para el 8088 hay 20 "bits" de dirección. Los cuatro "bits" más significativos de la dirección comparten terminales con algunas de las señales de estado. En el 8086, los 16 "bits" menos significativos se comparten con los "bits" de datos, y en el 8088 sólo 8 "bits" menos significativos comparten terminales con los datos, debido a que el "bus" externo de datos es de 16 bits para el 8086 y de 8 "bits" para el 8088.

Multiproceso y procesamiento en paralelo. El 8086/8088 tiene unas características especiales, que permiten coordinar sus actividades con otros procesadores en contexto de multiproceso y de procesamiento en paralelo. El procesamiento en paralelo es un sistema en el cual dos ó más procesadores trabajan en la misma porción de un programa. Multiproceso es un sistema en el cual dos o más procesadores trabajan en diferentes programas, pero comparten la misma memoria.

Capacidad de direccionamiento. Tanto el 8086 como el 8088 tienen un "bus" de direccionamiento de 20 "bits" de amplitud, lo que provee una capacidad de direccionamiento de un "megabyte" de memoria (1,048,576 "bytes"). Sin embargo el registro de direccionamiento del 8086/8088 tiene únicamente una amplitud de 16 "bits". Esto es equivalente a 64 "kbytes". Este procesador usa un método llamado segmentación para direccionar a todo el "megabyte" de memoria.

El 8086/8088 tiene además otra memoria separada, llamada espacio de entrada/salida; en dicha memoria se encuentran los dispositivos de E/S. El espacio de E/S del 8086/8088 utiliza un direccionamiento de 16 "bits".

Modos de direccionamiento. El 8086/8088 tiene 25 modos de direccionamiento o reglas para localizar un operando de una instrucción.

Instrucciones. En general, las instrucciones del procesador de 16 "bits" se pueden clasificar en los siguientes grupos:

1. Transferencia de datos
2. Aritmética entera binaria
3. Operaciones lógicas

4. Desplazamientos y rotaciones
5. Manejo de "bits"
6. Aritmética codificada en binario
7. Manejo de cadenas
8. Control del programa
9. Control del sistema.

Controladores de Dispositivos y del Sistema

Controladores del Sistema

Ciertas funciones tales como las transferencias de bloques de información (transferencia DMA), ó el control de las interrupciones de E/S se realizan más eficazmente si son otros elementos independientes del CPU los que los lleven a cabo. Tales dispositivos reciben el nombre de controladores del sistema. La existencia de los controladores no solo hace posible ciertas funciones, sino que a la vez libera al CPU de la supervisión de ciertas tareas distribuyendo la inteligencia de la máquina a lugares externos del CPU.

A continuación se mencionan brevemente dos controladores del sistema, el controlador programable de DMA 8237 y el controlador programable de interrupciones 8259.

Controlador Programable DMA 8237

El controlador de DMA (Direct Access Memory) es un dispositivo capaz de controlar las transferencias directas de información de una parte a otra del sistema. Debido a que frecuentemente es necesario mover muchos datos, a velocidades mayores de las que se podrían obtener a través del CPU. Existe un tipo de memoria que sirve para almacenar temporalmente datos, conocida con el nombre de memoria intermedia ("buffer"). Este tipo de controlador es utilizado, cuando se quiere mover rápidamente una imagen, o parte de una imagen, de la memoria intermedia a la memoria, o cuando se necesita una transferencia rápida de los contenidos de las memorias intermedias de un disco flexible a otras posiciones.

Controlador Programable de Interrupciones 8259

El controlador programable de interrupciones PIC 8259 (Programable Interruption Controller) se utiliza en el manejo directo de las interrupciones de hasta ocho dispositivos distintos, de hasta 64 dispositivos si se conectan otros ocho "chips" 8259 juntos.

El PIC 8259 actúa a modo de recepcionista en un sistema en el cual cada dispositivo transmite alguna petición de interrupción al CPU. El PIC 8259 se encarga de que en cada momento, sólo una llamada le llegue al CPU, respetando ciertas prioridades.

El PIC determina a quién debe atender primero de acuerdo con ciertos esquemas, definibles para el usuario. Dichos esquemas incluyen una prioridad fija y una prioridad rotatoria. El dispositivo a quien no se atiende espera turno en una área de recepción manejada por el propio PIC. Cuando el dispositivo es atendido, el PIC lo saca de esta área de espera y lo lleva a la de trabajo.

Controladores de Dispositivos

En un sistema de cómputo existe un cierto número de dispositivos conectados al computador central, como son los teclados, monitores de videos, discos flexibles o impresoras. Tales elementos reciben el nombre de dispositivos de entrada/salida (E/S). Un dispositivo de entrada/salida nunca se conecta directamente al "bus" principal. Los controladores de dispositivo actúan de interfase entre los dispositivos y la computadora mientras que los controladores del sistema realizan funciones internas.

A continuación se describen brevemente algunos controladores:

Controlador Programable Serie de Interfase 8250

El controlador programable serie de interfase PSIC 8250 (Programable Serie Interface Controller) sirve de ayuda en la conexión a la computadora de dispositivos que envían la información "bit" a "bit"; esto es lo que se llama transmisión serie. La transmisión serie se prefiere sobre la transmisión paralela en aquellos casos en los que la información debe transmitirse a grandes distancias.

Puesto que la computadora trabaja internamente con los datos en paralelo, se necesita un controlador serie de interfase para pasar de paralelo a serie. Un dispositivo de este tipo recibe el nombre de receptor-transmisor asíncrono universal (UART).

Controlador Programable Paralelo de Interfase 8255

El controlador programable paralelo de interfase PPIC 8255 (Programmable Parallel Interface Controller) sirve de ayuda en la conexión a la computadora de dispositivos que envían "bytes" completos cada vez (o incluso palabras de 12, 16 ó 24 "bits"). La transmisión **paralelo** es útil en todas aquellas aplicaciones que requieran una transmisión a gran velocidad, y utilicen dispositivos no demasiado alejados del computador central. No hay **ninguna** sincronización especial en las transmisiones paralelo. Los "bytes" se envían tan rápidamente, o tan lentamente como lo permita el "software". La mayor velocidad se obtiene utilizando el DMA.

Controlador Programable de CRT 8275

El controlador programable CRT PCRTC 8275 (Programmable CRT Controller) proporciona la sincronización y el control de la visualización de texto sobre la pantalla. El texto se almacena en una sección de memoria reservada a este efecto y que recibe el nombre de RAM de visualización de texto. El CPU coloca el código ASCII digital de los caracteres del texto en esta RAM de visualización, y el PCRTC genera las señales de video necesarias para visualizar los símbolos correspondientes sobre la pantalla.

Controlador de Discos Flexibles de Simple/Doble Densidad 8272

El controlador de discos flexibles de simple/doble densidad FDC 8272 (Floppy Disk Controller), provee la circuitería de control para la interfase de hasta cuatro discos flexibles con una computadora basada en el 8080, 8085, 8086, 8088 y la mayoría de los microprocesadores.

El FDC puede trabajar solo o junto a un controlador de DMA.

El FDC 8272 tiene disponibles 15 órdenes básicas de control del disco, incluyendo órdenes de movimiento de la cabeza de lectura/escritura de una unidad determinada a una cierta pista, de transferencia de sectores de datos entre la unidad seleccionada y la computadora, búsqueda de sectores dependiendo de determinados formatos, formato de discos, y de definición de algunos parámetros de sincronización.

Interfases

Ya con anterioridad se describieron brevemente los principales componentes de "hardware" del sistema PC, por último, se mencionarán las interfases de "hardware" necesarias para la interconexión de equipo.

De acuerdo a la función de las señales que se envían y se reciben a través de los circuitos de interfase se pueden dividir en :

- Señales de Datos
- Señales de Control
- Señales de Tiempo

Con el fin de estandarizar la interfase (conector) a utilizarse para la interconexión, los fabricantes desarrollaron la interfase serial RS-232C y la interfase Centronics para transmisión en paralelo, las cuales especifican el estándar físico y lógico para la interconexión de dispositivos de comunicación.

3.2 Lenguajes de Programación

Actualmente existen diferentes lenguajes de programación, que debido al desarrollo de "software" que se ha dado proporcionan herramientas con bastantes facilidades para desarrollar sistemas de diferentes tipos y enfoques.

El lenguaje utilizado para el desarrollo de este sistema en el equipo PC es el Lenguaje C, el cual, fue seleccionado primeramente por las características que lo describen como un lenguaje capaz de soportar el desarrollo de sistemas, así como la portabilidad de estos.

Por otra parte, dado que los que desarrollamos este sistema desconocíamos totalmente este lenguaje, y sabiendo que se hace indispensable buscar y probar nuevas alternativas se decidió empezar a conocer este lenguaje a través del desarrollo de un sistema.

Se describirá primeramente el lenguaje C, el cual fué usado para programar el sistema en el microcomputador PC.

3.2.1 Descripción del Lenguaje C

C es un lenguaje de programación de propósito general, el cual fué inventado por Dennis Ritchie e implementado primeramente en una DEC PDP-11 que usó el Sistema Operativo UNIX. Con la popularidad de las microcomputadoras, fueron creadas un gran número de implementaciones en C y el código aceptado por la mayoría de estas implementaciones es altamente compatible.

C ha sido utilizado especialmente en la programación de sistemas debido a que permite escribir programas compactos y rápidos. En algunos casos, programas bien escritos en C son comparables en velocidad a otros programas en lenguaje ensamblador y ello ofrece la facilidad de mantenimiento así como la comprensión del código fuente. Sin embargo, este lenguaje también se ha utilizado para escribir programas numéricos, programas de procesamiento de textos y bases de datos.

C es llamado lenguaje de "bajo nivel" o de "nivel medio", lo cual no significa que sea un lenguaje menos poderoso, difícil de usar o menos desarrollado que otros de alto nivel, lo que significa es que C es similar al lenguaje ensamblador y que requiere en mayor grado de la atención del programador.

C combina elementos de los lenguajes de alto nivel con el funcionalismo del lenguaje ensamblador. Permite la manipulación de "bits", "bytes" y direcciones, tiene cuatro tipos de datos básicos y permite casi todo tipo de conversiones. C no realiza chequeo de errores en tiempo de ejecución, tales como, límites en los arreglos o compatibilidad en el tipo de los argumentos, estos son responsabilidad del programador.

En C los objetos fundamentales son caracteres, enteros de varios tamaños así como también números de punto flotante de simple y doble precisión. Además existe una jerarquía de tipos de datos derivados creados con apuntadores, arreglos, estructuras, uniones y funciones.

El principal componente estructural en C es la función, y es en la cual ocurre toda la actividad del programa, ellas permiten que las tareas puedan ser definidas y codificadas en forma separada permitiendo así que los programas sean modulares.

C no incluye funciones hechas para realizar tareas tales como entrada y salida, alojamiento en memoria, manipulación de la pantalla y manipulación de cadenas. Todos estos mecanismos de alto nivel son provistos por la biblioteca de funciones que acompaña la instalación de C.

C sólo ofrece proposiciones de control de flujo sencillas, secuenciales, de selección, de iteración de bloques y subprogramas, pero no multiprogramación, paralelismo, sincronización o corrutinas. Las construcciones fundamentales de control de flujo que permiten escribir programas estructurados: agrupamiento de sentencias, toma de decisiones (if),

ciclos, con comprobación de la condición de terminación al principio (`while`, `for`) o al final (`do`) y selección entre un conjunto de casos posibles (`switch`).

C además de ofrecer un amplio conjunto de operadores, incluye apuntadores y capacidad aritmética de direcciones. Se pueden declarar "apuntadores" a variables y funciones. Un apuntador a un elemento corresponde a la dirección de ese elemento. Usando apuntadores prudentemente puede incrementarse la eficiencia de los programas.

Cualquier función puede ser llamada recursivamente y sus variables locales son normalmente creadas de nuevo en cada invocación. Las definiciones de función no pueden anidarse, pero las variables pueden ser declaradas con una estructura de bloques. Las funciones de un programa en C pueden compilarse por separado. Las variables pueden ser internas a una función, externas (pero conocidas únicamente a lo largo de un sólo archivo fuente) y completamente globales. Las variables internas son automáticas o estáticas. Las variables automáticas pueden colocarse en un registro para incrementar su eficiencia, pero su declaración como registro es únicamente una indicación al compilador y no se refiere a los registros de la máquina.

El compilador de C dispone de un preprocesador con capacidades para la sustitución de macros, compilación condicional e inclusión de archivos. Las líneas que comienzan con `#` contienen instrucciones para este preprocesador. Su sintaxis es independiente del resto del lenguaje, pueden aparecer en cualquier lugar del programa y sus efectos perduran hasta el final del archivo fuente.

Características de Turbo C

El desarrollo del Sistema se realizó con la implementación de Turbo C. A continuación se describen algunas de sus características generales.

Turbo C se ejecuta en el Sistema Operativo MS-DOS de dos maneras. El comando `tc` ejecuta el compilador de C en un ambiente manejado por menus con un editor de texto. El comando `tcc` ejecuta Turbo C como un compilador solo y automáticamente llama el ligador si la compilación se ejecuta sin errores. Turbo C también proporciona el comando para llamar al ligador en forma separada.

El compilador tiene opciones para crear diferentes modelos de memoria para los programas y tiene llaves para especificar apuntadores de 16 y 32 "bits". Turbo C también soporta las características de ANSI C, además de que proporciona rutinas de interrupción al Sistema Operativo, código de ensamblado en línea y pseudo-variables.

Modelos de Memoria

Turbo C soporta seis modelos de memoria: "Tiny", "Small", "Middle", "Compact", "Large" y "Huge".

Modelo "Tiny"	Este modelo agrupa todas las áreas de programa en un segmento de 64K bytes
Modelo "Small"	Este modelo crea segmentos separados para texto y datos. Los apuntadores de texto y de datos son de 16 "bits", y el texto del programa y datos debe ser cada uno menor de 64K "bytes".
Modelo "Middle"	Este modelo tiene más de un segmento de texto pero solamente un segmento de datos. Los apuntadores de texto son de 32 "bits" y los apuntadores de datos de 16 "bits".
Modelo "Compact"	El modelo compacto tiene más de un segmento de datos pero solamente un segmento de texto. Los apuntadores de datos son de 32 "bits" y de 16 los apuntadores de texto.
Modelo "Large"	Este modelo tiene múltiples segmentos para texto y datos. Los apuntadores en este modelo son de 32 "bits", pero el compilador realiza aritmética de 16 "bits" para operaciones con apuntadores.
Modelo "Huge"	Este modelo tiene la misma configuración de memoria que el modelo "Large". Los apuntadores son de 32 "bits", y el compilador realiza aritmética de 32 "bits" para operaciones con apuntadores.

Soporte de Bajo Nivel

Turbo C tiene características especiales para operaciones de bajo nivel. En los programas de C pueden declararse funciones de interrupción y pueden aparecer también instrucciones en código ensamblador. Turbo C permite también acceder directamente el contenido de los registros de la máquina, mientras se está ejecutando el programa.

Instrucciones de Ensamblado en Línea

Turbo C maneja instrucciones en lenguaje ensamblador dentro de archivos fuente en C. Cuando se usa esta característica, el compilador invoca al programa Microsoft Macro Assembler (MASM).

Pseudo-Variables

Turbo C proporciona nombres especiales llamados pseudo-variables para acceder registros del CPU. Este tipo de variables pueden cargarse con valores o acceder datos de ellos como una variable de C. Por supuesto estas variables tienen limitaciones en cuanto al tipo de datos que pueden tener y si arbitrariamente se cambia el contenido de ciertas variables el programa puede fracasar. Además no se puede estar seguro de que un dato en una pseudo-variable será siempre válido, ya que el compilador puede cambiarlo cuando ejecute otras instrucciones. Y, ya que los registros no tienen una dirección, no se puede tomar la dirección de una pseudo-variable.

La biblioteca es especialmente importante en la programación en C porque ella provee las funciones básicas no provistas por el lenguaje. Turbo C tiene una gran variedad de funciones y macros que se pueden invocar dentro de los programas en C para realizar una amplia variedad de tareas, incluyendo E/S de alto y bajo nivel, manipulación de "strings" y archivos, alojamiento dinámico en memoria, control de procesos, conversión de datos y cálculos matemáticos entre otras.

3.3 Equipo A9

3.3.1 Hardware del Equipo A9

El sistema A9 utiliza una arquitectura con procesador de lógica múltiple (dentro de un solo CPU), mejorando la capacidad de un solo procesador por la técnica llamada "pipelining". Este concepto significa que varios submódulos dentro del CPU realizan, en paralelo, las varias etapas de ejecución de una instrucción. El CPU puede comenzar el procesamiento de la siguiente instrucción mientras completa el proceso para instrucciones previas.

El sistema A9 comprende los siguientes módulos principales

1. Unidad de Procesamiento Central.
2. Subsistema de Memoria.
3. Subsistema de Entrada/Salida
4. Subsistema de Comunicaciones.
5. Subsistema de Mantenimiento

Unidad de Procesamiento Central

Multiple Logical Processor (MLP)

El MLP considera que cada operador es un microprograma o tarea y divide estas tareas por medio del Sistema Operativo. El "pipeline" contiene tres procesadores lógicos, los cuales pueden ejecutar tres tareas concurrentemente. Los principales módulos del MLP son descritos abajo.

- Program Controller (PC). Examina el código objeto y establece los requerimientos y prioridades de las tareas.
- Task Controller (TC). Asigna y controla los recursos para llevar a cabo una tarea. El TPC mantiene el estado de "listo" y espera de las tareas, selecciona la siguiente tarea a ser ejecutada, suspende tareas cuando es necesario, y sincroniza la ejecución de las tareas.

DIAGRAM DE BLOQUES DEL SISTEMA

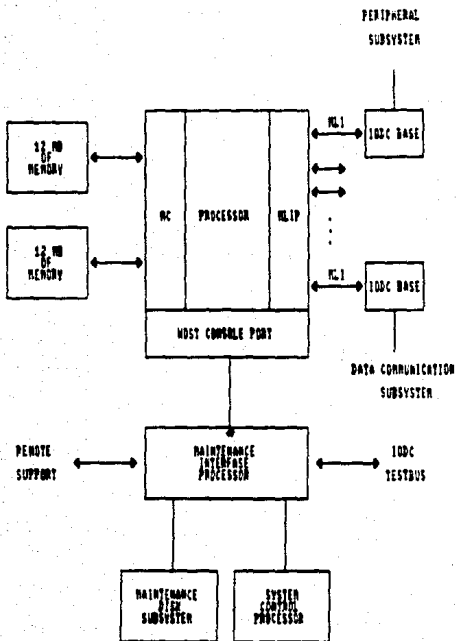


FIGURA 3.4

-Data Path (DP). Almacena datos para todos los operadores en ejecución y realiza las operaciones aritméticas y lógicas en estos datos.

-Address And State Unit (ASU). Convierte las direcciones relativas de memoria calculadas durante la compilación en direcciones absolutas asignadas en tiempo de ejecución.

-Stored Logic Controller (SLC). Es microcódigo en memoria, el cual contiene algoritmos usados para control y ejecución de diferentes partes del proceso. Cuando el SLC recibe un operador, localiza el algoritmo para ese operador y controla la ejecución de la etapa requerida. Debido a que tiene un SLC, la A9 es clasificada como un equipo con "E-mode".

Host Console Port (HCP)

El HCP es la interfase entre el procesador y la Ergonomic Work Station (EWS). Esta terminal es usada para la inicialización y mantenimiento del sistema.

Memory Controller (MC)

El MC es la interfase entre el procesador y el subsistema de memoria.

Subsistema de Memoria

El subsistema de memoria del equipo A9 es capaz de direccionar 6, 12, 18 o 24 MB de memoria. Todos los módulos de memoria se encuentran en un solo gabinete.

Los siguientes parámetros caracterizan el subsistema de memoria

Capacidad	4 M "words" o 24 M "bytes"
Longitud de palabra	60 "bits"
	48 "bits" - dato
	4 "bits" - "Tag"
	8 "bits" - Para manejo de errores

El subsistema de memoria consiste de memoria principal, memoria "cache", una Memory Change Queue (MCQ) y un Memory Read Request Buffer.

Memoria Principal

La memoria principal es un dispositivo de almacenamiento de acceso aleatorio de un número variable de palabras. Esta memoria es accesible solamente al MLP al cual está conectada.

Memoria "Cache"

La memoria "cache" es una memoria de alta velocidad que contiene copias de bloques de palabras recientemente accedidos en memoria. Una solicitud de lectura es satisfecha leyendo las palabras de la memoria cache, si el bloque requerido está presente ahí, se evita el tener que hacer un acceso a memoria principal. Sin embargo si el bloque apropiado no está presente, es traído de la memoria principal y reemplaza algún otro bloque de la memoria "cache" si es necesario.

Las operaciones de escritura a la memoria, se realizan siempre en la memoria principal y en la memoria cache, si el bloque apropiado está presente. Así, los datos en la memoria cache son siempre idénticos a los datos en la memoria principal y no es necesario actualizarla cuando el bloque en memoria cache es reemplazado.

Memory Change Queue (MCQ)

La MCQ es usada para poner en cola cambios en la memoria, es decir todas las operaciones de escritura.

Un operador inicia una solicitud para una operación de escritura a memoria generando una entrada en la MCQ.

Memory Read Request Buffer

Con el propósito de evitar retrasos innecesarios para cualquier solicitud de memoria, el subsistema está estructurado para permitir una solicitud de lectura desde cada tarea, estas solicitudes son almacenadas en el "buffer" de requerimiento de lecturas a memoria hasta que puedan ser satisfechas.

Subsistema de Entrada/Salida

Consiste de una serie de microprocesadores especializados (Data Link Processors) que son responsables de transferir información a y desde el subsistema de periféricos. Estos procesadores se encargan de atender a los periféricos, relevando así de esa responsabilidad al procesador central. Los DLP son también responsables de transferir información a y desde memoria via el Message Level Interfase (MLI).

Los DLP's son agrupados en módulos base I/O Data Communication (IODC). Cada módulo base IODC puede mantener hasta 8 DLP's. El sistema A9 puede incluir de uno a tres gabinetes de IODC dependiendo del número de módulos requeridos en la configuración.

Los módulos IODC son conectados por interfases de cables llamadas Message Level Interface (MLI) al Message Level Interface Processor (MLIP).

Los datos dirigidos a los dispositivos son transferidos desde el sistema central al MLIP, a través de MLI a los módulos IODC, después a un DLP y finalmente al dispositivo asignado. Los datos de entrada desde los periféricos siguen la ruta inversa.

Subsistema de Comunicación de Datos

El subsistema de comunicación de datos de equipo A9 presenta un diseño modular. Este diseño permite que las funciones de comunicación sean distribuidas entre una serie de procesadores que incluyen el Network Support Processor (NSP), Line Support Processor (LSP) y Quad Line Adapters (QLA).

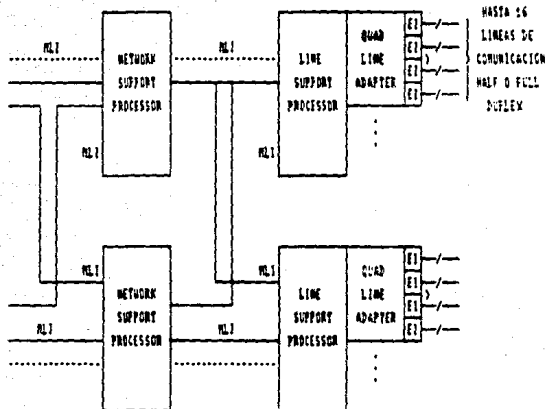
El NSP forma el control del subsistema de comunicaciones. La interacción entre el NSP y el sistema central es realizada a nivel de mensajes, eliminando interrupciones al sistema central.

Las funciones de un NSP incluyen el control de las líneas de comunicación y estaciones, la transferencia de la información y la reconfiguración de la red. El control del enlace incluye; seguirle la pista al estado lógico de las líneas de comunicación, establecer y romper conexiones y determinar qué protocolo de línea se ejecutará. La reconfiguración de la red permite a la instalación asignar, remover o cambiar estaciones o líneas de comunicación permitiendo reconfiguraciones dinámicas de la red.

El LSP es responsable de ejecutar las funciones del protocolo de línea (desconectar, contestar el teléfono, marcar un número telefónico...) dirigido por el NSP y realizar la traducción entre el conjunto de caracteres de la línea y el conjunto de caracteres del sistema central.

El LSP proporciona una conexión entre los "Line Adapters" y el NSP. La información es transferida entre el LSP y el NSP a nivel de mensajes. El concepto de transferencia a nivel de mensajes a diferencia de los niveles de "bits" o caracteres, reduce las interrupciones y permite que más información sea transferida en un determinado tiempo.

CONFIGURACION DE LA COMUNICACION DE DATOS



MLI = MESSAGE LEVEL INTERFACE

EI = ELECTRICAL INTERFACE

LAS LINEAS TRUQUEADAS MLI REPRESENTAN DATOS LOGICAMENTE ACTIVADAS.

LAS LINEAS CONTINUAS MLI REPRESENTAN DATOS FISICAS.

FIGURA 3.5

Un solo LSP es capaz de soportar hasta 16 líneas de comunicación "half" o "full duplex". Cada línea de comunicación puede ser configurada para proveer a una red de terminales, con una amplia variedad de tipos de terminales incluidas en la red.

Hay gran flexibilidad en las configuraciones en el sistema de comunicación. Un NSP es configurado en el sistema de la misma manera como otro DLP y, como tal está sujeto a las mismas reglas de configuración.

El componente final del "hardware" requerido para el subsistema de comunicación es el Quad Line Adapter (QLA). Este componente proporciona la interfase entre la línea de comunicación y el LSP. El QLA mantiene el control físico de cada línea de comunicación y acumula los caracteres, los cuales son transmitidos al LSP. Cada QLA está configurado con 4 KB de memoria local, la cual es usada para almacenar tablas de traducción, "buffers" para mensajes, parámetros de la línea y el código requerido para control de la línea de comunicación.

El NSP es programado usando una versión avanzada del Burroughs Network Definition Language (NDL II) el cual permite una rápida y eficiente definición e implementación de la red. NDL es un lenguaje descriptivo que simplifica la instalación aun de complicadas redes de comunicaciones.

Subsistema de Mantenimiento

El subsistema de mantenimiento provee facilidades para mantenimiento y diagnóstico, y un sistema para facilidades de inicialización. Consiste de un System Control Processor (SCP) y Host Maintenance Hardware .

El SCP es una unidad de procesador y display externa. Este procesador es capaz de inicializar el sistema y ejecutar programas de mantenimiento y diagnóstico en el sistema.

El Host Maintenance Hardware provee de detección de errores, registro, y funciones de control así como también una interfase entre el SCP y el sistema principal.

3.3.2 Software del Equipo A9

En esta sección se presentan los aspectos generales del "software" del equipo A9.

El Sistema Operativo en el sistema A9 se llama MCP (Master Control Program), es el administrador del sistema, él se encarga de asignar la memoria; iniciar, terminar o suspender tareas; asignar prioridades, etc.. Está escrito en un lenguaje llamado "NEWP", el cual facilita el mantenimiento y la integridad del Sistema Operativo.

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

Principales Secciones del MCP

Manejo de Interrupciones

Los módulos de manejo de interrupciones del MCP son usados para la interfase con los módulos de manejo de recursos del MCP. Hay tres tipos o grupos de interrupciones.

- **Causadas por el procesador de datos:**
 - "Interval timer", puede causar el MCP al conmutar a diferentes tareas.
 - "Presence bit", causa una solicitud al módulo administrador de la memoria.
 - Operando inválido, usualmente es debido a errores en el código objeto.
 - Índice inválido, es debido a un error de indexación en el programa del usuario.
 - "Stack Overflow", aparece cuando el programa del usuario necesita más área de trabajo.

- **Causadas por Entrada/Salida**
 - Cambio de Estado, cuando una unidad periférica cambia su estado.
 - Terminación de Entrada/Salida, una solicitud previa de Entrada/Salida es completada.

- **Causada por el Software**
 - Una tarea puede interrumpir a otra tarea del mismo "job".

Manejo de Memoria

Hay tres métodos para el manejo de memoria en estos equipos. Cada uno de los tres son diferentes implementaciones del concepto de memoria virtual.

Todas las tareas corriendo en el sistema están alojadas, al menos con un bloque en el "disk pack". Si se requieren más, el MCP asignará tantos como sean necesitados. Este espacio en "disk pack" es usado para almacenamiento temporal de datos que no son usados concurrentemente por la tarea. El tamaño de la región de "overlay" es definido al momento de inicializar el sistema. Los tres métodos estándar de manejo de memoria son:

"On Demand"

Cuando se usa este método, la memoria se asignará como lo requieran las tareas. Esto sucede porque toda la memoria que necesita una tarea no puede ser asignada cuando la tarea comienza. Elementos tales como archivos, no se les asigna espacio en memoria hasta que el archivo es abierto. Cuando no hay suficiente espacio en memoria, el MCP desasignará y volverá a usar áreas que no han sido usadas en forma concurrente.

"Working Set"

Este método es invocado poniendo ciertos elementos en el formato de un mensaje, el cual está dividido en cuatro partes en donde se le indica al sistema la cantidad de memoria que debe usarse para "overlays", así como el porcentaje de memoria que debe estar siempre disponible. Cuando la cantidad de memoria baja de ese porcentaje, el MCP suspenderá las tareas que tengan la prioridad más baja. Esto causará que muchas de las áreas de memoria sean liberadas y vuelvan a los recursos del sistema. El MCP no olvidará esas tareas; tan pronto como haya suficiente memoria las reactivará.

"Swapper"

Permite al usuario seleccionar una porción de memoria que es llamada "swap space" con el propósito de repartir el tiempo. Una tarea se encuentra en "swap space" cuando existe una actividad ejecutándose de esa tarea, de lo contrario la tarea residirá en un área llamada "swappedisk" en disco. Una tarea se encontrará en "swappedisk" por las siguientes razones:

1. La tarea está esperando entradas de datos
2. Los "buffers" de salida están llenos
3. Su tiempo ha terminado
4. La tarea ha sido suspendida por el sistema

MCP I/O

Son procedimientos que inicia el MCP para realizar funciones de I/O. Generalmente estos requerimientos de I/O son llamados por funciones del MCP como : Memory Management, MCP Library Maintenance, Swapper, Data Communications.

Desde que el MCP realiza este tipo de operaciones, es importante que el código de las funciones sea reentrante para así ser compartido por todas las tareas.

Control de Procesos

Estos procedimientos son usados para controlar el comienzo y la terminación de todos los trabajos y tareas en el sistema. También determina cuáles tareas correrán y cuando. Además estos procedimientos checan los cambios en los estados de los dispositivos periféricos así como también registra los nuevos archivos, nuevos "jobs", y tareas.

Control de periféricos

Son los procedimientos responsables del control de todos los dispositivos periféricos. Existen procedimientos para asignar dispositivos de salida a los programas, leer etiquetas desde todas las unidades que están listas y mantener una tabla de todas las unidades disponibles. Además de llevar un registro de los errores en las unidades.

Manejo de Disco

Las responsabilidades en el manejo del disco que tienen estos procedimientos son:

1. Inicialización del disco

Estos procedimientos verifican la integridad de los archivos en disco, establecen las tablas adecuadas en el disco y reconstruyen los directorios.

2. Alojamiento en disco

Asigna espacio adecuado a nuevos archivos de salida y crean entradas en el directorio. Estos procedimientos permiten al usuario controlar a cuáles tareas les será permitido leer y escribir en archivos de datos en el disco.

Interfase con el Operador

Estos procedimientos son responsables de aceptar y transmitir mensajes al operador.

Diagnósticos

Proporcionan la habilidad de obtener vaciados de la memoria, ya sea cuando lo requiere el operador o cuando ocurre una falla en el programa. Es útil para la depuración de sistemas.

Entrada/Salida del usuario

Son aquellas rutinas que manejan diversos aspectos del manejo de los datos definidos por el usuario. Incluyen abrir y cerrar archivos, checar y modificar los atributos de los archivos, etc.

Comunicaciones

El módulo controlador de las comunicaciones del MCP es un mecanismo de interfase entre los programas del usuario y el subsistema de comunicaciones.

Manejo de Datos

Este módulo es la interfase entre los programas del usuario y las bases de datos del usuario.

Workflow

Es un lenguaje por medio del cual el usuario puede indicar y describir al sistema A9 cuales "jobs" deberá ejecutar y en que secuencia. El compilador de WFL produce código para controlar las tareas.

Sort

Se encarga de ordenar los archivos de la manera en la cual se le indique. Mediante estos procedimientos es posible mezclar archivos.

Procedimientos Independientes

Estos procedimientos correrán en forma asincrónica con otras tareas en el sistema, algunos de ellos deben estar ejecutándose siempre.

Utilerías

Las utilerías son procedimientos que tienen una función específica en el sistema. Algunas de ellas las podemos considerar de uso exclusivo del sistema, otras de uso general. Entre las más importantes se pueden mencionar:

SYSTEM/BACKUP

El funcionamiento de algunos dispositivos como impresoras es lento, respecto a la velocidad con que el procesador lleva a cabo sus operaciones. Es por ello la importancia de esta utilería. Cuando se utiliza la técnica de BACKUP, un programa puede solicitar una impresora, en este caso, el Sistema Operativo le asigna dispositivos que tienen una mayor velocidad (disco, cinta o "pack"). Estos simulan a los anteriores y así el programa escribe en ellos lógicamente. Cuando el dispositivo lento ha sido liberado por otro proceso, será posible enviar a ellos la información que se quedó grabada en los periféricos veloces. De esta manera se logra optimizar el tiempo de máquina.

SYSTEM/COMPARE

Sirve para comparar uno o más pares de archivos, la comparación se lleva a cabo "bit" por "bit" en cada registro.

SYSTEM/DCSTATUS

Se utiliza para desplegar información del estado del subsistema de comunicación de datos.

SYSTEM/DUMPANALIZER

Este es un programa que analiza el estado de la memoria.

SYSTEM/FILEDATA

Se utiliza para obtener directorios y mapas del disco o "pack".

SYSTEM/GUARDFILE

Se utiliza para definir la seguridad de los archivos. Por medio de él se definen los usuarios y que programas tienen acceso a que archivos, también es posible definir el tipo de acceso.

SYSTEM/LOGANALIZER

Utiliza la bitácora del sistema para obtener reportes de mantenimiento.

SYSTEM/LOGGER

Con él es posible obtener reportes que permiten contabilizar los recursos utilizados en un periodo deseado.

SYSTEM/MAKEUSER

Ya que el acceso al sistema se hace por medio de claves, estas deben ser dadas de alta, baja o modificadas, esto se realiza mediante esta utilidad que sirve para acceder al User Data File (UDF) que es donde está guardada la información de cada una de las claves del sistema.

CANDE (Command and Edit)

CANDE es 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.

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

- Edición y corrección de archivos.
- Ejecución de programas
- Lanzado de trabajos
- Preguntar el estado de cierta línea
- Preguntar los recursos de máquina que un programa ha consumido hasta un momento determinado
- 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

Compiladores

Los compiladores con que cuenta el sistema A9 son:

ALGOL (DMALGOL, DCALGOL)

COBOL (DMCOBOL)

FORTRAN

PL1/1

BASIC

NEWP

NDL

PASCAL

RPG

Software Ambiental

Manejo de Bases de Datos

DMS II

DMS II INQUIRY

DATA DICTIONARY

Redes de Terminales

NDL

NDL II

Sistema de Control de Mensajes

GEMCOS

Generación de Reportes

**REPORTER II
AUDIT REPORTER
ON LINE REPORTER**

Procesamiento de Transacciones

TPS

Control y Edición de Programas

**CANDE
EDITOR
WFL**

Comunicaciones Inter-Sistemas

**RJE
SYCOM**

Redes de Computadoras

BNA

Generación Sistemas de Aplicación

(LINC) LOGIC AND INFORMATION NETWORK COMPILER

Consultas y Reportes

EXTENDED RETRIEVAL AND GRAPHICS OUTPUTS (ERGO)

CAPITULO 4

DISEÑO DEL SISTEMA

4.1 Metodología del Diseño del Sistema

La fase de diseño en el desarrollo de sistemas implica concebir, planear y especificar las características para un nuevo sistema o subsistema propuesto, de manera que satisfaga los requisitos determinados durante el análisis. En el diseño de "software" existen dos tipos de actividades, las cuales se agrupan en dos partes a las que llamaremos respectivamente diseño externo y diseño interno¹.

El diseño externo contempla la especificación de características tales como definición de entrada y salida de datos, definición de requerimientos funcionales, así como la definición estructural del sistema.

El diseño interno, incluye la concepción, planeación y especificación de la estructura interna y de los detalles de proceso del sistema. Las metas del diseño interno son especificar los detalles de procesamiento, los detalles de los algoritmos y las estructuras de datos. Además del refinamiento del enfoque conceptual del sistema, identificando funciones internas del proceso, descomposición de funciones de alto nivel en subfunciones, así como la definición de las estructuras de datos específicas que instrumentan el almacenamiento de los mismos y las interacciones entre datos y funciones.

Conceptos Fundamentales del Diseño

Los conceptos fundamentales en el diseño de la programación incluyen la abstracción, estructura, almacenamiento de información, modularidad, concurrencia, verificación y los aspectos estéticos en el diseño.

1 Fairley, Richard, ingeniería de software, p. 144

Abstracción

Permite organizar y dirigir los procesos de pensamiento, y posponer las consideraciones estructurales y algorítmicas hasta que las características funcionales, los datos y su almacenamiento hayan quedado definidos; las consideraciones generales son entonces estudiadas antes de ver los aspectos de la instrumentación. El enfoque anterior reduce la complejidad a la que nos enfrentamos en cualquier momento durante el diseño.

Estructura

El uso de una estructuración permite que un sistema sea definido en términos de unidades más pequeñas y manejables, con una clara definición de las relaciones entre las diferentes partes del sistema.

Modularidad

Los sistemas modulares consisten en unidades claramente definidas con interfaces entre los diversos módulos. La modularidad mejora la claridad del diseño que a su vez facilita la instrumentación, la depuración, las pruebas, la documentación y el mantenimiento de un sistema.

Concurrencia

Los sistemas pueden ser clasificados como secuenciales o concurrentes. En un sistema secuencial sólo una porción del sistema se encuentra activo en un momento dado; los sistemas concurrentes tienen procesos independientes que pueden ser activados en forma simultánea, si existen procesadores múltiples. En un procesador sencillo, los procesos concurrentes pueden quedar entrelazados en el momento de ejecución, esto permite la instrumentación de sistemas multiprogramados en tiempo real o tiempo compartido.

Verificación

La verificación es un concepto fundamental en el diseño de la programación, ya que el diseño es el puente entre los requerimientos del usuario y la instrumentación que satisface esos requerimientos.

Estética

Las consideraciones estéticas son fundamentales para el diseño; la simplicidad, la elegancia y claridad distinguen a un producto de alta calidad.

Instrumentos para el Diseño

Así como se cuenta con instrumentos que facilitan el análisis de un sistema, así también se han desarrollado instrumentos que ayudan en el diseño de un sistema. A continuación se describen algunos de estos instrumentos.

Diagramas de Flujo

Un diagrama de flujo consiste en una serie de símbolos y conexiones entre ellos. Un diagrama puede representar una gran variedad de actividades del proceso de información de un computador hasta los pasos detallados de un programa.

Tablas de Decisión

Las tablas de decisión son otra técnica gráfica que se emplea para facilitar la comunicación entre el usuario y los analistas. Las tablas de decisión o los algoritmos manuales se utilizan para lograr el beneficio de una decisión automática, de modo que se pueda generar fácilmente un programa basándose en una tabla de decisión. Una tabla de decisión expresa una serie de condiciones, cuando se cumplen las condiciones, se ejecuta una orden asociada regularmente con las citadas condiciones.

HIPO (Hierarchy and Input-Process-Output)

Esta técnica fué desarrollada para presentar de modo gráfico lo que hace un sistema o programa. La técnica HIPO se concentra en las funciones más que en los detalles inmediatos. Una función es un proceso que acepta entradas de datos y produce salidas de información. HIPO consiste habitualmente de una tabla visual de componentes que describen el sistema, diagramas de visión general y diagramas de detalle con descripciones externas. HIPO es también una forma de diseño modular, en la cual se dividen las partes del sistema en módulos independientes. El objetivo es producir módulos que tengan potencia funcional, es decir, que contengan todas las etapas necesarias para realizar una función sencilla.

Pseudocódigo

El pseudocódigo no es tanto un enfoque de diseño, como un medio de comunicar especificaciones a los programadores. El propósito del pseudocódigo es reemplazar un diagrama de flujo por un texto, pero un texto que no dependa de las peculiaridades de un lenguaje de programación en particular.

Diagramas de Warnier-Orr

Como una alternativa a la técnica HIPO, existe la metodología de diseño Warnier-Orr. Un diagrama de Warnier-Orr es una carta jerárquica en la cual se describe la estructura de un programa, considerando además la secuencia en la cual los módulos son invocados, y la frecuencia con que estos son ejecutados. Un diagrama de Warnier-Orr puede ser usado para describir una estructura de datos, un conjunto de detalles de programación lógica o una estructura completa del programa.

4.2 Etapas a Desarrollar en el Diseño del Sistema

Tomando en cuenta los conceptos fundamentales y técnicas específicas que nos proporcionan los principios para el desarrollo del diseño de un sistema, a continuación se describen los puntos a seguir en el diseño del sistema.

- **Diseño externo del sistema**
 - Consideraciones para el diseño
 - Características funcionales
 - Estructura general del sistema
- **Diseño interno del sistema**
 - Descripción conceptual de los archivos del sistema
 - Descripción funcional de los módulos del sistema

4.2.1 Diseño Externo del Sistema

4.2.1.1 Consideraciones para el Diseño

La mayoría de los sistemas en el ambiente administrativo son de propósito particular, es decir, el procesamiento de los datos tiene como objetivo final la generación de resultados, cuya interpretación ayude en la toma de decisiones.

El presente sistema puede ser clasificado como de propósito general si analizamos los siguientes puntos:

- Las entradas al sistema pueden ser de diversos tipos como registros de nómina, registros bibliográficos, resultados de encuestas, etc..
- Las pantallas de captura de los registros son definidas de acuerdo al criterio de cada usuario, en virtud de la facilidad que le proporcione en la captura.
- El comportamiento de la captura es de acuerdo al tipo de entradas y validaciones definidas por el usuario.
- El procesamiento de la información de entrada genera archivos de datos que no están sujetos a ninguna interpretación, sino que son datos de entrada a diversas aplicaciones totalmente independientes.

Si bien el sistema tiene un objetivo bien definido, su uso es de propósito general, lo que permitirá que los sistemas puedan usarse como herramienta auxiliar en el mantenimiento de archivos.

Para el diseño de este sistema consideramos como aspectos principales la necesidad de procesos locales y la comunicación con el equipo "main frame", en este caso el equipo AS.

Los procesos locales implican el que al microcomputador se le asignen todas las tareas referentes al mantenimiento de los archivos. En cuanto al proceso de comunicación, es indispensable establecer un enlace entre los equipos para la transferencia de archivos, hacia el equipo PC para su mantenimiento o bien al equipo AS para los procesos de aplicación que los requieran.

La figura siguiente muestra el planteamiento básico del cual partimos.

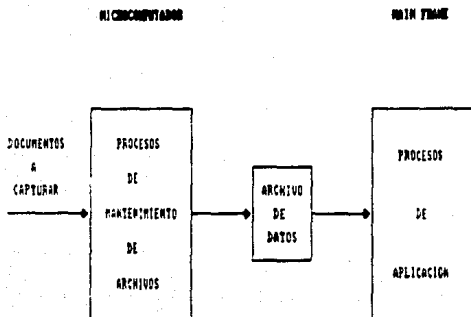


FIGURA 4.1

4.2.1.2 Características Funcionales del Sistema

Primeramente consideraremos los procesos que se requieren en el Departamento de Captura.

Uno de los procesos que se realizan cuando se recibe un trabajo, es el de la generación de un formato, el cual contiene las características que determinan la validación de los datos. Una vez que se cuenta con este formato se procede a la captura de la información, ya que se ha capturado la información, se transfiere al equipo "main frame" para otros procesos.

En el caso de que se requiera modificar información que ya ha sido transferida, el usuario podrá solicitar la transferencia del archivo hacia el microcomputador y realizar las modificaciones que requiera.

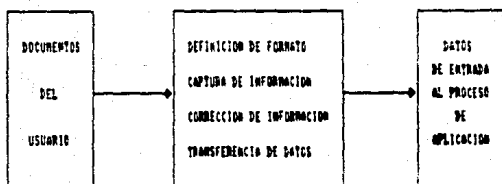


FIGURA 4.3

4.2.1.3 Estructura General del Sistema

Una vez que se han planteado los dos puntos anteriores, podemos presentar el siguiente esquema con las partes que consideramos fundamentales para integrar el sistema.

Menú

Este módulo permitirá el acceso a cada uno de los módulos que conforman el sistema.

Generación de Formato

En esta parte del sistema, se podrán generar los formatos que se requieran para capturar la información.

Modificación de Formato

Este módulo permitirá generar nuevos formatos a partir de los ya existentes.

DEFINICION GENERAL DEL SISTEMA

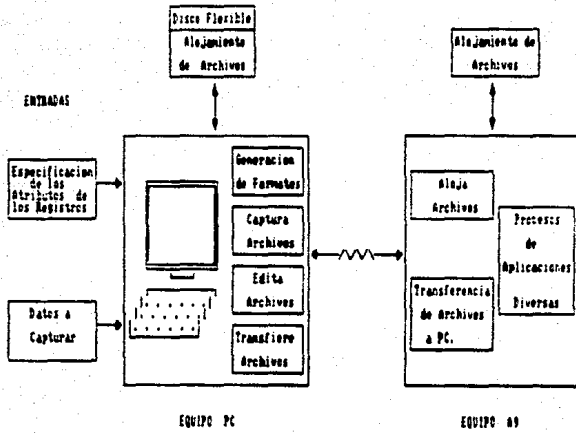


FIGURA 4.3

Captura

Es en este módulo, en donde se llevará a cabo el proceso de captura de información del usuario.

Edición

Aquí se podrán realizar las modificaciones o cambios, que se requieran a la información ya capturada previamente; ya sea, que la información haya sido transferida al "main frame" ,o bien, que aún se encuentre en el microcomputador.

Transferencia

Estos módulos permitirán establecer un enlace entre los equipos (microcomputador y "main frame"), para la transferencia de archivos.

4.2.2 Diseño Interno del Sistema

4.2.2.1 Descripción Conceptual de los Archivos del Sistema

A continuación se describirán los archivos que utilizará el sistema, así como la información que contendrán.

Archivo de Formato

Este archivo tiene como finalidad el conservar el formato con el que fué capturado un registro, para posteriores capturas del mismo tipo de trabajo o para uso del formato en el módulo de edición. Este tipo de archivo tendrá la extensión .fmt y contendrán la información que a continuación se describe.

En el primer registro de información de este archivo se encuentran los datos:

- Tamaño del registro
- Número de Campos
- Número de repeticiones

Tamaño del Registro	Se refiere al tamaño en caracteres que va a ocupar un registro de datos.
Número de Campos	Es el número de Campos del cual está formado un registro.
Número de Repeticiones	En la generación de un archivo de datos pueden usarse varios formatos. El número de repeticiones es el número de registros que se van a capturar con un formato determinado, antes de cambiar a otro formato para capturar el siguiente registro.

En la siguiente parte del archivo se tendrá el nombre y atributos de cada uno de los campos que forman el registro, los cuales se listan a continuación.

1. Renglón
2. Columna del nombre del campo
3. Longitud
4. Renglón
5. Columna del contenido del campo
6. Longitud
7. Columna de Inicio
8. Columna de Fin
9. Tipo de Dato
10. Constante
11. Obligatorio
12. Referencia (Izquierda, Derecha)
13. Caracter de llenado
14. Cota Inferior
15. Cota Superior

Renglón, Columna, Longitud (del nombre del campo)

Estos datos refieren a la localización del nombre con cual se identificará el campo e indica en qué lugar de la pantalla aparecerá en la captura.

Renglón, Columna, Longitud (del contenido del campo)

Estos campos se refieren al contenido del campo, los cuales son necesarios para indicar en el formato de captura el lugar que ocupará la información.

Columna de Inicio, Columna de Fin

Se refieren al lugar que ocupará el campo dentro del registro de salida.

El usuario podrá definir el orden que lleven los campos dentro del registro de salida, independientemente de como sean capturados, por lo que es necesario proporcionar los datos anteriores.

Tipo de Dato

Este atributo definirá el criterio de validación que se aplicará al campo captado. Los criterios de validación para este campo serán los siguientes:

1. Alfabético
2. Alfanumérico
3. Numérico
4. Alfabético2
5. Alfanumérico2
6. Numérico UNIVAC
7. Alfanumérico UNIVAC
8. R.F.C.
9. Caracteres Especiales
10. Cadenas de Caracteres

Alfabético

Este tipo de validación incluye exclusivamente los caracteres del alfabeto (A..Z), tanto mayúsculas como minúsculas.

Alfanumérico

Incluye los mismos caracteres del tipo anterior más los caracteres 0..9.

Numérico

Valida exclusivamente los números 0..9.

Alfabético2

Valida los caracteres del tipo alfabético e incluye caracteres especiales y de puntuación.

Alfanumérico2

Valida los caracteres del tipo alfanumérico e incluye caracteres especiales y de puntuación.

Numérico UNIVAC

Esta es una validación especial de los caracteres numéricos (0..9) que en este caso corresponden a las teclas ('I', 'U', 'l', 'O', 'J', 'K', 'L', 'M', 'N', 'P') respectivamente. Además incluye caracteres especiales y de puntuación.

Alfanumérico UNIVAC

Este tipo valida los caracteres del tipo anterior y además incluye los caracteres alfabéticos.

R.F.C.

Este tipo valida un campo en el que la información es el Registro Federal de Causante, el cual consiste de cuatro letras y seis números correspondientes a una fecha.

Caracteres Especiales

Este tipo es para validar un campo en el que solo se permitan ciertos caracteres definidos cuando se genera el formato.

Cadenas de Caracteres

En este se validará la información del campo, de acuerdo a cadenas de caracteres definidas durante la generación del formato.

Constante

La afectación de este atributo permite definir si el campo va a ser constante durante la captura o no. Si el campo es definido como constante el sistema pedirá sólo una vez el contenido del campo y en los registros siguientes el campo se repetirá automáticamente.

En la captura es posible manejar varios formatos, y declarar campos constante de otro formato, por lo que si un campo es constante, deberá indicarse además de qué formato es constante.

Obligatorio

Permitirá definir si un campo debe ser llenado o no. Esta característica ayuda a no saltar información que debe contener un registro. Por omisión de esta opción, todos los campos serán no obligatorios

Referencia (Izquierda, Derecha)

Con este atributo se indicará hacia que lado va a ser justificada la información del campo tanto en el formato de captura como en el registro de salida. Por omisión los campos serán justificados hacia la izquierda.

Caracter de llenado

Se refiere al caracter con que se va a completar el espacio faltante de algún campo. Puede ser blanco, cero, o cualquier otro caracter que el usuario defina. Por omisión será blanco.

Cota Inferior, Cota Superior

Se podrá validar el rango de una cantidad numérica durante la captura definiendo un límite inferior y un límite superior. Estos campos son opcionales.

Referencia a Registro de Validación1, Referencia a Registro de Validación2.

Estos campos los proporcionará el sistema en el caso de que el campo sea del tipo definición de caracteres o definición de claves.

La tercera parte del archivo de formato contendrá las claves y caracteres válidos, según se haya definido para algún campo en la creación del formato.

Archivo de Datos

Este archivo contendrá solamente los registros capturados, a través de un formato. Este archivo será de tipo texto y tendrá la extensión .dat.

Archivo de Control

Cada archivo de datos tendrá asociado un solo archivo de control, el cual tendrá la extensión .ctl y contendrá información concierne a la generación del archivo de datos, dicha información permitirá al departamento de captura llevar un control de los trabajos que se realizan, así como del personal que lo realiza. Contendrá también datos indispensables para poder crear el archivo de datos en el equipo central. La información que contendrá se menciona a continuación:

1. Longitud del registro en caracteres
2. Clave del usuario en el equipo A9
3. Nombre del archivo en el equipo A9
4. Nombre del formato con el cual fué generado este archivo
5. Número de registros en el archivo
6. Clave de la capturista que realizó el trabajo

7. Clave de la dependencia a la que se le hizo el trabajo
8. Bloqueo del archivo
9. Unidades
10. Fecha de la última actualización

4.2.2.2 Descripción Funcional de los Módulos del Sistema

Sistema de Captura y Edición de Archivos

En la figura 4.4 se muestran los módulos con los que contará el sistema, así como su interrelación con los archivos utilizados.

A continuación se describen cada uno de los módulos del sistema:

Módulo Principal

Este módulo permitirá el acceso al sistema así como a cualquiera de sus módulos.

Módulo de Generación de Formato.

Antes de iniciar el proceso de captura de un trabajo es necesario crear un formato. Este contendrá la información de como se distribuyen los campos del registro en la pantalla de captura, además de que se definirán los siguientes atributos.

- Posición en el registro de salida
- Columnas de inicio y fin del dato en el registro de salida
- Tipo de dato
- Si el campo es constante
- Si el campo es obligatorio
- Hacia donde se justifica la información del campo
- Especificación del caracter de llenado.

ESTRUCTURA FUNCIONAL DEL SISTEMA

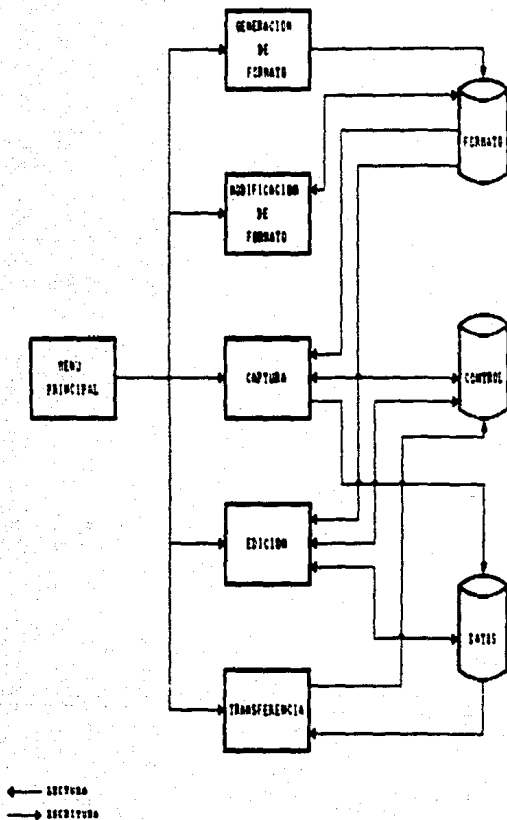


FIGURA 4.4

La definición de estas características se conservará en un archivo como se ya se dijo anteriormente para posteriores usos en trabajos que requieran las mismas características de formato.

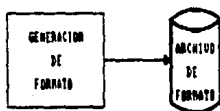


FIGURA 4.5

Módulo de Captura

En este módulo se generará un archivo de datos, donde cada uno de los registros tendrá las características definidas en el formato o formatos utilizados para su captura, para ello requerirá de la presencia de los archivos de control y formato.

Este módulo proporcionará las funciones comunes en la edición de caracteres (borrar, insertar, sobrescribir); además de facilidades para la agilización de la captura, las cuales se describen a continuación.

Borrado de Campo

Este comando permitirá limpiar un campo con una sola tecla, es decir, lo llenará con el carácter de llenado definido para ese campo.

Borrado de Registro

Permitirá limpiar el registro presente, es decir, se llenará cada campo con su respectivo carácter de llenado.

Duplicación de Campo

Permitirá que en el campo actual se duplique la información de ese mismo campo pero del registro anterior.

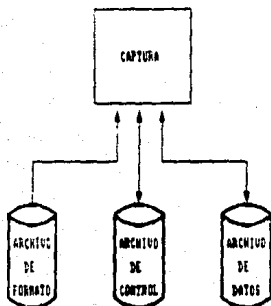


FIGURA 4.6

Duplicación de Registro

Este comando servirá para duplicar la información del último registro capturado en el registro presente.

Duplicación de Campo de un Formato Específico

Pudiendo manejarse varios formatos de captura en un solo trabajo, será posible duplicar un campo de otro formato especificando el número de formato del que se quiere la duplicación.

Duplicación de Registro de un Formato Específico

Duplicará la información del último registro capturado con el número de formato que se indique. Los comandos de duplicación tienen como objeto el ahorro de tiempo en el proceso de captura.

Selección del Formato Requerido

Si bien cada formato tiene un número de repeticiones definido, será posible cambiar de formato en el momento que se desee.

Característica de Constante

Un campo puede ser constante, de esta forma solo se captura una vez y las siguientes se repite automáticamente. Cuando en un trabajo, la información de un campo es la misma para un grupo consecutivo de registros, podrá definirse un campo normal como constante y cuando ya no se requiera esta característica, el campo podrá volver a ser normal.

Módulo de Edición

Este módulo se podrá acceder desde el módulo principal o a través del módulo de captura. Proporcionará las funciones comunes de edición en caracteres (borrar, insertar, sobrescribir), y la facilidad de realizar estas mismas operaciones a nivel registro, así como el acceso a los registros a través de una llave específica.

Durante el proceso de edición, los registros se mostrarán en la pantalla con el mismo formato con que fueron capturados, y al mismo tiempo se validará toda la información que sea modificada en este módulo (figura 4.7), de acuerdo a lo definido en el formato de captura.

Módulo de Modificación de Formato.

Muchas veces, teniendo ya un formato generado, se requiere generar otro muy similar, o debido a cambios en el formato a capturar se requiere realizar cambios en un formato ya existente. Tomando en cuenta este requerimiento de las personas que capturan la información, se hace necesaria la existencia de este módulo (figura 4.8).

Este módulo permitirá obtener nuevos formatos a partir de los ya existentes o la modificación de alguno de ellos.

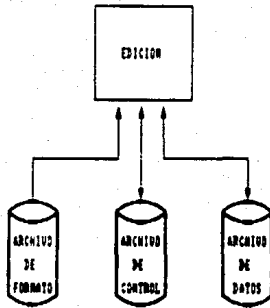


FIGURA 4.7



FIGURA 4.8

Módulo de Transferencia.

En este módulo será posible enviar archivos de datos o parte de ellos al equipo A9, para su procesamiento. También se podrá realizar el proceso inverso, es decir, enviar total o parcialmente archivos al microcomputador para su edición.

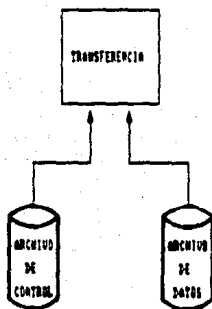


FIGURA 4.9

CAPITULO 5

INSTRUMENTACION DEL SISTEMA

La fase de instrumentación del sistema se refiere a la traducción de las especificaciones del diseño a código fuente. El objetivo principal de la instrumentación es el escribir código fuente y la documentación interna de modo que la concordancia del código con sus especificaciones sea fácil de verificar, y que se faciliten la depuración, pruebas y modificaciones.

La claridad del código fuente se mejora mediante técnicas de codificación estructurada, buenos comentarios internos, y por las características que proporcionan los lenguajes de programación.

El lenguaje de instrumentación utilizado facilita la documentación interna. Una rutina en C es más autodescriptiva que una rutina escrita en ensamblador, y requerirá menor información en el prólogo de documentación ya que las palabras reservadas en los lenguajes de alto nivel, proporcionan una idea más clara del proceso que realiza cierto código.

La instrumentación del Sistema de Captura y Edición comprenderá si no el código fuente estrictamente, si las estructuras con las que se instrumentó; considerando de mayor utilidad la presentación y explicación de estas; pudiendo además, consultar el código fuente en la documentación del sistema. También se expondrán los formatos utilizados en la codificación interna, así como la documentación interna.

5.1 Formato y Documentación Interna

En esta sección se describirá brevemente la forma en que se realizó la programación, así como la documentación interna utilizada. Básicamente se tomaron en consideración los siguientes elementos:

-Prólogo de documentación de programa

Cada módulo puede tener varios archivos que lo conforman. El prólogo de documentación aparece al inicio de cada archivo, describiendo en forma breve su propósito.

-Prólogo de documentación de rutina

Este prólogo consiste en un texto al inicio de la rutina que contiene el propósito de la rutina, los parámetros de entrada y los parámetros de salida.

-Comentarios entre líneas

Estos comentarios se usan dentro del código de las rutinas cuando es necesaria una explicación adicional. Esto ayuda tanto a la persona que programa como a quien consulta el código fuente.

-Formato de código del programa

Cada persona que programa tiene un estilo propio, es decir, su propia manera de poner los comentarios y de dar sangría a las líneas del programa, sin embargo, cuando la programación es realizada por varias personas este estilo debe ser uniformizado para facilitar la depuración.

Un ejemplo de documentación usada en las rutinas es el siguiente:

```
/*****  
  
Rutina para verificar la existencia de un archivo. Los parámetros de entrada son: coordena-  
das de pantalla, variable para nombre del archivo, tamaño máximo del nombre del archivo,  
tipo de validación de caracteres, extensión del archivo, atributo de video.  
La salida es: 27 (ESC aborta), 0 (el archivo existe), -1 (el archivo no existe).  
  
*****/  
  
verif_arch(x,y,arch,tam,tipo,ext,atr)  
char *arch, *ext, tipo;  
int x, y, tam, atr;  
{  
    int result = 0;  
    do  
        result = lee_var(x,y,arch,tam,tipo,atr,0); /*captura nombre del archivo */  
    while( result == 0 );  
    if( result == 27 )  
        return(result);  
    else  
        strcpy(arch,ext);  
        result = access(arch); /* valida existencia */  
        return(result);  
}
```

5.2 Estructuras Utilizadas en el Sistema

Módulo de Generación de Formato

El programa de Generación de Formato guarda los datos referentes a los atributos de los campos de un registro de captura, en una tabla (figura 5.1), una vez que el formato ha sido terminado, esta tabla se graba a un archivo en disco.

Módulo de Captura

A continuación se describen las estructuras usadas en los programas de Captura.

Tabla de Formatos de Captura

Descrita anteriormente (figura 5.1).

Tabla de Archivos de Formato de Captura

Esta tabla contiene los nombres de los archivos en los que se encuentran los formatos a utilizar (figura 5.2); además de una referencia a la posición de cada formato en la tabla t[], donde se encuentran definidas las características de cada uno de sus campos, lo que permite pasar de uno a otro formato con sólo hacer referencia al número de formato que se desea.

Tabla de "Buffers" de Captura

Existe un "buffer" de captura del registro para cada formato que se utilice (figura 5.3). Cada uno de los "buffers" es inicializado con el caracter de llenado del registro (atributo definido en la Generación de Formato). Si existen campos constantes para este registro, se le asignan sus valores respectivos en este "buffer".

TABLA DE FORMATOS DE CAPTURA

(E 3)

							A ₆	A ₈										
FORMATO 14	CAMPO 1	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃	A ₁₄	A ₁₅	A ₁₆	A ₁₇
	CAMPO 2	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃	A ₁₄	A ₁₅	A ₁₆	A ₁₇
FORMATO 24	CAMPO 1	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃	A ₁₄	A ₁₅	A ₁₆	A ₁₇
	CAMPO 2	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃	A ₁₄	A ₁₅	A ₁₆	A ₁₇
FORMATO 34	CAMPO 1	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃	A ₁₄	A ₁₅	A ₁₆	A ₁₇
	CAMPO 2	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆	A ₇	A ₈	A ₉	A ₁₀	A ₁₁	A ₁₂	A ₁₃	A ₁₄	A ₁₅	A ₁₆	A ₁₇

- A₁ NOMBRE DEL CAMPO
- A₂ TIPO
- A₃ CONSTANTE
- A₄ OBLIGATORIO
- A₅ REFERENCIA
- A₆ CARACTER DE LLENADO
- A₇ RENDON DEL NOMBRE
- A₈ COLUMNA DEL NOMBRE
- A₉ LONGITUD DEL NOMBRE
- A₁₀ RENDON DEL DATO

- A₁₁ COLUMNA DEL DATO
- A₁₂ LONGITUD DEL DATO
- A₁₃ COLUMNA DE INICIO
- A₁₄ COLUMNA DE FIN
- A₁₅ POSICION
- A₁₆ COTA INFERIOR
- A₁₇ COTA SUPERIOR

FIGURA 5.1

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

TABLA DE ARCHIVOS DE FORMATO DE CAPTURAS
 anexo 1

FORMATO 1	NUMERO DE FORMATO	NOMBRE ARCHIVO FORMATO	REFERENCIA A POSICION DEL FORMATO EN LA TABLA T11
FORMATO 2	NUMERO DE FORMATO	NOMBRE ARCHIVO FORMATO	REFERENCIA A POSICION DEL FORMATO EN LA TABLA T11
.			
FORMATO N	NUMERO DE FORMATO	NOMBRE ARCHIVO FORMATO	REFERENCIA A POSICION DEL FORMATO EN LA TABLA T11

FIGURA 5.2

TABLA DE "MAYORES" DE CAPTURAS
 Anexo 1

	CAMPO 1	CAMPO 2	CAMPO 3	CAMPO N
FORMATO 1	CARACTER DE LLENADO	CARACTER DE LLENADO	CARACTER DE LLENADO		CARACTER DE LLENADO
FORMATO 2	CARACTER DE LLENADO	CARACTER DE LLENADO	CARACTER DE LLENADO		CARACTER DE LLENADO
.					
FORMATO N					

FIGURA 5.3

Tabla de Duplicación

A continuación se muestra la tabla `bf_dup[]`. Cada arreglo de esta matriz es un "buffer" que contiene el último registro capturado de cada formato. El objeto de esta tabla es permitir tanto la duplicación del último campo así como registro capturado del formato que se desee.

TABLA DE DUPLICACION
bf_dup [i] j

ULTIMO REGISTRO CAPTURADO DEL FORMATO 1
ULTIMO REGISTRO CAPTURADO DEL FORMATO 2
⋮
ULTIMO REGISTRO CAPTURADO DEL FORMATO n

FIGURA 5.4

Registro de Captura

La figura 5.5 muestra el "buffer" `registro[]`. Este es un arreglo de caracteres en el cual se encuentra la información que está siendo capturada actualmente.

REGISTRO DE CAPTURAS
registro[]

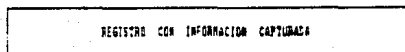


FIGURA 5.5

Buffer de Grabación

El vector registro[] de la figura 5.6 es usado para almacenar varios registros que se han capturado antes de grabarlos en el correspondiente archivo de datos.

BUFFER DE GRABACION
registro[]

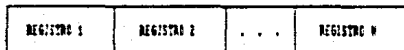


FIGURA 5.6

Registro de Campos Constantes

El arreglo t_cte[], mantiene el estado de constante o no constante para cada uno de los campos de los formatos que se están utilizando.

Este es un arreglo auxiliar puesto que esta característica no puede ser modificada en el formato original y se requiere para recuperar la condición original.

REGISTRO DE CAMPOS CONSTANTES

t_estado

CAMPO 1	CAMPO 2	CAMPO 3	CAMPO 4	. . .	CAMPO N
CONSTANTE	CONSTANTE	NO CONSTANTE	CONSTANTE		NO CONSTANTE

FIGURA 5.7

Registro de Control

A continuación se menciona la información del registro regctl:

- Número de registros
- Longitud del registro
- Bloqueo del archivo
- Unidades
- Nombre del archivo en la PC
- Nombres de los archivos de formato
- Nombre del archivo en AS
- Clave del usuario
- Clave de la dependencia
- Clave de la capturista
- Fecha de la última actualización

REGISTRO DE CONTROL

regctl 3

NO.	LOGITUC	BLOQUEAJE	UNIDADES	NOMBRE	NOMBRE	NOMBRE	CLAVE	CLAVE	CLAVE	FECHA
REGISTROS	REGISTRO	ARCHIVO		ARCHIVO FC	FORMATOS	ARCHIVO 69	USUARIO	CAPTURISTA	DEPENDENCIA	ACTUALIZACION

FIGURA 5.8

Esta información es solicitada si el archivo de datos se va a crear de lo contrario es solamente modificada por el programa de captura. Permite que se tenga control de la información que se captura en el Departamento de Captura, así como mantener datos necesarios para la transferencia de archivos al equipo "main frame".

En el módulo de Captura es en donde se llevan a cabo mayor número de acciones, y son empleadas la mayoría de las estructuras.

Las estructuras en este módulo interactúan con los procesos de la manera siguiente:

Primeramente, el sistema definirá si el archivo de captura es nuevo o ya existe. En el primer caso, se solicitan los datos de la estructura regctl, los cuales incluyen los nombres de los archivos de formato que se van a utilizar, estos datos serán grabados en un archivo al finalizar la captura.

En el segundo caso, existe ya un archivo con los datos de la estructura regctl y únicamente son leídos del archivo correspondiente.

A continuación se llenará la información de la tabla ar[] con los nombres de los archivos de formato tomados de la estructura regctl. Los archivos de formato son leídos y alojados en la tabla t[]. En ar[] se asigna la referencia a t[] de cada formato.

Los "buffers" de la tabla bfmt[] son iniciados con el caracter de llenado. La primera vez que se capture información con un formato seleccionado se capturan los campos constantes y se copian en su respectivo formato en bfmt[], esto permite limpiar el registro que se está capturando con la información de la tabla bfmt[].

Una vez que se inicia la captura, se selecciona el formato deseado y con ello se actualizan los índices a las tablas ar[], t[], bfmt[].

El registro en el que se va a capturar es inicializado con la información de `bfmt[]` correspondiente a ese formato. Los campos del registro se capturan de acuerdo a las validaciones definidas en la tabla `t[]`.

Cada registro capturado se agrega al "buffer" `regsto[]`, esto con el propósito de acumular cierto número de registros antes de grabarlos en el archivo de datos.

Cuando se termina de capturar un registro, la información es guardada en su "buffer" correspondiente en `bf_dup[NFMT][TREG]`, de tal manera que por cada formato se mantiene una copia del último registro capturado, con el fin de permitir duplicaciones del último registro capturado de un formato determinado.

Existe en el sistema la posibilidad de modificar el estado de los campos constantes, para ello se utiliza el arreglo `t_cte[]`, el cual mantiene el estado de cada campo durante el proceso de captura.

Módulo de Edición

En este módulo se requieren implementar las siguientes operaciones:

- Acceso a los registros a través de una o múltiples llaves
- Actualización de registros
- Borrado de registros
- Inserción de registros
- Además de conservar las validaciones correspondientes a los datos.

Para implementar dichas operaciones se contemplan dos opciones:

Desarrollar e implementar un sistema manejador de registros

Utilizar un manejador ya existente

El seleccionar la primera opción implica el invertir mayor tiempo y esfuerzo para su realización, ya que se requeriría el diseño y desarrollo de otro sistema, lo cual no se justifica ya que se dispone de herramientas con las facilidades requeridas, que además proporcionan interfaces con diferentes lenguajes, requiriéndose un mínimo de programación.

Por lo tanto, se decidió utilizar un manejador ya existente, en particular el manejador "Btrieve", el cual tendrá una interfase con el programa de captura, en la edición de los registros.

A continuación se describen algunas de las principales características de este paquete.

Btrieve es un Sistema Manejador de Registros para programas de aplicación. Btrieve proporciona interfases para muchos de los lenguajes más populares como son Turbo Pascal, Turbo C, Microsoft C, Basic, etc.

Debido a las técnicas y estructuras de Btrieve, se puede ignorar la estructura física de los archivos y concentrarse en la estructura lógica.

Funciones

Algunas de las funciones que hacen de Btrieve un poderoso Sistema Manejador de Registros son las siguientes:

- Acceso a registros por múltiples llaves
- Manejo de Bases de Datos relacionales
- Mantenimiento automático de todas las llaves
- Manejo de llaves duplicadas, modificables, segmentadas y nulas
- Manejo de archivos particionales
- Memoria "cache"
- Integridad de datos
- Tamaño no restringido de archivos
- Utilerías para mantenimiento y creación de archivos
- Acceso de datos en un ambiente multiusuario

Conceptos Lógicos de los Archivos

Registro

Unidad de transferencia entre un programa de aplicación y el Manejador de Registros (Btrieve) en una operación.

No existe restricción en cuanto al número de registros permitidos en un archivo Btrieve.

Un registro puede ser de longitud fija o puede consistir de una parte de longitud fija, seguida de una parte de longitud variable.

Todas las llaves que se definen deben de estar localizadas dentro de una parte de longitud fija del registro.

La longitud máxima de la parte fija de un registro depende del tamaño de la página que se defina, así como del número de llaves duplicadas en el archivo.

Btrieve permite hasta 4096 "bytes" como longitud máxima de la parte fija de un registro, y hasta 64 kb para la longitud variable.

Llaves

Cuando se crea un archivo, se pueden especificar cuatro diferentes atributos para cada una de las llaves del archivo. Los cuatro atributos son duplicada, modificable, segmentada, nula.

Estructuras Físicas del Archivo.

Un archivo Btrieve consiste de una página de encabezado, una página de índices y una página de datos. Una página es la unidad de almacenamiento, la cual Btrieve transfiere entre memoria y disco durante una operación de entrada/salida.

Cada página tiene un tamaño fijo que se especifica cuando se crea el archivo. El tamaño de la página es siempre múltiplo de 512 "bytes", hasta 4096 "bytes". El tamaño óptimo de la página depende del número de llaves segmentadas y de la longitud del registro de datos.

Página de Encabezado

Cada archivo tiene un registro, el cual se localiza siempre en la primera página, esta página contiene el tamaño del archivo y otras características.

Página de Datos

Cada página contendrá uno o más registros de datos. El número de registros depende de la longitud del registro definida en la creación del archivo.

Página de Índices

Esta página contiene las llaves para acceder los registros de datos. Cada llave tiene una dirección del registro (o dos cuando se especifican llaves duplicadas).

Facilidades de Btrieve utilizadas en el Programa de Edición.

- Creación de un archivo
- Abrir un archivo
- Cerrar un archivo
- Inserción de un registro
- Borrar un registro
- Obtención de un registro con una llave
- Obtención del siguiente registro
- Obtención del registro anterior
- Obtención de un registro con una llave mayor o igual a la proporcionada.
- Obtención del registro con la llave de menor valor
- Obtención del registro con la llave de mayor valor
- Actualización de registros
- Alojamiento y desalojamiento del manejador (Btrieve)

Hasta aquí se han descrito las facilidades y características del paquete Btrieve, enseguida se describirán las estructuras en el programa de Edición, así como el funcionamiento de estas (interactuando con el manejador Btrieve).

Estructuras del Programa de Edición.

regct: Esta estructura es utilizada para saber con que archivos de formato fué generado el archivo de datos y cual es el total de registros que se tienen. Solamente se actualizan algunos de sus campos por este programa.

t[]: Esta tabla es usada para editar el registro de acuerdo al formato que se quiera y por lo tanto a las validaciones definidas para los campos de ese formato.

ar[]: Se usa en Edición para la selección del formato con el que se quiere trabajar.

registro[]: Este es un vector en donde se aloja el registro a ser editado por el programa .

a.reg[]: Es un vector, parte de una estructura que contiene la información del registro, tal como se requiere para editarlo a través de Btrieve.

Una vez, conocidas las estructuras del programa de edición pasaremos a explicar la interacción de estas, en un proceso de edición.

Cuando se inicia un proceso de edición, se proporciona el nombre del archivo de datos que se desea editar, con este dato se procede a verificar la existencia de dicho archivo, para que sea posible la edición.

Si es viable el proceso de edición se procede a generar un archivo con extensión .btr ya que la edición se realiza a través del manejador Btrieve, el cual requiere el archivo de datos con algunas características adicionales, como son, que cada registro de datos tenga al inicio la longitud del registro seguida por un caracter "," o por un "blanco" (figura 5.9), además se deben de proporcionar las características de la llave o llaves de acceso al archivo.

REGISTRO DE DATOS

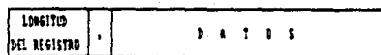


FIGURA 5.9

En el archivo Btrieve es en donde se llevarán a cabo todas las operaciones de edición.

Teniendo listo el archivo de datos a editar, se realiza el proceso de llenado de las tablas t[] y ar[], en las que se encontrarán las características de los formatos de captura así como la referencia a dicha tabla.

Con esto se procede a desplegar el primer registro de datos, en el primer formato definido en la tabla, si el registro no se capturó con dicho formato, se tiene la opción de seleccionar el formato correcto, ya que las modificaciones que se realicen a los datos, serán validadas de acuerdo al formato seleccionado.

Ahora bien, con respecto a la operación de inserción de un registro, el buffer en donde será capturada la información, será inicializado con los caracteres de llenado, definidos en la tabla de formatos.

Una vez que se termina un proceso de edición el archivo de datos deberá ser modificado, de tal forma que los registros de datos, solo contengan la información capturada.

CAPITULO 6

IMPLANTACION DEL SISTEMA

Las etapas finales del desarrollo de un sistema incluyen las pruebas y la instalación. Es necesario que el usuario esté muy involucrado para facilitar la operación del nuevo sistema.

Durante la etapa de pruebas se incluyen la verificación de la lógica básica de los programas y la verificación de que todo el sistema funciona adecuadamente. En todas las pruebas de programas no se puede ser exhaustivo, porque hay un gran número de combinaciones de caminos en un programa. Sin embargo, estas verificaciones son un intento de asegurarse de que los caminos que se emplearán más frecuentemente trabajen bien.

Es importante incluir a los usuarios en el desarrollo de los datos de prueba, porque entonces los datos serán menos imaginarios y más realistas.

Si se dispone de una codificación clara, se estará preparado para la supresión de errores y para los cambios que soliciten los usuarios. El hecho de que los usuarios estén muy implicados en el diseño reducirá el número de modificaciones, pero se realizarán solicitudes importantes de alteraciones cuando los usuarios adquieran experiencia en el sistema.

En las etapas de instalación, es necesario considerar el impacto del sistema en las personas, es decir, que cada persona puede tener una reacción diferente al sistema, dependiendo de diversas situaciones tanto de trabajo como personales.

Se pueden realizar diversas acciones para canalizar a los usuarios hacia el uso constructivo del sistema y su entrada en servicio con éxito. Para ello se deben definir los procedimientos y habrá que aplicarles algunas pruebas piloto que se consideran convenientes.

A continuación se describe el proceso de implantación del Sistema de Captura y Edición de Archivos.

ETAPAS DE LA IMPLEMENTACION DEL SISTEMA

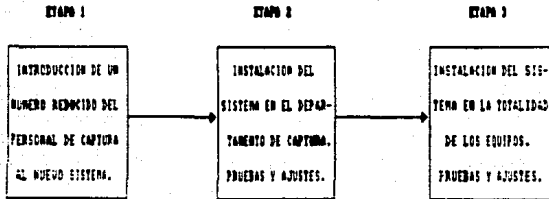


FIGURA 4.1

6.1 Primera Etapa: Introducción al Sistema

Como se ha expuesto anteriormente, el nuevo sistema no sólo implica un cambio en el "software" sino también en el "hardware"; el equipo microcomputador CROMEMCO con procesador Z-80 y dos unidades de disco flexible con capacidad de 360 Kb cada uno fué sustituido por el equipo microcomputador PC GAMAB8 con procesador 8088, unidad de disco duro de 30 Mb y unidad de disco flexible con capacidad de 360 Kb.

Previamente a la instalación del sistema en el Departamento de Captura, se seleccionó un grupo de personas (3 personas) que se consideró presentaban mayor disposición a conocer y aportar sugerencias referentes al nuevo sistema.

En esta etapa se les explicaron las principales diferencias entre el equipo que estaban usando y el equipo PC, además de algunas facilidades del Sistema Operativo, así como también se les presentaron los dos primeros módulos del sistema (Módulo de Generación de Formatos y Módulo de Captura) lo que nos permitió obtener las primeras impresiones acerca del funcionamiento del sistema.

6.2 Segunda Etapa: Instalación Parcial del Sistema

En esta etapa se presenta el sistema a todo el personal del Departamento de Captura (12 personas). Para ello se inicia con un curso introductorio del Sistema Operativo MS-DOS que les permita obtener las herramientas necesarias para la administración y control de su información (manejo de directorios, respaldos, etc.).

A partir de entonces, una parte del personal trabajará en el nuevo sistema (solamente 3 equipos), y la otra continuará con el sistema anterior. Cabe aclarar que el personal será rotado cada determinado tiempo según lo considere conveniente el jefe del departamento, de forma tal, que todo el personal conozca y maneje el nuevo sistema con el objeto de generar una retroalimentación que permita corregir, modificar y realizar los cambios convenientes.

6.3 Tercera Etapa: Instalación Total del Sistema

Hasta este momento el sistema se encuentra funcionando con los elementos básicos para la captura de información, pero instalado en un mayor número de equipos, por lo que la adición y/o modificación de módulos se llevará a cabo de la misma forma que en la instalación inicial (instalando y probando en un número reducido de equipos) de tal forma, que el trabajo en el equipo PC no sea entorpecido.

A partir de ahora, se considera que el sistema ha sido probado en forma satisfactoria y se le puede asignar cualquier trabajo del departamento, siendo necesaria la instalación del sistema en un mayor número de equipos.

De esta forma, la implantación del sistema se realiza como un proceso en paralelo entre la instalación del nuevo sistema y la eliminación de equipos CROMEMCO.

Aunque el sistema ha sido liberado y se encuentra funcionando, no está exento de cambios y adiciones requeridas por el surgimiento de nuevas necesidades o situaciones no previstas, esto recaerá en la fase de mantenimiento mientras dure la vida útil del sistema.

Las actividades del mantenimiento implican mejorar los sistemas, adaptarlos a nuevos ambientes, y corregir problemas. Estos procesos de mantenimiento tendrán en adelante como mayor apoyo la documentación del sistema, la cual consiste de un manual de operación para el usuario y la documentación detallada del "software".

CAPITULO 7

DOCUMENTACION

Documentación es un término que se usa para describir todas las instrucciones, programas y prácticamente todo lo que se escribe sobre un sistema de información. La documentación sirve para diversos propósitos; primero, durante el diseño de los sistemas, es el producto desarrollado por los analistas y los usuarios. Después de la instalación es la ayuda para los usuarios en la operación además de ser la base para hacer cambios en el sistema.

7.1 Documentación Técnica

En la documentación técnica se encontrará toda aquella información referente a la instrumentación del sistema. Tal como:

- Los archivos que maneja el sistema
- La descripción de los registros de cada archivo.
- Las estructuras principales que se manejan en los diferentes archivos que constituyen el sistema.
- Diccionario de datos, en donde se expliquen las variables más importantes y generales.
- La descripción de las rutinas de propósito general, sus entradas y salidas también constituyen elementos de documentación técnica.
- La documentación dentro del propio programa, además de ser de buen estilo, contribuye a la legibilidad y entendimiento de los mismos por personas que no realizaron la programación. Si se emplean la fijación de límites del código, los comentarios y los programas estructurados, los bloques de lógica dentro del programa son una forma de documentación.

-Los listados y programas fuente constituyen otra parte importante en la documentación del sistema.

La documentación técnica será de utilidad, principalmente a quien esté encargado de realizar las labores de mantenimiento del sistema.

7.2 Documentación de Operación

La persona o personas que van a operar el sistema necesitan información sobre los procedimientos normales de operación y sobre cómo responder a todos los errores.

Entre el contenido de esta documentación se encuentra un diagrama de flujo completo del sistema que presenta la entrada que requiere cada programa, los archivos necesarios, y la salida que debería producir.

Este manual debe explicar en forma clara y sencilla la manera de operar el sistema desde el momento en que se ejecuta, explicación módulo por módulo de su propósito, las acciones que realiza así como los comandos que se pueden ejecutar en cada uno.

En el caso del manual del sistema de captura, este contendrá una explicación de cada módulo, y específicamente cada pantalla, el propósito de cada opción y la manera de seleccionarla, así como la explicación de los datos que son solicitados, tales como nombres de archivos, datos específicos de cada trabajo como es el caso de la pantalla que solicita los datos de control, y el orden en que deben ejecutarse las acciones en el caso por ejemplo de la pantalla para generación de formato. El funcionamiento de la captura y la edición, la explicación de los datos que aparecen en la pantalla, así como los comandos que se pueden utilizar.

7.3 Documentación de Referencia

Este tipo de documentación se emplea para referencia del usuario después de que el sistema ya está funcionando. El usuario que tenga una duda o un problema debe consultar esta documentación. Esta documentación incluye los siguientes elementos:

- La descripción de la nomenclatura utilizada en esta documentación.
- Los requerimientos de "software" y "hardware" del sistema, es decir los programas ejecutables propios del sistema de captura y edición de archivos.
- Los nombres de los archivos que se requieren a la entrada de cada módulo, así como las salidas y modificaciones que el proceso realice.

- Explicación de las capacidades y los límites del sistema. Tales como:
 - El número de formatos diferentes que se pueden manejar en la captura
 - El número de campos que se pueden definir en un formato
 - La longitud de los nombres de los archivos, así como los caracteres que pueden constituirlos.
 - Los tipos de campos que se pueden definir.

- La explicación detallada de cada uno de los campos solicitados por el sistema.
- El propósito bien detallado de cada opción de cada uno de los módulos.
- Esta documentación también deberá contener una lista de los errores que puedan ocurrir, y se indicará además la manera de corregirlos.
- Es necesario que la documentación contenga un índice detallado para que se pueda utilizar fácilmente como referencia.

CAPITULO 8

DISEÑO DEL SISTEMA DE COMUNICACION

En la búsqueda para lograr mayor oportunidad en la información, mejorar la exactitud de los datos y finalmente optimizar el uso de las computadoras, se encontró la manera de lograrlo mediante el empleo de terminales por el usuario final, conectadas con el computador.

Como respuesta a la necesidad de poder comunicarse con la computadora desde lugares remotos, se han diseñado sistemas de comunicación digital, que permiten hacer más eficientes los canales de comunicación. Para ello se utilizan los diferentes medios de comunicación existentes, entre los que podemos contar líneas telefónicas, cable coaxial, redes de microondas, satélites, fibra óptica, rayo laser, etc..

Con la utilización de microcomputadoras en la red de teleproceso, se pretende que los usuarios realicen el máximo posible de trabajo en forma local, utilizando la amplia variedad de "software" que actualmente existe para este tipo de equipo, y transferir los archivos posteriormente para su utilización en la computadora central.

Antes de plantear el diseño correspondiente a este capítulo, se realizará un estudio de los aspectos generales para la comunicación, así como los elementos con que se cuenta para este desarrollo en ambos equipos.

8.1 Conceptos Generales de Comunicación

Para establecer un enlace de comunicación entre dos dispositivos es necesario un canal, el cual se puede definir como una línea a través de la cual fluye información. Todos los canales tienen limitaciones en el manejo de información, dependiendo de sus características físicas y eléctricas.

A continuación se mencionan los conceptos básicos involucrados en un enlace de comunicación:

Tipos de canales

Simplex

Transmisión de datos en una única dirección.

Half-Duplex

Transmisión de datos en dos direcciones , pero no simultáneamente.

Full-Duplex

Transmisión de datos en dos direcciones y simultáneamente.

Tipos de Transmisión

Transmisión Asíncrona

Transmisión secuencial de caracteres. Utilización de los "bits" de "start" (arranque) y "stop" (parada); los cuales se incluyen en cada caracter que sea transmitido. Por lo general el "bit" de "start" tiene un tiempo de duración igual al tiempo de una unidad de información. El "bit" de "stop" puede tener 1, 1 1/2 o 2 veces el tiempo de una unidad de información.

Transmisión Síncrona

La sincronización se realiza enviando la información en bloques, cada bloque es prefijado con un código único, el cual será reconocido por el receptor. Este tipo de transmisión tiene relojes de transmisión y recepción, los cuales tienen como función muestrear los "bits" de información. Cuando la terminal receptora ha detectado el caracter de sincronía se procede a la transmisión de datos caracter por caracter, sin utilizar señales adicionales, hasta que el receptor detecte un caracter especial de fin de transmisión.

Transmisión en Serie

Datos transmitidos a través de un solo canal de comunicación. Nos permite transmitir en forma síncrona o asíncrona.

Transmisión en Paralelo

Se utiliza un canal de comunicación para cada uno de los "bits" que se requieren, para transmitir un caracter. Este tipo de transmisión se utiliza generalmente en el enlace de la computadora con los dispositivos periféricos que están ubicados en forma local.

Modem

Dispositivo electrónico que se intercala entre la computadora y la línea de transmisión para transformar, adecuar y amplificar la salida digital de un equipo terminal en una señal analógica (modulación) capaz de ser transmitida a través de un canal¹.

Protocolo

El protocolo de datos se puede definir como las características de control de una red, las cuales son un conjunto de convenciones que tienen por propósito gobernar la transmisión de datos e información de control.

8.2 Características del Equipo PC para Comunicaciones

Interfase RS-232

Al conjunto de circuitos y cables que permiten la unión entre un equipo terminal de datos (DTE) y un equipo de comunicaciones de datos (DCE)² se le conoce como interfase digital.

La RS-232 es una interfase digital estándar para la conexión de componentes del sistema como modems, impresoras y computadoras. El estándar fué establecido por "Electronic Industries Association (EIA)".

1 Las líneas telefónicas son los canales a través de los cuales se realizan la mayoría de los enlaces de equipo de teleproceso.

2 Términos genéricos para el equipo de cómputo y modem respectivamente.

El RS-232 define un camino de señal de 25 conductores, es decir 25 circuitos de intercambio, definidos y asignados a un número de "pin". Estas asignaciones se muestran en la tabla 8.1.

Las funciones del conector estándar RS-232 se pueden dividir en funciones de datos y funciones de control. Las funciones de datos son únicamente, el transmisor y el receptor en los "pins" 2 y 3 respectivamente. Todas las demás funciones son funciones de control, llamadas así debido a que tienen asignadas las señales de estado y comandos que coordinan el comportamiento de los "modems".

Aunque el estándar RS-232 especifica el número de "pins", no especifica el conector. El conector usual se referencia normalmente como un conector DB-25 y está disponible en los géneros macho y hembra, el DB-25P y DB-25S.

Los "pins" del estándar RS-232 que generalmente se utilizan en un puerto serie en un equipo PC se describen a continuación.

Datos Transmitidos (TD "pin" 2)

La línea de transmitir datos envía un dato serial desde el DTE al modem. El DTE no transmitirá un dato a menos de que se hayan activado las siguientes señales.

1. Solicitud de Envío (RTS, "pin" 4)
2. Disposición de Envío (CTS, "pin" 5)
3. Modem Dispuesto a Enviar (DSR, "pin" 6)
4. Terminal Lista (DTR, "pin" 20)

Datos Recibidos (RD "pin" 3)

Esta señal no depende de ninguna función del RS-232; esta señal transporta los datos generados por el modem en respuesta a unos datos transmitidos por un DTE.

Solicitud de Envío (RTS "pin" 4)

Esta señal es enviada por un DTE para informar al modem que se desea realizar una transmisión. De hecho, su única función es cambiar un "modem" "half-duplex" entre los modos de transmisión y recepción.

ASIGNACION DE CONEXIONES DE LA INTERFAS EN LOS ESTANDARES IN-222C

"PIN"	FUNCION
1	TIERRA DE PROTECCION (PROTECTIVE GROUND) PG
2	DATOS TRANSMITIDOS (TRANSMITTED DATA) TD
3	DATOS RECIBIDOS (RECEIVED DATA) RD
4	SOLICITUD DE ENVIO (REQUEST TO SEND) RTS
5	DISPOSICION DE ENVIO (CLEAR TO SEND) CTS
6	MODER DISPUESTO A ENVIAR (DATA SET READY) DSR
7	SEÑAL DE TIERRA (SIGNAL GROUND) SG
8	DETECTOR DE SEÑAL RECIBIDA (RECEIVED LINE SIGNAL ELEMENT TYPING) CLD
9	PRUEBA DE MODER
10	PRUEBA DE MODER
11	SIN DESIGNACION
12	DETECTOR SECUNDARIO DE SEÑAL RECIBIDA (SECONDARY RECEIVED LINE SIGNAL DETECT) SCR
13	DISPOSICION DE ENVIO SECUNDARIO (SECONDARY CLEAR TO SEND) SCTS
14	DATOS TRANSMITIDOS SECUNDARIO (SECONDARY TRANSMITTED DATA) STD
15	SINCRONIZACION DE LA SEÑAL TRANSMITIDA (TRANSMITTER SIGNAL ELEMENT TYPING) CLMTD
16	DATOS RECIBIDOS SECUNDARIOS (SECONDARY RECEIVED DATA) SRD
17	SINCRONIZACION DE LA SEÑAL RECIBIDA (RECEIVED SIGNAL ELEMENT TYPING) CLMRD
18	SIN DESIGNACION
19	SOLICITUD DE ENVIO SECUNDARIO (SECONDARY REQUEST TO SEND) SRTS
20	TERRMINAL LISTA (DATA TERMINAL READY) DTR
21	DETECTOR DE LA CALIDAD DE LA SEÑAL (SIGNAL QUALITY DETECTOR)
22	INDICADOR DE LLAMADA (RING INDICATOR) RI
23	SELECTOR DE VELOCIDAD DE SEÑALIZACION (DATA SIGNAL RATE SELECTOR)
24	SINCRONIZACION DE LA SEÑAL TRANSMITIDA (TRANSMIT SIGNAL ELEMENT TYPING)
25	SIN DESIGNACION

Tabla 8.1

Disposición de Envío (CTS "pin" 5)

Esta señal es enviada al DTE cuando el modem esta listo para recibir datos. (Esta señal es la respuesta a una señal RTS).

Modem Dispuesto a Enviar (DSR, "pin" 6)

Una señal DSR es recibida por un DTE cuando el "modem" está encendido y conectado físicamente y puede transmitir o recibir datos.

Señal de Tierra (GND, "pin", 7)

Este circuito debe estar incluido en todas las interfaces, establece una referencia a tierra para todas las otras señales.

Detector de Señal Recibida (DCD, "pin" 8)

Proporciona un indicador de que las señales de un equipo remoto están siendo recibidas.

Terminal Lista (DTR, "pin" 20)

Señal establecida por un DTE siempre que la comunicación de datos está activa. Su inhibición origina una desconexión del "modem" con la línea de comunicación.

Indicador de Llamada (RI, "pin" 22)

Señal recibida por un DTE cuando el "modem" está recibiendo una llamada.

UART 8250

La tarjeta del puerto serie "Asynchronous Communication Adapter" utiliza un circuito integrado conocido como el 8250 ("Universal Asynchronous Receiver/Transmitter") para generar y procesar las señales descritas en el punto anterior.

DIAGRAMA DE BLOQUES DEL CS20

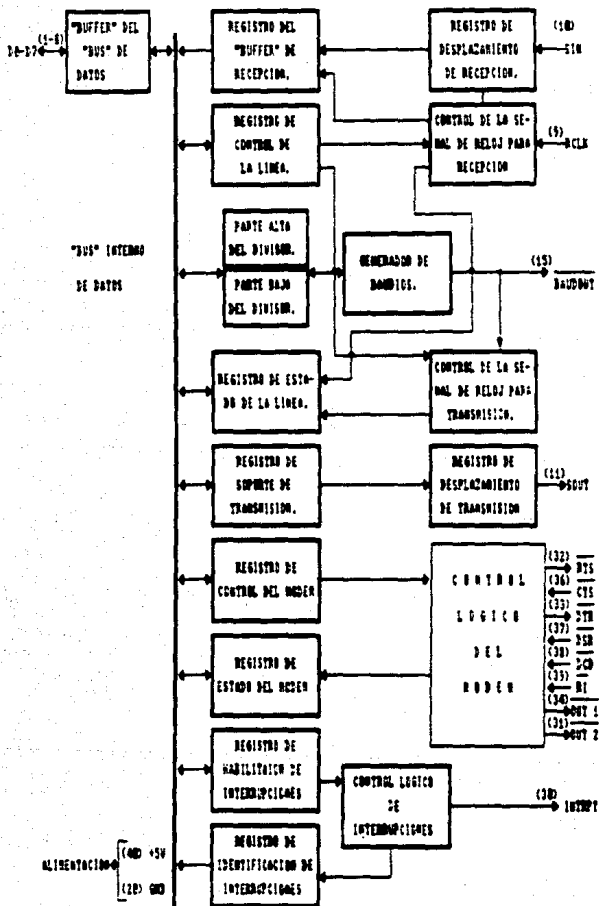


FIGURA 0.3

Para esto el 8250 requiere tres interfaces básicas: El "bus" de E/S del sistema, el reloj y las entradas y salidas del RS-232. El 8250 es conectado a los 8 "bits" menos significativos del "bus" de datos del CPU.

Las operaciones de lectura y escritura son diferenciadas de las entradas y salidas de datos a través de las líneas DISTR y DOSTR. El 8250 tiene varios registros internos, los cuales son direccionados individualmente a través de tres entradas de selección. Por lo que la transmisión de un dato (un "byte") es una operación de tres etapas:

1. El CPU pone el dato en las líneas de datos D0-D7.
2. El número del registro "buffer" de transmisión es puesto en las entradas de selección A0-A2.
3. La lógica en las líneas DISTR y DOSTR mueve el dato de D0-D7 al registro "buffer" de transmisión. El 8250 mueve el dato del registro "buffer" de transmisión al registro de desplazamiento de transmisión, el cual momentos después estará vacío.

La recepción de un dato es similar a la transmisión; en las siguientes etapas se asume que el dato ya ha sido recibido y se encuentra en el registro de recepción.

1. El número del registro "buffer" de recepción es puesto en las entradas de selección A0-A2.
2. Una operación de lectura es ejecutada, de acuerdo a la lógica en las líneas DISTR y DOSTR.
3. Se mueve el dato del registro "buffer" de recepción a las líneas de datos D0-D7 donde es capturado por el CPU.

El UART facilita la transmisión y recepción de datos ya que un programa simplemente tendría que leer y escribir "bytes" al UART, los cuales son vistos por el procesador como localidades ordinarias de memoria o puertos de entrada/salida (registros del 8250).

Registros del 8250.

Registro "Buffer" de Recepción

Registro que contiene el dato que se acaba de recibir. El dato es recibido serialmente, llegando primero el "bit" menos significativo ("bit" 0).

Registro "Buffer" de Transmisión

Este registro contiene el siguiente carácter a ser transmitido serialmente. El "bit" menos significativo ("bit" 0) es el primero que se transmite.

Registro de Habilitación de Interrupción

Este registro permite activar las señales de interrupción (tabla 8.3).

"bit" 0	(RXRD) SI ES 1, UNA INTERRUPCION ES GENERADA CUANDO UN DATO ESTA LISTO PARA LECTURA EN EL "BUFFER" DE RECEPCION.
"bit" 1	(TXDE) SI ES 1, UNA INTERRUPCION ES GENERADA CUANDO UN "BYTE" ES TRANSFERIDO DEL "BUFFER" DE TRANSMISION AL REGISTRO DE CORRIENTE DE TRANSMISION.
"bit" 2	SI ES 1, UNA INTERRUPCION SE GENERA DEDIDO A UN ERROR DE PARIDAD, UN ERROR DE CONSTRUCCION O UNA INTERRUPCION DE BREAK.
"bit" 3	SI ES 1, UNA INTERRUPCION SE GENERA DEDIDO A UN ERROR EN CUALQUIERA DE LAS ENCAPSULAS DEL RS-232C HUBO UN CAMBIO DE ESTADO.
"bit" 4-7	SIEMPRE 0.

TABLA 8.3

Registro de Identificación de Interrupción

Cuando una interrupción ocurre, este registro identifica su origen. Un 0 en el "bit" 0 de este registro significa que una interrupción está pendiente; los "bits" 1-3 identifican la interrupción de acuerdo a la tabla 8.4.

"BIT" 2	"BIT" 1	"BIT" 0	PRIMIDAD	FUENTE DE INTERRUPCION
0	0	1	--	-
1	1	0	0	ERROR SERIALIZACION O "BREAK"
1	0	0	1	DATO LISTO
0	1	0	2	"BUFFER" DE TRANSMISION VACIO
0	0	0	3	ENTRADAS DEL RS-232C

TABLA 8.4

Registro de Control de la Línea

Este registro especifica el formato del dato (longitud del dato, "bits de stop" y paridad).

Registro de Control del Modem

Este registro controla el estado de las 2 salidas del RS-232 DTR y RTS. Un 1 lógico en el "bit" 0 de este registro mantiene la señal de salida "DTR" del 8250. Un 1 lógico en el "bit" 1 mantiene la señal de salida "RTS" del 8250.

Registro de Estado de la Línea

Contiene información sobre el estado de la transferencia de datos y de las condiciones de error asociadas.

Registro de Estado del Modem

Este registro proporciona el estado actual de las señales de control del modem. Es decir, reportan un cambio en el estado de los "pins" del RS-232.

8.3 Características del Equipo A9 para Comunicaciones

En el capítulo 3 se trató en forma general tanto el "software" como el "hardware" del equipo A9, en esta sección se ampliará el subsistema de comunicaciones de dicho equipo.

Se mencionarán elementos tanto de "hardware" como de "software" de este equipo.

El diseño del subsistema de comunicaciones permite que las funciones de comunicación se distribuyan entre los siguientes elementos, el Network Support Processor (NSP), el Line Support Processor (LSP) y Quad Line Adapters (QLA). El Sistema Operativo (MCP) debe ser tan independiente como sea posible del sistema de comunicación³.

La comunicación de datos es la interfase entre usuarios remotos y dispositivos con funciones específicas. Aunque más de una computadora puede estar en la red de comunicación, una será identificada como la computadora central y las otras como terminales remotas.

En el flujo de información de una terminal remota a un programa objeto corriendo en el sistema central, el "hardware" y "software" para la comunicación de datos maneja progresivamente grandes unidades de información y diferentes aspectos de la información. El mensaje es una unidad de intercambio entre las interfaces.

³ En un sistema "main frame", es necesario la presencia de diferentes procesadores, cada uno con funciones específicas de no ser así, el sistema operativo llegaría a ser muy grande y cualquier cambio resultaría complejo. Además, las funciones del subsistema de comunicaciones son en su mayor parte distintas de las del sistema operativo principal.

Terminales

Quando la información es enviada de un dispositivo remoto al sistema central, la terminal transmite la representación en "bits" del carácter por la línea.

Líneas y Modems

La conexión de terminales remotas al sistema de comunicación de datos es usualmente llevada a cabo por líneas telefónicas. Para distancias cortas (terminales en la sala de máquinas o en el mismo edificio) pares de alambres ordinarios pueden ser usados. Esto es llamado conexión directa. Las líneas telefónicas pueden ser de tipo normal o líneas especiales privadas. En la mayoría de los casos deben usarse sistemas moduladores-demoduladores (MODEM) para lograr características aceptables en la línea. Los "bits" de información son alimentados al modulador, transformados en una señal modulada, y entonces transmitidos a través de la línea telefónica. La señal llega a un demodulador, donde es re-transformada al patrón de "bits" original y enviada al sistema de comunicación de datos.

Adaptadores

La línea telefónica es conectada a un adaptador en la A9. Este adaptador es la primera parte de la unidad de "hardware" que encuentra la información. El tipo de adaptador usado depende del dispositivo terminal asociado con esa línea. El propósito de los adaptadores es tomar los "bits" de la línea y acumularlos para formar caracteres.

Network Support Processor

Este procesador es el centro del subsistema de comunicaciones de datos. Dentro de sus funciones de control, están: control de líneas de comunicación y estaciones, transferencia de información y reconfiguración de la red.

Este procesador es programado usando una versión del Burroughs Network Definition Language (NDL II), el cual permite una rápida y eficiente definición e implementación de la red.

Message Control System (MCS)

Este es un programa escrito en DCALGOL, para comunicaciones. La función primaria de un MCS, es como su nombre lo indica, controlar el flujo de mensajes entre la terminal y el sistema principal. Los proveedores proporcionan un MCS estándar, sin embargo, el usuario puede escribir uno propio o modificar el estándar si así lo desea.

Data Communications Controller (DCC)

El DCC es la interfase básica entre el subsistema de comunicación de datos y el sistema principal. El DCC es un subconjunto del MCP (Master Control Program) y opera como un grupo de tareas independientes.

Todos los movimientos de información entre el sistema central y el NSP son controlados por el DCC. El DCC actúa como intermediario entre los MSC's y los programas de aplicación. El DCC también se encarga de checar la validez de las instrucciones pasadas por los MCS's a los NSP's.

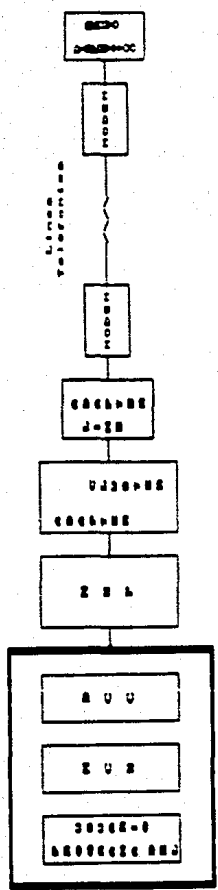
Los mensajes, los cuales contienen requerimientos para el NSP, son ordenados por el DCC en una lista ligada, llamada "Request Queue". El NSP escribe los resultados al DCC a través de otra lista ligada llamada "Result Queue".

Programa Objeto

El programa objeto no tiene que ver con peculiaridades de la red de comunicaciones. El objetivo de un programa objeto es procesar los datos de entrada/salida de acuerdo a las consideraciones de la aplicación.

Mensajes

En la mayoría de los casos la información mandada a , o recibida desde, un dispositivo remoto es el texto de un mensaje. El texto en el sistema de comunicación de datos es siempre precedido por información de control. La información de control y el texto juntos constituyen un mensaje. Los mensajes también son usados en la comunicación entre el DCC, NSP, y el MCS, y el formato varía con el tipo de mensaje.



00000000 00000000 00000000 00000000

00000000 00000000

Network Definition Language (NDL)

El sistema de comunicación de datos es principalmente un conjunto de procesos dirigidos por tabla. En gran parte, esta es la descripción escrita en el lenguaje de definición de la red que se usa para controlar la red de comunicación de datos.

Un programa fuente en NDL describe una red de comunicación de datos física, lógica y funcionalmente. Los componentes físicos de una red de comunicación de datos incluye las especificaciones y capacidades de varios elementos de "hardware" que componen la red. Las características lógicas de una red son las asociaciones entre los varios componentes de un sistema de comunicación de datos (programas de usuario, MCS, ...) y los nombres simbólicos usados para designar elementos físicos dentro de la red.

NDL también especifica el comportamiento funcional o la forma en la cual cada línea de datos es controlada. Estas especificaciones consisten en rutinas individuales permitiendo implementar el protocolo requerido para adecuar las características físicas y aplicaciones de los tipos de terminales que han sido definidas.

Las rutinas son compiladas dentro de un grupo de instrucciones, las cuales son ejecutadas por el NSP.

Una descripción en NDL tiene las siguientes secciones primarias.

-Request Section

Es dentro de esta sección que el usuario da al sistema de comunicaciones las disciplinas de línea a ser usadas con los diferentes tipos de terminales dentro de la red. El programa o descripción provista por el usuario es convertida por el compilador de NDL en un "macro string" subsecuentemente convertido por el NSP ASSEMBLER/generator, DCPROGEN, en código de máquina que es incluido en el sistema operativo del NSP. Por ejemplo, una rutina de disciplina de línea causará una llamada a un procedimiento, el cual permitirá la entrada a un caracter, traducirá el caracter a EBCDIC y checará la paridad de los caracteres.

-Modem Section

Esta sección es usada para describir los modems que pueden ser incluidos dentro de la red de comunicación de datos. Es importante tomar en cuenta ciertas características como formato del caracter y velocidad de los tipos de adaptadores con los cuales el modem puede ser conectado.

-Terminal Section

Dentro de esta sección, el usuario describe aquellos aspectos físicos de los diferentes tipos de dispositivos terminales que pueden estar dentro de la red de comunicación de datos, los cuales son requeridos con el propósito de que el "software" de comunicaciones pueda manejar propiamente transmisión y recepción de mensajes con ese tipo de dispositivo.

-Station Definition

La sección de estaciones proporciona al usuario la habilidad de especificar los aspectos lógicos de los elementos remotos dentro de la red. Además de especificar con que tipo de terminales está asociada una estación.

-Line Section

Esta sección proporciona al usuario el medio para describir las características físicas y lógicas de las líneas en el sistema de comunicaciones. Dentro de la definición de una línea, la clase de adaptador debe ser especificada. Adicionalmente, el tipo de línea debe ser implícita o explícitamente especificado. Por omisión es conexión de línea privada o directa dependiendo de una especificación de un modem para la línea.

Procesador NDL

Proporciona la configuración de la red y las macros asociadas con las rutinas de disciplina de la línea que habilitan a DCPROGEN para proporcionar el código necesario para el sistema operativo del NSP para realizar la propia comunicación con la parte de comunicaciones remotas del MCP, el Data Communications Controller (DCC).

8.4 Diseño General del Sistema

Para la comunicación se requiere "software" a diferentes niveles, independientemente del tipo de conexión que se realice entre el microcomputador y el equipo central.

Cuando se envía o se recibe un mensaje, este pasa por diferentes etapas, en las que intervienen distintos elementos de "hardware" de los equipos, en la mayoría de los sistemas de computadoras hay diferentes niveles de protocolos los cuales gobiernan el intercambio de información. Estos van desde los niveles físicos de intercambio de datos entre un modem y una terminal (protocolo RS232-C), hasta el intercambio de información entre un

programa de aplicación y el usuario final. El diseño que se realizará se planteará de acuerdo a los siguientes niveles:

1. Conexión Física
2. Detección y Corrección de Errores
3. Comandos para Transferencia

En los que consideramos que existen distintos protocolos a diferente nivel.

Nivel 1

En este nivel consideraremos el uso de la interfase RS-232C. A través de esta norma se realizará la conexión física de los computadores.

Nivel 2

En este nivel consideramos al "software" que se requiere para implementar un protocolo que permita detectar y corregir errores en la transmisión de datos.

Las consideraciones que se toman en cuenta para el diseño de un protocolo de este tipo varían según la aplicación que se requiera así como los equipos en los que se va a implementar.

En general, se basan en dos puntos principalmente:

1. La interfase de la línea de comunicación
2. La interfase funcional entre los diferentes componentes del sistema

La primera es un protocolo de línea que proporciona el enlace entre una terminal remota y el sistema central, cuyos objetivos principales son, por un lado maximizar el "throughput"⁴ de la línea considerando aspectos como enviar múltiples registros por

4 Información útil procesada durante un período de tiempo específico.

mensaje, aplicar técnicas de compresión y expansión de campos para reducir el tiempo de transmisión de ciertos datos.

Por otro lado, también se requiere de un protocolo de línea la capacidad de detectar y recuperar errores en la línea, teniendo en consideración casos como :

- La pérdida de mensajes por fallas en los modems
- Recuperación parcial de mensajes
- Recepción de mensajes con basura
- Errores causados por interferencias eléctricas en la línea.

Un protocolo que contempla las características antes mencionadas es el protocolo RJE, el cual es proporcionado por los proveedores del equipo AS, y se considera apropiado para una comunicación serial, por lo que se propone su uso en este nivel.

Protocolo de Comunicación de Datos RJE (B-771).

El formato del mensaje estará en código ASCII-67, y tendrá la siguiente configuración:

`/ SOH / ARM / STX / DA1 / DA2 / < TEXTO > / ETX / BCC /`

El significado de cada uno de los mensajes es el siguiente:

SOH (Start of Header)

Este es el caracter estándar ASCII (01H); el cual indica que se inicia la transmisión.

ARM (Acknowledge Received Message)

Este caracter es utilizado para indicar un reconocimiento de mensaje, ya sea positivo o negativo, al mensaje previo, se forma de acuerdo a la paridad del mensaje y al número de la transmisión (TN).

Durante los mensajes iniciales, este caracter se inicia como el caracter NULL (00H). Cuando la secuencia (SOH-ARM) es recibida, esta es comparada con el último número de transmisión enviado. Si el ARM concuerda con el TN, el último mensaje enviado ha recibido acuse de recibo. Si el caracter no concuer-

da, el último mensaje debe ser retransmitido. Cuando un mensaje válido es recibido, el número de la transmisión contenido en ese mensaje es guardado para la utilización del siguiente carácter transmitido.

TN (Transmission Number)

Este carácter es conocido como el número de la transmisión, este es la porción del mensaje que se utiliza para propósitos de acuse de recibo de mensajes, ya sean positivos o negativos. Se utilizan solo dos caracteres de transmisión, estos son: "T" y "U".

STX (Start of Text)

Este es el carácter estándar ASCII de STX (02H), el cual sirve como delimitador entre la porción del encabezado precedente y la porción del texto del mensaje.

DA1 y DA2 (Device Address)

Estas dos direcciones de dispositivos se utilizan para asociar un mensaje con uno de los dispositivos remotos en particular. Estos caracteres son utilizados por la computadora para rutear datos al dispositivo remoto, y son insertados en los mensajes al ser transmitidos desde estos dispositivos.

ETX (End of Text)

Este es el carácter estándar ASCII (03H); el cual es utilizado para indicar el fin del texto en un mensaje.

BCC (Block Check Character)

Este es el carácter de la paridad del mensaje. Este representa el "X-OR" de todos los caracteres que siguen al carácter "SOH" incluso hasta el carácter ETX.

El propósito de este carácter es el de asegurar la recepción correcta de los mensajes en el otro lado. Los mensajes conteniendo caracteres de BCC incorrectos recibirán acuse de recibo negativo.

Descripción del Flujo del Protocolo de Línea RJE

La descripción del flujo de mensajes se muestra en la figura B.6.

PROYECTO DE COMUNICACION BLX

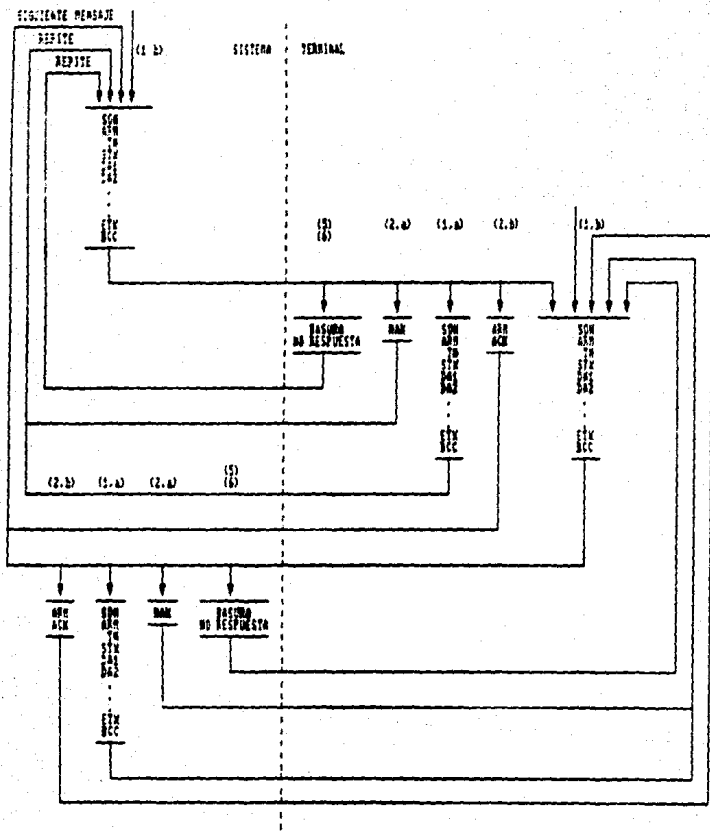


FIGURA 0.4

Se asume que el sistema transmite un mensaje a la terminal y espera respuesta de la terminal.

1. Si la terminal receptora tiene un dato para transmitir al sistema después de recibir un mensaje del sistema, la terminal acusa de recibido positiva o negativamente según sea el caso:

a) Acuse Negativo

El mensaje recibido contiene errores de paridad, o el mensaje tiene un formato inválido. Como resultado el TN del último mensaje válido recibido es retornado en el campo ARM del mensaje transmitido al sistema.

b) Acuse Positivo

El mensaje recibido fué válido y el TN del mensaje recibido es retornado en el campo ARM del mensaje transmitido al sistema.

2. Si la terminal no tiene datos para transmitir al sistema después de haber recibido un mensaje del sistema, la terminal acusa la recepción del mensaje positiva o negativamente según sea el caso:

a) Acuse Negativo (NAK)

El mensaje recibido fué inválido; como resultado, un simple caracter ASCII de control (NAK) es transmitido al sistema.

b) Acuse Positivo (ACK)

El mensaje recibido fué válido, y un mensaje constituido por dos caracteres el TN del mensaje recibido, seguido por el caracter de control ACK es transmitido al sistema.

El sistema responde a los mensajes de acuse de la terminal en la siguiente forma:

3. Si un caracter NAK es recibido desde la terminal (caso 2.a) o si la respuesta de la terminal es un mensaje conteniendo texto (caso 1.a) con un caracter ARM que concuerda con el último TN enviado, entonces el sistema retransmite el último mensaje enviado sin cambiar el TN. El sistema acusa positiva o negativamente los mensajes de respuesta poniendo el TN del último mensaje válido recibido desde la terminal en el campo ARM del mensaje retransmitido.

Si el campo ARM del mensaje recibido desde la terminal concuerda con el TN del último mensaje transmitido (casos 1.a y 2.b), el sistema incrementa el TN almacenado y transmite el siguiente mensaje a la terminal.

5. Si el sistema no puede reconocer la respuesta de la terminal, el sistema asume que un acuse negativo ha sido recibido, lo que causa la retransmisión del último mensaje enviado sin cambiar el TN. Si la terminal había acusado positivamente el último mensaje recibido, la terminal ignora el mensaje retransmitido y envía un acuse positivo.

6. Si la terminal no responde en un tiempo determinado al último mensaje enviado por el sistema, el sistema retransmite el mensaje sin cambiar el TN. De esta manera el ciclo regresa a la etapa 1.

Cabe aclarar que en el enlace del equipo PC y el equipo A9, el "main frame" cuenta con protocolos que permiten tener un enlace eficiente a este nivel (RJE), sin embargo, en el equipo PC las consideraciones antes mencionadas tendrían que ser implementadas o bien hacer uso de uno ya existente para este equipo.

Nivel 3

Este nivel corresponderá al "software" que a nivel de comandos permitirá al usuario la transferencia de archivos y que tomará las acciones necesarias para obtener y alojar los archivos en cada equipo.

A continuación se muestra el flujo de mensajes requerido para las tareas que se realizarán en ambos equipos.

PROCOLO DE COMUNICACION

PC

A9

(*SUBPROGRAMA en 09*)

OSUBMIO . . .
97
(ONE)

IF ONE

° Selecciona Opcion

CTRL T

CTRL R

CTRL F

Opcion ?°

(*FIN*)

TRANSFERENCIA DE ARCHIVOS

PC

A9

```
IF CTRL T
  *Nombre del Archivo de Datos ?*
  Verifica que exista el Archivo de Datos.
```

```
IF NO EXISTE
```

```
(ONE)
```

```
ELSE
```

```
Busca el Archivo de Control Asociado al
Archivo de Datos.
```

```
IF NO EXISTE
```

```
(ONE)
  GO FIA
```

```
ELSE
```

```
(ONE)(ONE)ER
```

```
Envia Respuesta.
```

```
IF ONE
```

```
Abre Archivo de Datos.
Prepara Informacion para Enviar.
```

```
(Primer Bloque de Datos)
```

```

  .
  .
  .
```

```
(Ultimo Bloque de Datos)
```

```
(fin de Archivo)
```

```
Termina Programa.
Regresa a COME.
```

```
IF ONE
```

```
GO FIA
```

```
ELSE
```

```
Valida Header.
```

```
IF ERROR
```

```
(ONE)("Error en la Clave del Usuario")
```

```
ELSE
```

```
IF ARCHIVO EXISTE
```

```
(ONE)("El Archivo Existe. Concatenacio
L/W")
```

```
ELSE
```

```
(ONE)("Envia Datos")
```

```
(ONE)
```

```
(ONE)
```

```
(ONE) Cierra Archivo
Termina Programa.
```

PC

A9

IF CTRL R

Verifica que el Archivo de Datos No Exista.
Verifica que haya Suficiente Espacio.

IF ERROR

GO FIN

ELSE

(ONE)(Nombre del Archivo de Datos)

IF OE

GO FIN

ELSE

Verifica si Existe el Archivo.

IF ERROR

(OE)("No Existe el Archivo")

GO FIN

ELSE

(ONE)(Envia Header)

IF OE

GO FIN

ELSE

Crea Archivo de Control.
Abre Archivo para Datos.
(ONE)(Envia Datos)

IF OE

GO FIN

ELSE

Prepara Informacion a Enviar.

(Primer Bloque de Datos)

:

(Ultimo Bloque de Datos)

(ONE)

(ONE)

(Fin de Archivo)

(ONE) Cierra Archivo.

Termina Programa.

Termina Programa.

FIN DE PROGRAMA

PC

A9

IF CIRC F

FIN: Termina Programa

" "	Mensaje
< Mensaje >	Envía Mensaje al Otro Equipo
OME	Mensaje que indica que no ocurrió error
OE	Mensaje que indica la ocurrencia de un error

Una vez que se han definido los protocolos de cada uno de los niveles planteados, el siguiente paso es la realización de la instrumentación de estos, la cual no se llevó a cabo, debido a que en el tiempo en el que se desarrolló el sistema de captura, fué desarrollado un sistema de transferencia de archivos que satisface ampliamente las necesidades que se tienen para la comunicación con el equipo central.

Por lo tanto no se justifica el invertir recursos y tiempo en el desarrollo de esta parte del sistema, solo se plantea la base teórica de su desarrollo.

CONCLUSIONES

Tomando en cuenta la experiencia que se tiene desde que se implantó el sistema, podemos decir, que se encuentra funcionando en forma satisfactoria; aunque claro está, no exento de algunos problemas, como fallas en el equipo, la aceptación del sistema por parte del usuario, así como el manejo adecuado de este. No obstante estos detalles, el sistema ha proporcionado una herramienta útil y sencilla que permite agilizar los procesos del departamento, teniéndose en cuenta también que como todo producto de "software" es posible hacerlo más eficiente, lo cual se considerará en la fase de mantenimiento.

El desarrollo del presente trabajo representa experiencia adquirida en diferentes aspectos. Por una parte, la comunicación con el usuario, punto que es muy importante ya que muchas veces el analista y el usuario tienen una visión diferente tanto del problema como de su solución. El buen entendimiento de ambas partes lleva al éxito de un sistema.

Por otra parte, el desarrollo del trabajo en equipo, con la idea de que la programación y la documentación se realice en forma ordenada siguiendo alguna metodología acordada previamente, de tal manera que sean fáciles de entender los documentos y programas del sistema tanto por los miembros del equipo como por personas ajenas al proyecto.

Una reflexión final que deseamos hacer es sobre la importancia que reviste el tener acceso al conocimiento, el ser parte del proceso de evolución en cualquier disciplina y señalar que este proceso es en ambos sentidos ya que el ser humano, el ser pensante se transforma así mismo durante el aprendizaje, y se da la valiosa oportunidad de conocer y crecer con el mundo circundante.

APENDICE A

RUTINAS DE USO GENERAL

En la mayoría de los sistemas se requiere instrumentar algunas rutinas de uso general que pueden ser utilizadas por cualquier programa del sistema. A continuación se mencionan el propósito y parámetros de cada una.

Rutina : `esc1(c,atr,pag)`

Función : Despliega caracteres en la pantalla.

Parámetros de Entrada :

`c` - Caracter a desplegar

`atr` - Atributo de video

`pag` - Número de página

Parámetros de Salida : Ninguno.

Código Fuente de la Función:

```
esc1(c,atr,pag)
char c;
int atr,pag;
{
  AH = 0x00;
  AL = c;
  BH = atr;
  BH = pag;
  CX = 1;
  geninterrupt(2x10);
}
```

Esta función, como las demás para manejo de video, hace uso de algunas facilidades del lenguaje C, que nos permiten un acceso más directo a los registros del procesador,

siendo esto necesario, ya que este tipo de funciones se implementaron a través de interrupciones.

Rutina : poner(x,y,c,atr,pag)

Función : Despliega un caracter en la pantalla, en una posición específica y con un atributo de video.

Parámetros de Entrada :

x - Renglón

y - Columna

c - Caracter a desplegar

atr - Atributo de video

pag - Número de página

Parámetros de Salida : Ninguno.

Rutina: cursor(x,y,pag)

Función : Posiciona el cursor en un lugar específico de la pantalla.

Parámetros de Entrada :

x - Renglón

y - Columna

pag - Número de página.

Parámetros de Salida : Ninguno.

Rutina: pagina(pag)

Función : Selecciona la página de video en la que se quiere trabajar.

Parámetros de Entrada :

pag - Número de página

Parámetros de Salida : Ninguno.

Rutina : ventana(x1,y1,x2,y2,atr,pag)

Función : Abre una ventana con algún atributo de video.

Parámetros de Entrada :

x1 - Coordenadas de la esquina superior

y1 - izquierda

x2 - Coordenadas de la esquina inferior

y2 - Derecha

atr - Atributo de video

pag - Número de página

Parámetros de Salida : Ninguno.

Rutina : marco(x1,y1,x2,y2,atr,pag,tipo)

Función : Despliega un marco en la pantalla, seleccionando el tipo de

marco que se desea.

Parámetros de Entrada :

x1 - Coordenadas de la esquina superior

y1 - izquierda

x2 - Coordenadas de la esquina inferior

y2 - Derecha

atr - Atributo de video

pag - Número de página

tipo - Tipo de marco que se desea

1. margen delgado

2. margen grueso

3. margen doble

Parámetros de Salida : Ninguno.

Rutina : linea(x1,y1,c,atr,n,pag)

Función : Despliega una línea horizontal en la pantalla.

Parámetros de Entrada :

x1 - Renglón y columna de la posición

y1 - inicial

c - Caracter con el cual se desea generar la línea

atr - Atributo de video

n - Longitud de la línea en caracteres

pag - Número de página

Parámetros de Salida : Ninguno.

Rutina : p_mensaje(i,j,mens1,atr,pag)

Función : Despliega una cadena de caracteres en la pantalla.

Parámetros de Entrada :

i - Renglón y columna de la posición

j - donde se va a desplegar la cadena

mens1 - Apuntador al inicio del mensaje

atr - Atributo de video

pag - Número de página

Parámetros de Salida : Ninguno.

Rutina : p_error(mens_e,atr,pag)

Función : Despliega un mensaje de error en el sistema. Este mensaje aparece centrado en el renglón 23 de la pantalla

Parámetros de Entrada :

men_e - Apuntador al inicio del mensaje

atr - Atributo de video

pag - Número de página

Parámetros de Salida : Ninguno.

Rutina : des_var(x,y,var,tam,atr,pag)

Función : Despliega el contenido de una variable tipo string

Parámetros de Entrada :

x - Rengión y columna de la posición en

y - la que se va a desplegar la variable

var- Apuntador al inicio de la variable

tam- Longitud de la variable

atr- Atributo de video

pag - Número de página

Parámetros de Salida : Ninguno.

Rutina : lee_car(c)

Función : Lee un caracter del teclado

Parámetros de Entrada :

c - Variable en la que se almacena el caracter leído

Parámetros de Salida :

c - Variable leída

sc - Número de Tecla

Rutina : lee_var(x,y,var,tam,tipo,atr,pag,f)

Función : Lee una cadena de caracteres desde el teclado

Parámetros de Entrada :

x - Rengión y columna de la posición en

y - la que se va a desplegar la variable

var - Apuntador al inicio de la variable

tam - Longitud de la cadena al leer

tipo - Tipo de validación

atr - Atributo de video

pag - Número de página

t - Bandera

Parámetros de Salida :

Caracter de salida

-ESC

-'\\0'

-c

-sc

-RET

Rutina : verif_arch(x,y,arch,tam,tipo,ext,atr,pag)

Función : Verifica la existencia de un archivo en el directorio

Parámetros de Entrada :

x - Renglón y columna de la lugar en el

y - que se va a desplegar

arch - Apuntador al inicio del nombre del archivo

tam - Longitud del nombre del archivo

tipo - Tipo de validación

ext - Extensión del archivo

atr - Atributo de video

pag - Número de página

Parámetros de Salida :

Estado del archivo

ESC- Aborta rutina

0- Existe en el directorio

1- No existe en el directorio

Rutina : ob_num(i,j,pnum,tam,atr,f)

Función : Lee una cadena de números desde el teclado

Parámetros de Entrada :

x - Renglón y columna de la posición de
y - la cadena
pnum - Apuntador al inicio de la variable
tam - Longitud de la cadena al leer
atr - Atributo de video
t - Bandera

Rutina : crea(archivo)

Función : Crea un archivo en el directorio actual

Parámetros de Entrada :

archivo - Apuntador al inicio del nombre del archivo

Parámetros de Salida :

fh - "File handle" del archivo

-1 - Error en la creación del archivo

Rutina : val_car(car,tipo)

Función : Valida caracteres

Parámetros de Entrada :

car - Caracter a validar

tipo - Tipo de validación

A - Alfabético

a- Alfabético o blanco

9 - Numérico

X - Alfanumérico

x - Alfanumérico o blanco

* - Cualquier caracter

Parámetros de Salida :

-1 - Caracter inválido

BIBLIOGRAFIA

Fairley Richard. Ingeniería de Software, McGraw-Hill, 1ª ed., 1987, pp. 390.

Schildt. Herbert, C The Complete Reference, Covers Microsoft C, pp. 773.

Duncan. Ray, Advanced MS-DOS, Microsoft Press, 1986, pp. 468.

Kruglinski. David, Guía a las Comunicaciones del IBM PC, Osborne/McGraw-Hill, 1ª ed., 1985, pp. 300.

Tugal. Dogan A., Osman Tugal, Data Transmission, McGraw-Hill 2ª ed., 1982, pp. 493.

Rivers. Ponce Mario Alberto, Tesis "Enlace de una Micro-computadora con una Burroughs B6700", Director de Tesis M. en C. Angel Kuri Morales, Facultad de Ingeniería U.N.A.M., 1982, pp 226.

Salgueiro. Ergueta Emilio, Tesis "Consideraciones de Hardware para la Conexión de Terminales a una Red de Teleproceso", 1981, pp. 214.

Burroughs Corporation, Introduction to Large Systems Student Guide EP-4195, pp. 200.

Burroughs Corporation, B6700 Data Communications Functional Description, Information Manual, pp. 56.

Burroughs Corporation, A-Series Remote Job Entry (RJE) Reference Manual, 1985, pp.260.

Borland International, Turbo C Reference Guide, 1988, pp. 608.

Borland International, Turbo C User's Guide, 1988, pp. 608.