



B1-13148

DON
J. L. de



CINVESTAV-IPN
Biblioteca de Ingeniería Eléctrica



FB0000009732

**CENTRO DE INVESTIGACION Y DE ESTUDIOS
AVANZADOS
DEL
INSTITUTO POLITECNICO NACIONAL**

**DEPARTAMENTO DE INGENIERIA ELECTRICA
SECCION DE COMPUTACION**

**CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL**

**I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA**

SOFTWARE DE SERVICIOS PARA UN CONMUTADOR DE DATOS

Tesis que presenta la L.C. GRACIELA ROMAN ALONSO
para obtener el grado de

MAESTRO EN CIENCIAS

en la especialidad de
INGENIERIA ELECTRICA CON OPCION EN COMPUTACION

Trabajo dirigido por el M. en C. Andrés Vega García

México, D.F., Agosto de 1992



X14

CLASIF.	22.5
ADQUIS.	101-13448
FECHA:	Y-52
PROCED.	OK
\$	

A mis padres, Baltazar y Luisa,
por todo cuanto han hecho por mí.
Por darme amor y libertad, respeto y protección

A mis hermanos Rosario, Irma, y Alfredo
amigos de verdad y ejemplos de perseverancia en el trabajo.
Quienes siempre han creído en mí.

A Alejandro Guzmán De León,
de quien estoy orgullosa por su gran calidad humana.
Por todo el empeño y cariño que me ha dedicado.

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

Agradecimientos

Deseo hacer patente mi agradecimiento a las siguientes personas:

Al M. en C. Andrés Vega García, quien aceptó ser mi director de tesis después del retorno del Dr. Jan Janecek a su país. Sus consejos y observaciones son invaluable.

Al Dr. Sergio Chapa y al M. en C. Jorge Buenabad, por su paciencia en la lectura y revisión del presente trabajo.

A todos los profesores que han contribuido a mi formación. En particular a la planta docente del CINVESTAV.

A mis compañeros de este centro de estudios, de quienes siempre recibí un aliciente para seguir adelante y con quienes pasé momentos muy agradables.

A la Lic. Ma. Cristina Vega Rubio, quien gentilmente puso a mi disposición una de las impresoras de su empresa, contando más que nada con todo su apoyo y amistad.

A la familia Barrientos Camarena, quienes siempre me han acogido en su casa y me han permitido el uso de su impresora, siempre que lo he necesitado.

A mi tía y abuelo Emma y Pascual Alonso, con quienes he vivido los últimos dos años. De no ser por ellos, la culminación de este trabajo se hubiera visto obstaculizada por otros factores.

Para con todos ellos no tengo sino una infinita gratitud.

Graciela Román Alonso.

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

INDICE

I. Introducción	4
I.1 Objetivo	4
I.2 Antecedentes	4
I.3 Justificación del proyecto	4
I.4 Descripción operativa del software propuesto	5
II. Ambiente de trabajo	9
II.1 Arquitectura del conmutador	9
II.2 El núcleo de concurrencia	12
III. Descripción de servicios	15
III.1 Usuario maestro	15
III.2 Altas usuarios	15
III.3 Bajas usuarios	16
III.4 Modificación de atributos de usuario	17
III.4.1 Otorgamiento y restricción de acceso a servicios	17
III.4.1.1 Conexión inmediata	17
III.4.1.2 Cola de espera	18
III.4.1.3 Desalojo de usuarios de baja prioridad	18
III.4.1.4 Conexión y desconexión externa	19
III.4.2 Otorgamiento y restricción de acceso a puertos	21
III.4.3 Clave de acceso	22
III.4.4 Asignación de jerarquía	22
III.5 Especificación de grupos de puertos	23
III.6 Modificación de los atributos de puerto	24
III.7 Comunicación punto o multipunto	25
III.8 Correo electrónico	25
III.9 Deshabilitación de la interrupción del enlace por otros usuarios	26
III.10 Deshabilitación del eco	26
III.11 Enlace permanente	26
III.12 Estadísticas	27
III.13 Estado del conmutador	28
III.14 Menú de dispositivos libres	28
III.15 Menú de servicios	28
III.16 Temporización de uso de puertos (dispositivos)	28
III.17 Temporización de uso del sistema	29
III.18 Resumen de la especificación sintáctica de comandos	30
IV. Elementos básicos del software	37
IV.1 Estructuras de datos importantes	37

Software de servicios	
IV.2 Funciones básicas	43
IV.2.1 Funciones de inicialización	43
IV.2.2 Funciones del conmutador	50
IV.2.3 Funciones del analizador lexicográfico	51
IV.3 Servicios para usuarios	52
V. Manejador de E/S	68
V.1 Funciones de entrada	68
V.2 Funciones de salida	71
VI. Conclusiones	74
Apéndice A	
Manual de usuario	76
Apéndice B	
Módulo de ayuda	81
Apéndice C	
Simbología de LED	85
Bibliografía	90

CENTRO DE INVESTIGACION Y DE
ESTUDIOS AVANZADOS DEL
I. P. N.
BIBLIOTECA
INGENIERIA ELECTRICA

I.1 Objetivo

Implantación de servicios a un conmutador de comunicación digital de datos ya existente, mediante el diseño de un paquete de programas.

I.2 Antecedentes

En el CINVESTAV se desarrolló el hardware para un sistema que funciona como un conmutador de datos digitales de baja velocidad (DataCon 30), controlado por un microprocesador 8086.

El hardware fué probado mediante un programa de control que cumple con funciones mínimas de comunicación entre los canales.

I.3 Justificación del proyecto

En la actualidad, en el mercado existe una tendencia a desarrollar sistemas de comunicación de tipo digital, que tienden a abatir costos y aumentar su funcionalidad.

Los sistemas más populares se enfocan al servicio de microcomputadores y ofrecen servicios básicos como comunicación continua entre usuarios y periféricos de manera transparente. Otros sistemas son capaces de atender a varios usuarios de manera simultánea sin afectar su funcionalidad.

En el CINVESTAV se desarrolló un sistema que pretendía, de manera experimental, atender las necesidades de comunicación digital en el mercado nacional de manera que sustituyera los equipos importados que en él se manejan.

Una vez diseñado y armado el conmutador, se hace imprescindible un software de manera tal que pueda implementar las funciones básicas, buscando la comunicación de manera fácil y transparente entre los usuarios, agregando además servicios auxiliares. De esta manera, si se tuviera al conmutador (DataCon 30) configurado como se muestra en la figura I.1, conectado a 7 puertos serie de un computador GOULD, a 5 puertos serie de un computador PDP-11, a 3 impresoras y 15 usuarios, cada usuario podrá acceder a cualquiera

de los computadores grandes a través de los puertos conectados al conmutador, o a cualquier impresora, favoreciendo el aprovechamiento de los recursos disponibles.

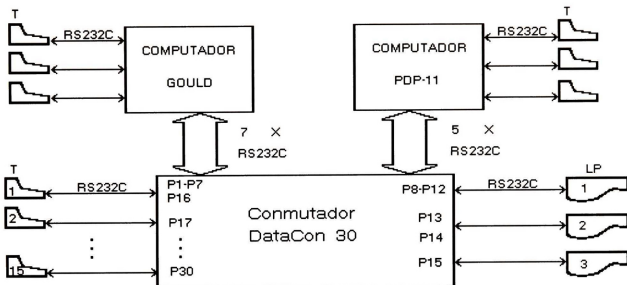


Fig. I.1

I.4 Descripción operativa del software propuesto

Por lo que se ha mencionado, el sistema de servicios deberá proporcionar herramientas sencillas para una adecuada funcionalidad del sistema, permitiendo a los usuarios, mediante la ejecución de algunos comandos, el mejor aprovechamiento de los recursos mediante el conmutador.

El sistema en general se soporta de dos partes esenciales, la primera son los usuarios que pueden gozar de sus servicios, y la segunda son los dispositivos o puertos que administra el conmutador, tomando en cuenta algunas de sus características importantes.

Basicamente, se gira alrededor de un conjunto de usuarios que demandan conexión con algún otro dispositivo, existiendo entre ellos exclusivamente un usuario distinguido por el sistema, denominado usuario maestro, que podrá establecer en determinado momento, a qué recursos y servicios tiene acceso cada uno de los usuarios restantes. Para tal efecto se ha atribuido un conjunto de características a todo usuario dentro del sistema que serán asignadas por el usuario maestro, tales como:

- Un identificador único mediante el cual es reconocido, como por ejemplo el registro federal de contribuyentes
- Una clave de acceso al sistema o contraseña, modificable por cada usuario
- Su jerarquía
- La máscara de acceso a puertos
- La máscara de acceso a servicios

En lo que se refiere a la administración de usuarios, el usuario maestro se encargará de darlos de alta o baja, mediante algunos comandos especiales y podrá modificar sus atributos en el momento que se requiera (exceptuando el identificador único que los distingue).

Considerando que cada dispositivo tiene características y propiedades diferentes, se ha asignado también un conjunto de atributos a cada uno de los puertos que integran el conmutador, esto son:

- Atención

El atributo de atención, indica qué dispositivos son formados en la cola de espera de solicitudes y cuales son ignorados, es decir no se toma en cuenta si el puerto está encendido o apagado. Esto es útil cuando se desean establecer conexiones permanentes.

- Activo

Este atributo indicará cuándo un puerto permite la interacción con el sistema, o bien si es un dispositivo pasivo, el cual no puede conversar.

- Eco

Dependiendo del valor que tenga el atributo Eco de un dispositivo (0/1), el conmutador determinará a qué puertos les regresará los caracteres recibidos.

- Paridad

Para cada puerto se establece paridad par, impar o no paridad.

- Velocidad

Expresa la velocidad de transmisión .

De acuerdo con estas propiedades, el sistema sabrá qué acciones deberá efectuar, cuando haya que establecer alguna conexión o desconexión de algún dispositivo.

El usuario maestro, podrá cambiar en cualquier momento los atributos de algún puerto específico.

Los servicios que ofrece el sistema son los siguientes:

- El de conexión

Que todos los usuarios pueden emplear, bajo el grupo de puertos permitidos otorgado por el usuario maestro.

- Cola de espera

Un usuario específico podrá quedar en cola de espera de algún otro puerto, si éste se encuentra ocupado.

- Conexión inmediata

Al iniciar una sesión de atención, si algún usuario posee el acceso al servicio de conexión inmediata, podrá conectarse automáticamente a algún puerto que se encuentre disponible, dentro del grupo de puertos que él haya definido con anterioridad, sin necesidad de pedirlo en forma explícita.

- Desalojo de usuarios de baja prioridad

La mayor prioridad la tiene el usuario maestro (Jerarquía=0), y así éste puede asignar a todos los demás usuarios su prioridad correspondiente (Jerarquía>0).

Con este atributo, el usuario que solicite la conexión con algún otro dispositivo X que ya esté conectado, podrá romper su conexión y posteriormente conectarse, si acaso los integrantes de dicha conexión, tienen menor prioridad que él.

- Conexión y Desconexión externa

Si un usuario se conecta al sistema mediante el puerto i, podrá realizar conexiones o desconexiones de los puertos j, k siempre y cuando j,k sean distintos de i.

En los capítulos siguientes, se hablará del ambiente de trabajo mediante el que se desarrolló el sistema, hablando en general del conmutador y el núcleo de concurrencia que se utilizó, en seguida se tratarán con mayor detalle los comandos y servicios que ofrece el sistema, así como la especificación sintáctica de cada uno de ellos, continuando con la explicación de las rutinas de software y estructuras de datos más importantes que lo constituyen, para terminar con las operaciones de Entrada / Salida y las conclusiones. Añadiendo como apéndices, el manual de usuario, el módulo de ayuda y la simbología retomada del lenguaje de especificación y de descripción de sistemas de conmutación y telecomunicaciones.

II. AMBIENTE DE TRABAJO

II.1 Arquitectura del Conmutador

ESPECIFICACIONES TECNICAS

Microprocesador	8086 de Intel
Memoria	128 KBytes
Configuración	Programable
Puertos Seriales	Mínimo 8 máximo 30 (síncronos o asíncronos)
Distancia entra puertos seriales	150 mts.
Velocidad de transmisión	300 - 9600 bits/seg.
Señales de control del canal	EIA RS232 C
Alimentación	110v. +/- 10% @ 60 Hz.
Potencia	80 watts
Temperatura de operación	5 - 40 grados centígrados
Humedad	0 - 95% sin condensar
Altitud	0 - 4500 mts.

El conmutador de datos DataCon 30, consta básicamente de 3 bloques funcionales, éstos son:

- Red de conmutación e interfaz
- Memoria de comando
- Procesador de control

cuya operación y función se tratan brevemente a continuación.

- Red de conmutación e interfaz

DataCon 30 conmuta puertos que cumplen con la norma EIA RS232C (v.24 y V.28 del CCITT). La tarea principal de la Red de Conmutación, es de realizar la conexión física entre los puertos requeridos. Su capacidad máxima es de 32 puertos, 2 de los cuales se utilizan por el procesador interno. En su configuración mínima se tienen 8 puertos, de los cuales 2 son de uso interno, la capacidad puede incrementarse de 8 en 8 hasta 32 puertos. La red de conmutación con capacidad máxima está implantada en 4 tarjetas idénticas, llamando a cada una de estas, tarjeta de puertos. La red de conmutación, en

sus diferentes capacidades, carece de bloqueo interno, es decir, dados dos puertos libres, siempre será posible la interconexión entre ellos. Una conexión proporcionada por la red de conmutación es unidireccional, por lo que si es preciso establecer una comunicación bidireccional, tendrá que implantarse mediante el establecimiento de dos conexiones unidireccionales.

Las líneas de la norma EIA RS232C se pueden dividir en dos grupos:

Señales de control (DTR,CTS,DSR,RTS,etc) y

Señales de información (TxD,RxD).

Cabe mencionar que las señales de control no cambian tan frecuentemente como las señales que llevan datos (información). Ambos tipos de señales se conmutan para formar un enlace.

La información que se transfiere por las líneas es digital, por lo que la interfaz es sencilla. Tanto para las señales de datos TxD y RxD como para las de control RTS y CTS, se utiliza la conmutación temporal con transferencia en serie, usando la conmutación por paquete para las señales DTR y DSR.

Con objeto de que el procesador de control se entere de una solicitud de conexión o desconexión, se cuenta con un circuito por medio del cual el procesador explora la existencia de solicitudes. El circuito de explorado junto con un accionador, se usan para realizar la conmutación por paquete de las señales DTR a DSR, leyendo las 8 señales DTR de una tarjeta, y pudiendo actualizar también las 8 señales DTR de alguna tarjeta. Se nota que el circuito de explorado tiene dos funciones, una para detectar solicitudes de conexión y desconexión (mediante un cambio de señal DTR), y otra para conmutar las señales de control DTR a DSR.

La interfaz está formada por los convertidores de nivel TTL-RS232, los cuales permiten a la red de conmutación comunicarse con el ambiente exterior donde se maneja el estandar RS232C.

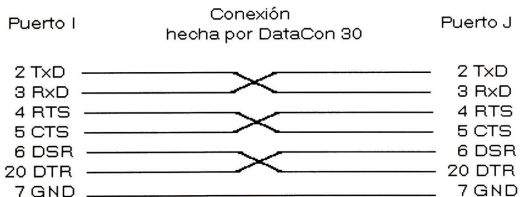
Normalmente con la interfaz RS232C se utilizan conectores DB-25 los cuales no están especificados por la norma. La siguiente tabla muestra la asignación de señales a las terminales del conector DB-25.

Terminal	Nombre	Función
1	PGND	Tierra de protección
2	TxD	Transmisión de datos

3	RxD	Recepción de datos
4	RTS	Solicitud de envío
5	CTS	Despejado para enviar
6	DSR	Modem preparado
7	GND	Tierra de señal
8	DTR	Terminal de datos lista

Un equipo de datos que utiliza la terminal 2 para transmitir se dice que está configurado como DTE (Data Terminal Equipment), si la usa para recibir se dice que está configurado como DCE (Data Circuit Equipment). Los puertos de DataCon 30 están configurados como DCE.

La siguiente figura muestra la forma en que DataCon 30 interconecta las distintas señales de los puertos.



- Memoria de comando

La memoria de comando del Datacon 30, controla las rutas de la red de conmutación por división en tiempo, esto es, se maneja el concepto de

conmutación por circuito (conmutación temporal), la manera de llevar a cabo este control, es mediante el cierre y apertura de las compuertas de 3 estados y de los flip-flops en la red de conmutación, para las señales TxD-RxD y RTS-CTS de la norma EIA RS232C. Es la memoria de comando la que permite establecer las trayectorias de comunicación en la red de conmutación en base a la dirección de las localidades de memoria, y su contenido, el cual es leído de manera cíclica. La importancia de escribir en la memoria de comando radica en que es a través de ello que se pueden establecer o romper conexiones, la memoria de comando del DataCon 30, está constituida básicamente por lo siguiente:

- Memoria de conexión / desconexión
- Generador de direcciones
- Comparador
- Interfaz de acceso
- Dispositivo de acceso

Como referencia del hardware ver las publicaciones de R. Rosado y M. Guzmán [45RG86], [24RG86], [25RG86] y [26RG86].

- Procesador de control

El procesador de control tiene varias funciones: conmuta las señales de control DTR y DSR, detecta las peticiones de conexión/desconexión de los dispositivos conectados a los puertos y dialoga con ellos para establecer y romper conexiones en la red de conmutación. El establecimiento de conexiones se hace cuando así se demanda en el diálogo. Cuando se peticiona una conexión, el procesador de control determina si el puerto requerido está libre y si tiene algún dispositivo conectado, de ser así, el procesador ordena una conexión, cuando el procesador detecta que uno de los dispositivos involucrados en un enlace, se desconecta del conmutador, es considerado como una petición de desconexión, y el procesador rompe el enlace indicando al dispositivo que permanece conectado que el enlace ha sido roto.

II.2 El núcleo de concurrencia

El software de servicios que se propone, permite manejar varias solicitudes de atención de manera simultánea. Para este caso, se podrían tomar como opciones de implantación algún lenguaje o sistema operativo de naturaleza

paralela, sin embargo, cuando no se dispone de estas herramientas, una solución puede ser el utilizar algún núcleo de concurrencia (o kernel), donde es posible la ejecución de varias tareas "al mismo tiempo" mediante la creación de procesos como entidades independientes. Así mediante un lenguaje de programación secuencial (en este caso el lenguaje C), es posible hacer uso de las funciones de dicho núcleo.

El núcleo de concurrencia utilizado fue hecho por el M.C. Andrés Vega García, profesor investigador de la Sección de Computación del CINVESTAV. Este núcleo hace cambio de contexto con la sucesión de algún evento, es decir, después de llamar a alguna primitiva que implique una posible modificación en la cola de procesos listos.

Se describirán a continuación las funciones de algunas de las primitivas importantes que incluye el kernel.

Primitiva:

executek

Función:

Inicia al núcleo y le transfiere el control.

Declaración:

```
executek(byte btime, byte nslots, byte maxprio, byte tmrprio, word tmrstk,  
          dword mems, void far *mema);
```

Primitiva:

create

Función:

Crea un proceso, un procedimiento base, o un procedimiento periódico

Declaración:

Creación de un proceso:

Desc pd;

```
pd = create(funcprocess, byte type=PROCESS, byte prio, 0,0, byte atrib,  
            word stksize, char *name, word npar, arg0, arg1, arg2);
```

Creación de un procedimiento base:

Desc pd;

```
pd = create(funbaseproc, byte type=BASEPROC, byte prio, 0, word arg,  
            byte atrib, 0, char *name, 0);
```

Creación de un procedimiento periódico:

Desc pd;

```
pd = create(funperioproc, byte type=PERIOPROC, 0, byte slot, word arg,  
            0, 0, char *name, 0);
```


Primitiva:

creates

Función:

Crea un semáforo s con un valor inicial val y un nombre name.

Declaración:

s = creates(word val, char *name);

Primitiva:

createt

Función:

Crea un temporizador t con una cuenta count (dada en número de tics de reloj), un procedimiento tmrproc que es ejecutado cuando expira el temporizador con un argumento arg y el atributo atrib, que puede ser PERIODIC, implicando que el timer se arranque automáticamente cada vez que expira.

Declaración:

Tmr t;

t = createt(word count, tmrproc, word arg, byte atrib);

Para mayor referencia, consultar el Manual de usuario MULTITASK (Vega García).

III. DESCRIPCION DE SERVICIOS

El software de comunicaciones desarrollado, busca la máxima funcionalidad, la cual tiene como objetivo dar flexibilidad y comodidad a los usuarios en la utilización de algún recurso al cual se puede tener acceso a través del conmutador, así como fomentar su mejor aprovechamiento y facilitar la tarea administrativa de los mismos. A continuación se describe cada comando para el manejo y organización de usuarios y puertos, presentando un ejemplo en cada caso.

Como el Sistema distingue a cada usuario mediante su registro federal de contribuyentes, se tomará en lo sucesivo y con fines de ejemplificación, el RFC= ROAG651214 como un identificador de usuario particular.

III.1 Usuario maestro

Este usuario tendrá una clave de acceso especial y es quien tiene acceso a todos los servicios y puertos del sistema, pudiendo modificar su configuración a voluntad. Entre sus funciones están:

- a) Dar de alta a nuevos usuarios asignándoles su clave de acceso.
- b) Definir o modificar los servicios a los cuales tendrá acceso determinado usuario.
- c) Definir o modificar los puertos a los cuales tendrá acceso determinado usuario.
- d) Asignar o modificar la jerarquía de cada usuario.
- e) Modificar los atributos de caracterización de puertos
- f) Definir o modificar los grupos de puertos

III.2 Altas usuarios

Este comando recibe como parámetro un RFC, que se insertará en la tabla de usuarios si se cumplen las dos condiciones siguientes:

- a) Hay espacio en la tabla de usuarios
- b) El RFC no se ha dado de alta anteriormente.

En caso contrario, se mandarán los mensajes de error correspondientes.

Formato:

SET US ROAG651214

En la manera en que está implantado, no existe orden de inserción, se hace en el primer espacio que se encuentra libre, de la tabla de usuarios.

III.3 Bajas usuarios

Será una facilidad del Usuario Maestro, el dar de baja a un usuario del sistema.

El comando para ello es el siguiente:

RST US ROAG651214

O bien

RST US ALL

Este último, dejará vacía la tabla de usuarios, con excepción del usuario maestro, siendo el único que existirá en todo momento.

Cada usuario que es dado de alta, se inicializará de la siguiente manera:

Sin clave de acceso

RFC igual al RFC del usuario

Ningún servicio permitido

Puertos permitidos = Puertos_Default

Jerarquía = 10

Grupo de puertos default = vacío

En donde Puertos_Default es un conjunto de puertos establecido por el usuario maestro, para la inicialización de todos los usuarios que se den de alta. El usuario maestro podrá invocar al comando de asignación de clave de acceso, después de la operación de alta.

Cuando ocurra una baja y el usuario esté conectado al sistema mediante varios puertos, todos ellos se desconectarán e inicializarán si se requiere.

III.4 Modificación de los atributos de un usuario

Es facilidad del usuario maestro, el poder cambiar los atributos de un usuario específico, tales como:

- servicios permitidos
- puertos permitidos
- Clave de acceso
- Asignación de Jerarquía

Los comandos para éstas acciones se describen a continuación.

III.4.1 Otorgamiento y restricción de acceso a servicios

Para que un usuario goce de algún servicio otorgado por el sistema, debe existir una autorización por parte del Usuario Maestro, de modo que, mediante algunos comandos ejecutados por éste, se concederá o prohibirá a algún usuario específico, el empleo de los siguientes servicios:

Servicios:	Conexión inmediata
	Cola de espera
	Desalojo de usuarios de baja prioridad
	Conexión y desconexión externa.

El formato general para la autorización o negación al acceso a un servicio, está dado por la palabra SET seguida del nombre del servicio del cual se trata, escribiendo a continuación el RFC del usuario y el numeral 1 en caso de autorización y 0 en caso de prohibición. En base a esto, el sistema tendrá conocimiento, mediante la máscara de acceso a servicios asignada a cada uno de los usuarios, de todas las posibles facilidades de que éstos pueden gozar.

Descripción y especificación de servicios

III.4.1.1 Conexión inmediata

El usuario maestro otorgará a algún usuario particular, el acceso al servicio de conexión inmediata bajo el siguiente comando:

SET CI ROAG651214 1

El servicio de conexión inmediata, es la capacidad de especificar un puerto o un rango de puertos (grupo default de puertos para cada uno de los usuarios), con los que el sistema tratará de establecer una conexión, como primera acción en cada sesión con un usuario que goce de este servicio. Eligiendo entre ellos al primer puerto que encuentre disponible, y conectandolo una vez que el usuario haya dado la confirmación de conexión, sin necesidad de solicitarlo explícitamente al sistema. Esto es favorable cuando generalmente se trabaja con un puerto o grupo de puertos específico .

Cuando un usuario tiene posibilidad de acceder este servicio, podrá hacer uso de él o no, según le convenga. Si lo deshabilita no implica que dicho servicio quede restringido definitivamente, simplemente al empezar una sesión, el sistema no tratará de establecer una conexión automáticamente, solamente hasta que el usuario decida su nueva habilitación.

El comando para habilitar o deshabilitar el servicio es el siguiente:

SET CI 1

III.4.1.2 Cola de espera

Será una facilidad del usuario maestro, el establecer qué usuarios podrán acceder el servicio de cola de espera, bajo el siguiente comando:

SET QE ROAG651214 1

Cuando un usuario intente hacer una enlace con un puerto o rango de puertos y estos estén ocupados o no encendidos, el sistema indicará al usuario peticionador el número de usuarios en espera de usar tales puertos. Después el sistema quedará en espera de que el usuario le confirme o no su inclusión en la cola de espera. A continuación el sistema esperará por un nuevo comando o su indicación de fin de sesión. Cuando llega el turno al usuario en cuestión, de usar cierto dispositivo por el cual esperaba, el sistema establecerá el enlace. Este servicio resulta muy apropiado en aquellos casos en los que la demanda de unos cuantos dispositivos es muy alta, dejando al sistema la carga de dar turno a los usuarios para la utilización de cierto dispositivo.

III.4.1.3 Desalojo de usuarios de baja prioridad

El Usuario Maestro permitirá a un usuario específico, el acceso al servicio "desalojo de usuarios de baja prioridad" mediante el siguiente comando:

SET DU ROAG651214 1

Cuando un usuario requiere de cierto dispositivo (un puerto en general), y éste se encuentra ocupado, el sistema se lo indicará, y si su prioridad es superior a la del usuario que actualmente utiliza el puerto o dispositivo, el sistema le dará opción de decidir desalojarlo o abortar su petición, quedando en espera de un nuevo comando. Si el usuario decide hacer el desalojo, el afectado será avisado de que su enlace ha sido forzado a terminar, por un usuario de mayor prioridad.

III.4.1.4 Conexión y desconexión externa

El otorgamiento de este servicio por parte del Usuario Maestro, se efectúa al ejecutar el comando:

SET CDE ROAG651214 1

Cuando un usuario puede acceder este servicio, tendrá la posibilidad de establecer conexiones de puertos distintos al que ocupa, siempre y cuando cumplan con las siguientes características:

- Estén desconectados
- Estén habilitados o bien sean puertos de los que no importa su habilitación

Si ocurre la conexión y los puertos son de tipo activo, se les enviará un mensaje, anunciándoles su conexión.

Un usuario puede realizar una conexión externa si se le ha otorgado éste servicio por el usuario maestro y empleando el siguiente comando:

CNXE Pto1,Pto2

Si se trata de una desconexión externa, se podrá deshacer un enlace de puertos distintos al que ocupa el usuario, mediante el siguiente comando:

DNXE Pto1

Así, el Pto1 quedará desconectado junto con el que estaba enlazado.

Si los puertos son de tipo activo, se les enviará un mensaje anunciándoles su desconexión.

Otra posible opción para modificar el acceso a servicios es el siguiente:

Cada servicio tiene un número asociado:

CI	0	Conexión inmediata
QE	1	Cola de espera
DU	2	Desalojo de usuarios
CDE	3	Conexión y desconexión externa

por lo que el Usuario Maestro podrá conceder a algún usuario particular, el acceso a ciertos servicios, indicando su número o rango de números, mediante los siguientes comandos posibles:

Indicando los números de servicio separados por comas

SET SU ROAG651214 0,1,2,3

Indicando un rango de servicios

SET SU ROAG651214 0/2

Indicando todos los servicios

SET SU ROAG651214 ALL

el último comando permite el acceso de todos los servicios del sistema.

En lo que concierne a la restricción, el Usuario Maestro podrá hacer uso de los siguientes comandos, respectivamente:

SET CI ROAG651214 0
SET RE ROAG651214 0
SET DU ROAG651214 0
SET CDE ROAG651214 0

O bien, mediante su número

Indicando los números de servicio separados por comas

RST SU ROAG651214 0,1,2,3

Indicando un rango de servicios

RST SU ROAG651214 0/2

Indicando todos los servicios

RST SU ROAG651214 ALL

el último comando restringe el acceso a todos los servicios.

Cuando un usuario solicite determinado servicio, el sistema checará contra la máscara de acceso a servicios como primera acción, para proceder o denegar su acceso.

III.4.2 Otorgamiento y restricción de acceso a puertos

Será una facilidad del usuario maestro y le permitirá establecer a qué recurso tendrá acceso cada usuario. Esto lo hará mediante un comando especial el cual permitirá construir la nueva máscara de acceso a puertos para uno o más usuarios. Cuando se solicite el enlace a un dispositivo, el sistema checará contra la máscara de acceso a puertos como primera acción, para proceder o denegar el acceso al puerto solicitado.

El comando para la autorización de acceso a puertos (SET PU), o bien para la prohibición (RST PU), deberá estar seguido del RFC de quien se trate y la lista de puertos permitidos o negados respectivamente. En base a esto, el sistema tendrá conocimiento, mediante la máscara de acceso a puertos asignada a cada uno de los usuarios, de todas las posibles conexiones que éstos pueden solicitar.

Para autorización de acceso

SET PU ROAG651214 5,8,10/13,20

O bien

SET PU ROAG651214 ALL

Este último permitirá el acceso a todos los puertos

Para Restricción de acceso:

RST PU ROAG651214 5,8,10/13,20

O bien

RST PU ROAG651214 ALL

Este último restringirá el acceso a todos los puertos.

III.4.3 Clave de acceso

Cada usuario tendrá una clave de acceso al sistema, la cual consistirá de una secuencia de cinco caracteres alfanuméricos. Su función será la de llevar un control sobre las personas que puedan acceder el sistema, siendo restringidas mediante esta clave, personas no autorizadas.

Así mismo, los usuarios podrán, una vez que han entrado al sistema, modificar a voluntad la clave utilizada para tal efecto. De esta manera, el usuario podrá cambiar con cierta periodicidad su clave de acceso al sistema, evitando así que otros usuarios puedan hacer uso sin su autorización, de las facilidades de las que goza. El comando para asignar una nueva clave es el siguiente:

-En caso de asignar por primera vez una clave de acceso.

SET PW CLAVE

-En caso de contar con una clave y desear cambiarla

SET PW C-OLD C-NEW

El Usuario Maestro tendrá la facilidad de cambiar y de disponer la clave de acceso de todos los usuarios mediante el siguiente comando:

SET PW ROAG651214 CLAVE

III.4.4 Asignación de jerarquía

El usuario maestro exclusivamente establecerá una jerarquía entre los usuarios del sistema, lo que será una medida de la prioridad que deba darse a cada uno de ellos y será además una referencia para el sistema cuando se tomen decisiones al momento de dar servicios, tales como "Prioridad de atención" y "Desalojo de usuarios de baja prioridad". La asignación de prioridades juega un papel importante en sistemas donde diferentes usuarios

deben compartir recursos, ya que es el indicador de la importancia que dará el sistema a determinado usuario.

El comando empleado para tal efecto es:

SET JR ROAG651214 8

en donde la jerarquía de mayor prioridad es la cero y quien la posee unicamente es el usuario maestro.

III.5 Especificación de grupos de puertos

Este servicio permite al usuario especificar un nombre simbólico asociado a un puerto o rango de puertos. Esta clase de nombres comenzará con un caracter específico propio del sistema y podrán tener hasta un máximo de hasta 3 caracteres; un usuario común podrá hacer una especificación de nombre abreviado llamada "default", en cambio el usuario maestro, podrá definir hasta 10 grupos de puertos, a los cuales tendrán acceso todos y cada uno de los usuario del sistema. Para eliminar estos nombres se utilizará el comando "RST" apropiado para la eliminación de grupos de puertos.

El comando para la especificación, es el siguiente:

SET GP VAX 10
SET GP UTL 20/27
SET GP PRN 12,18

Si se quiere eliminar uno de estos grupos, el usuario maestro podrá emplear los siguientes comandos posibles:

RST GP VAX
RST GP ALL

el último comando, dejará vacía la lista de Grupos.

Existe también, para cada usuario, un grupo default que podrá especificar con alguno de los siguientes comandos:

SET GP 10
SET GP 20/27
SET GP 12,18

Este grupo default no tiene nombre asignado.

Si se quiere poner en blanco dicho grupo, el comando a efectuar será:

RST GP

El establecer grupos de puertos con un nombre asociado, resulta más familiar al usuario, que un simple número de puerto.

III.6 Modificación de los atributos de un puerto

Como el número de puertos en el conmutador, es constante, no se hablará de altas o bajas, únicamente se podrán modificar los atributos de los puertos ligados al sistema.

La tarea del usuario maestro, será de inicializar cada vez que se requiera, las características fundamentales de un puerto, que son necesarias para la correcta operación del sistema, como la velocidad de transmisión, la paridad, si se trata de un puerto de tipo activo (comunicación bilateral), o pasivo (comunicación unilateral), si se le da eco o si es un puerto utilizado para conexiones permanentes (no importa el estado de la señal DTR).

Los comandos que deberá emplear el usuario maestro para éstos fines serán:

SET PT X TYP:1
SET PT X TYP:0

Para asignar al puerto número X el tipo activo (1) o pasivo(0)

SET PT X REQ: 1
SET PT X REQ: 0

Con este comando, se considera al puerto número X, como un puerto al cual no se le dará jamás atención(0) , aunque la señal DTR cambie su estado, o bien como un puerto que se formará en la cola de solicitudes de atención(1).

SET PT X ECH:1
SET PT X ECH:0

Para determinar si se le hará(1) o no(0) eco a algún dispositivo X

SET PT X SPD:9600

Asignación de velocidad de transmisión. Unicamente 8 son las que están permitidas: 150, 300, 600, 1200, 2400, 4800, 9600, 19200.

SET PT X PAR:0

Asignación del tipo de paridad a un puerto X. Tipo 0: paridad = par, tipo 1: paridad = impar y tipo 2: sin paridad.

Algunos de estos comandos, pueden estipularse en uno sólo, apareciendo en cualquier orden, como se muestra a continuación:

SET PT X PAR:1 ECH:1 TYP:1 SPD:600 REQ:1

III.7 Comunicación punto o multipunto

Esta facilidad habilita al usuario para que pueda enviar mensajes simultáneamente a diferentes usuarios. El sistema realiza esta operación mediante el rompimiento de los enlaces de aquellos usuarios para quienes va dirigido el mensaje, reestableciendo sus previos enlaces al término del mensaje enviado. El usuario con esta facilidad indicará al sistema mediante un comando, su petición de establecer una conexión unidireccional punto multipunto, y el sistema le preguntará primeramente los usuarios destino, en seguida quedará en un modo de edición lo cual terminará dando un caracter especial al editor. En ese momento el usuario dará por satisfecha su petición, y el sistema se encargará de enviar los mensajes a los usuarios destino, agrupándolos por la velocidad a la que se les enviará el mensaje.

III.8 Correo electrónico

Este servicio permite que un usuario pueda enviar y recibir textos a/y de otros usuarios. El usuario con este servicio, dispondrá de cierta cantidad de memoria en la cual podrá ser depositado su correo. La edición y envío se realizará

mediante un determinado comando, al cual el sistema responderá con las listas de usuarios que tienen el servicio de correo electrónico, procediendo como sigue:

- a) El sistema preguntará por la identificación del destinatario.
- b) Si la selección del destinatario es válida, el usuario entrará a un modo de edición, el cual se dará por terminado cuando así lo requiera el usuario mediante un caracter específico dado al editor.
- c) Si durante la edición se detecta que el área de memoria disponible para el destinatario se ha agotado, se le indicará al usuario quien podrá decidir truncar su mensaje o abortarlo.

Al momento que el usuario entra al modo de edición, tendrá a su disposición un subconjunto de los comando disponibles en el procesador de palabras "Word Star" para facilitar la edición del mensaje. La existencia de mensajes se le hará saber a un usuario al momento en que entra al sistema o cuando dialoga con el mismo, sin que tal diálogo sea perturbado por la indicación que sea hecha por el sistema. El usuario podrá mediante un comando, leer sus mensajes y/o borrarlos o bien mandarlos a impresión.

III.9 Deshabilitación de la interrupción del enlace por otros usuarios

Un determinado usuario podrá deshabilitar la entrada de interrupciones de otros servicios, (cosa que suele presentarse cuando algún usuario hace uso de los servicios de "Comunicación punto Multipunto"). Esto tiene como finalidad que el usuario no sea interrumpido cuando sea importante que su enlace no sea roto para el buen término del trabajo que realiza.

III.10 Deshabilitación del eco

En el caso de terminales que realizan el ECO de manera automática, el usuario normal (o el usuario maestro), tendrá la capacidad de cancelar el ECO a una determinada terminal, dando esto la facilidad de no requerir reconfigurar la terminal cuando fuera posible.

III.11 Enlace permanente

Este servicio consiste en que dos dispositivos y/o terminales son permanentemente enlazados por el sistema sin importar que los dispositivos

estén o no conectados físicamente al sistema. Este servicio es de gran utilidad cuando cierto usuario debe tener de manera dedicada cierto equipo o cuando realiza algún trabajo especial con el mismo en el que la conexión debe permanecer aún cuando se apague la terminal del usuario.

III.12 Estadísticas

Este servicio será de uso exclusivo para el usuario maestro y las estadísticas levantadas por el sistema serán:

- Por usuario:

- * Número de accesos al sistema por día.
- * Promedio de tiempo entre entradas consecutivas al sistema
- * Máximo tiempo de espera por la atención del sistema desde que enciende su terminal

- Por dispositivo:

- * Tiempo de uso por día
- * Número de usuarios atendidos por día
- * Hora pico

- Por servicio de Correo Electrónico:

- * Tiempo de uso por día
- * Tiempo promedio de edición
- * Número de bytes promedio por mensaje

- Por servicio de Spooler:

- * Numero de archivos enviados por día
- * Tiempo máximo entre el envío del archivo y su total impresión.

Las estadísticas tienen por objeto el levantar información que pueda revelar aspectos en los cuales el sistema da un servicio deficiente, tal como el tiempo de espera de un usuario para que sea atendido. Con ésta información es posible decidir cuáles son los aspectos en los que el sistema se debe depurar o modificar, de modo que pueda dar un mejor servicio.

III.13 Estado del conmutador

Presenta al usuario un menú a cinco columnas del estado de todos los puertos del conmutador. En las primeras dos columnas, se dará el número de puerto y el nombre del usuario o dispositivo conectado. En la tercera y cuarta columna, estará el número de puerto y el nombre del usuario o dispositivo conectado con el cual está enlazado el primero cuando este sea el caso, y en la última el número de usuarios en espera de usar el dispositivo especificado en la tercera y cuarta columna.

III.14 Menú de dispositivos libres

El usuario podrá solicitar un menú con los dispositivos libres a los cuales podrá tener acceso. Esta información le será presentada a dos columnas, en la primera tendrá el número de puerto y en la segunda el nombre del dispositivo conectado. Tiene como finalidad recordar al usuario con qué dispositivo se puede enlazar y así evitar rechazos del sistema, agilizando también su diálogo con el mismo.

III.15 Menú de servicios

El usuario podrá solicitar al sistema un menú con los servicios a los que tiene acceso. Así, el usuario podrá recordar los servicios a los cuales tiene derecho y podrá evitar rechazos del sistema, agilizando también su diálogo con el mismo.

III.16 Temporización del uso de puertos (dispositivos)

A ciertos usuarios se les podrá restringir el uso de cierto dispositivo o la terminal misma a un determinado lapso de tiempo. Esto resulta muy útil en casos donde un equipo es muy frecuentemente usado, por ejemplo una impresora; y evitar que un usuario pueda acapararla impidiéndole el uso de la misma a otros usuarios, o bien, para obligar a ciertos usuarios a usar cierto recurso lo más rápidamente posible. Cuando el usuario solicite su conexión a cierto dispositivo y éste se encuentre libre, el sistema le indicará que será eliminado su enlace con tal dispositivo al término de un tiempo previamente especificado por el usuario maestro, de modo que el usuario tome sus precauciones.

III.17 Temporización del uso del sistema

Consiste en la restricción en cuanto al lapso de tiempo, en que un usuario puede conversar con el sistema para lograr sus objetivos (tiempo que dura una sesión).

III.18 Resumen de la especificación sintáctica de comandos

<RESTRICC_PORTS>::=

<ASIGNA_PORTS> | <NIEGA_PORTS>

<ASIGNA_PORTS>::=

**SET PU <RFC> <PUERTOS> |
SET PU <RFC> ALL**

<NIEGA_PORTS>::=

**RST PU <RFC> <PUERTOS> |
RST PU <RFC> ALL**

<PUERTOS>::=

**<PORT> |
<PORT> <INTERVALO> <PORT> |
<PORT> <SEPARADOR> <PUERTOS> |
<PORT> <INTERVALO> <PORT> <SEPARADOR> <PUERTOS>**

<GRUPO_DEFAULT>::=

<ASIGNA_PTO_DEFAULT> | <LIMPIA_PTO_DEFAULT>

<ASIGNA_PTO_DEFAULT>::=

**SET DEF_PORT <PUERTOS>
SET DEF_PORT ALL**

<LIMPIA_PTO_DEFAULT>::=

RST DEF_PORT <PUERTOS>

RST DEF_PORT ALL

<ASIGN_PASSWORD>::=

SET PW <RFC> <IDENT>

<PERMITE_COLA_ESP>::=

SET QE <RFC> 0 | SET QE <RFC> 1

<PERMITE_CNXION_INMED>::=

SET CI <RFC> 0 | SET CI <RFC> 1

<ACTIVA_CNXION_INMED>::=

SET CI 0 | SET CI 1

<ASIGNA_JERARQ>::=

SET JR <RFC> <ENTERO>

<PERMITE_DESALOJO_US>::=

SET DU <RFC> 0 | SET DU <RFC> 1

<RESTRICC_SERVS>::=

<ASIGNA_SERVS> | <NIEGA_SERVS>

<ASIGNA_SERVS>::=

SET SU <RFC> <SERVICIOS> | SET SU <RFC> ALL

<NIEGA_SERVS>::=

RST SU <RFC> <SERVICIOS> | **RST SU** <RFC> **ALL**

<SERVICIOS>::=

<NUM_SERV> |
 <NUM_SERV> <INTERVALO> <NUM_SERV> |
 <NUM_SERV> <SEPARADOR> <SERVICIOS> |
 <NUM_SERV> <INTERVALO> <NUM_SERV> <SEPARADOR>

<SE

RVICIOS>

<ADMON_USUARIOS>::=

<ALTAS_USUARIOS> | <BAJAS_USUARIOS>

<ALTAS_USUARIOS>::=

SET US <USUARIOS>

<USUARIOS>::=

<RFC> | <RFC> <SEPARADOR> <USUARIOS>

<BAJAS_USUARIOS>::=

RST US <USUARIOS> | **RST US** **ALL**

<ASIGN_ATRIB_PORTS>::=

SET PT PORT_i <ATRIB_PORT>

<ATRIB_PORT>::=

<P_ATRIB> | <P_ATRIB> <ATRIB_PORT>

<P_ATRIB>::=

TYP <ASIGNACION> **0** | **TYP** <ASIGNACION> **1** |
REQ <ASIGNACION> **0** | **REQ** <ASIGNACION> **1** |
ECH <ASIGNACION> **0** | **ECH** <ASIGNACION> **1** |
SPD <ASIGNACION> <ENTERO> |
PAR <ASIGNACION> <ENTERO>

<GRUPOS_PORTS>::=

<ALTA_GPO_PORT> | <BAJA_GPO_PORT>

<ALTA_GPO_PORT>::=

SET GP <IDENT> <PORT> |
SET GP <IDENT> <PORT> <SEPARADOR> <PORT> v |
SET GP <IDENT> <PORT> <INTERVALO> <PORT>

<BAJA_GPO_PORT>::=

RST GP <IDENT> | **RST GP ALL**

<GRUPO_DEFAULT>::=

<DEF_GPO_DEFAULT> | <LIMPIA_GPO_DEFAULT>

<DEF_GPO_DEFAULT>::=

SET GP <PORT> |
SET GP <PORT> <SEPARADOR> <PORT> |
SET GP <PORT> <INTERVALO> <PORT>

<LIMPIA_GPO_DEFAULT>::=

RST GP

<CORREO_ELEC>::=

SET CE <RFC> <MSGE>

<ASIGN_ATRIB_USUARIOS>::=

SET AU <ATRIB_USUAR>

<ATRIB_USUAR>::=

<US_ATRIB> | <US_ATRIB> <SEPARADOR> <ATRIB_USUAR>

<US_ATRIB>::=

QE <ASIGNACION> **0** | **TYP** <ASIGNACION> **1** |
CI <ASIGNACION> **0** | **REQ** <ASIGNACION> **1** |
DU <ASIGNACION> **0** | **ECH** <ASIGNACION> **1** |
JR <ASIGNACION> <ENTERO> |
PU <ASIGNACION> <PUERTOS> |
SU <ASIGNACION> <SERVICIOS>

<MODIF_TIMERS>::=

SET TM <TIMERS>

<TIMERS>::=

<NUM_TIMER> <ASIGNACION> <ENTERO> |
 <NUM_TIMER> <ASIGNACION> <ENTERO> <SEPARADOR>

MERS>

<TI

<LISTA_CLAVE_ACCESO>::=

LST PW <USUARIOS> | LST PW ALL

<LISTA_JERARQ_USUARIOS>::=

LST JR <USUARIOS> | LST JR ALL

<LISTA_JERARQ_PROPIA>::=

LST JR

<LISTA_PUERTOS_PERMITIDOS_USUARIO>::=

**LST PU |
LST PU <USUARIOS> |
LST PU ALL**

<LISTA_USUARIOS>::=

LST US ALL | LST US <USUARIOS>

<LISTA_PORTS_DEFAULT>::=

LST DEF_PORT

<LISTA_TIMERS>::=

LST TM ALL | LST TM <L_TIMERS>

<L_TIMERS>::=

**<NUM_TIMER> |
<NUM_TIMER> <SEPARADOR> <L_TIMERS>**

<LISTA_MENU_SERVICIOS>::=

LST SU ALL | LST SU <USUARIOS>

<LISTA_DISPOSITIVOS_LIBRES>::=

LST DF

<LISTA_PUERTOS>::=

LST PT ALL | LST PT <PUERTOS>

<LISTA_GRUPOS_PUERTOS>::=

**LST GP |
LST GP ALL |
LST GP <L_GPOS>**

<L_GPOS>::=

**<IDENT> |
<IDENT> <SEPARADOR> <L_GPOS>**

<LISTA_ESTADISTICAS>::=

LST STAD

<LISTA_EDO_CONMUTADOR>::=

LST CONM

IV. ELEMENTOS BASICOS DEL SOFTWARE

En este capítulo se introducirán las estructuras de datos y constantes primordiales que conforman al sistema, así como las funciones importantes para el establecimiento de conexiones y desconexiones, ayudándose de diversos diagramas LED, además se explicarán las funciones básicas de inicialización y rutinas en general.

IV.1 Estructuras de datos

CONSTANTES IMPORTANTES DEFINIDAS:

MAXPORT	32	Número máximo de dispositivos
NOCONNECT	255	Valor con que se denota la desconexión
NTJTAS	4	El número de tarjetas con las que se direccionan los 32 puertos.
NUMUSART	4	Número de usarts para atención de solicitudes
TAMBUFF	360	Tamaño de Colas de Espera de E/S
LBUFF	80	Longitud de buffers auxiliares
TAMQPORT	4	Tamaño de colas de espera a puertos
NUMUSUAR	15	tamaño de la tabla de usuarios
MAXJER	255	Jerarquía máxima asignada
MAXSERV	8	Número máximo de servicios
INPUT	0	Indica que se hace referencia a un evento de DTR
OUTPUT	1	Indica que se hace referencia a un evento de DSR
HHELP	250	Constante que sirve como caracter de control, para detectar cuándo se ha hecho una llamada al módulo de ayuda, insertándola al final de una cadena de caracteres

Las siguientes constantes indican la aparición de algún evento, mediante el número de bit, de derecha a izquierda dentro del byte para máscara de eventos de un usart .

BOK	0	Envío del último mensaje
BTEMPOR	1	Hubo time out por no recibir respuesta
BSESION	2	Hubo time out por fin de sesion
BDTR	3	El puerto que se atiende se ha deshabilitado

BQE	4	Se ha desocupado un puerto por el cual espera el usuario al que se atiende
-----	---	--

Como máximo deberán ser 8 los eventos de control para el usart, actualmente se han definido solamente 5.

Las siguientes constantes representan cada uno de los servicios que ofrece el sistema, mediante el número de bit, de derecha a izquierda dentro del byte para máscara de acceso a servicios de un usuario .

COLAESP	0	Cola de espera (QE)
DESALOJO	1	Desalojo de usuarios de baja prioridad (DU)
CNXINM	2	Conexión inmediata (CI)
CNXACTV	3	Conexión inmediata activada (CXA)
CX_DX_EXT	4	Conexión y desconexión externa (CDEX)

A lo más 8 servicios distintos podrán ser manejados por esta máscara. Actualmente, se han definido los cinco ya mencionados.

NUMGPOS	15	Tamaño de la tabla de grupos de puertos
NUMVEL	8	número máximo de velocidades de transmisión

VARIABLES Y ESTRUCTURAS DE DATOS:

int TRESP	=	500	Tiempo máximo para esperar por lo menos un caracter de entrada.
int TSESION	=	2600	Tiempo máximo en que un usuario puede interactuar con un usart

Con la siguiente tabla se establece una correspondencia entre el número de velocidad (de 1 a 8) y la velocidad misma.

```
char *L_Veloc[NUMVEL] = {"150", "300", "600", "1200", "2400", "4800", "9600",
                          "19200"};
```

Arreglo que simula la matriz de conmutación:

```
byte matriz[MAXPORT]; /* MATRIZ DE CONMUTACION, 0 - 31 */
```

Las siguientes estructuras son útiles para la detección de algún cambio en la señal DTR de los puertos, es decir para reconocer qué puertos requieren atención.

```
byte tjta[2][NTJTAS]; /* Señales DTR y DSR de cada puerto*/
byte DTR[2][MAXPORT]; /* Para guardar DTR actual y DTR anterior */
byte new=0; /* Variable que conmuta entre 0 y 1 para detectar
cambios en el arreglo DTR, de tal modo que si i es
un puerto y DTR[new][i] es distinta a
DTR[!new][i], se interpreta como una solicitud de
atención del puerto i, después new cambiará a
!new, desechando la lectura vieja de DTR's
para almacenar en su lugar una lectura nueva */
```

Estructura que constituye la lista de usuarios que accederán el sistema:

```
struct{
    char password[6];          Clave de acceso
    char RFC[11];              RFC del usuario
    byte ptomsc[NTJTAS];       Máscara de acceso a puertos
    byte srvmsc;                Máscara de acceso a servicios, según el bit:
                                bit 0 : QE
                                bit 1 : DU
                                bit 2 : CI
                                bit 3 : CXA
                                bit 4 : CDEX
    byte jerarqu;               Jerarquía en el sistema
    byte dfault[2];             Grupo de puertos default
} usuarios[NUMUSUAR];
```

Estructura fundamental de un USART:

```
typedef struct{
```

Software de servicios

Elementos básicos del software

<code>Sem sinp,inok;</code>	Para el control de inserción o eliminación de caracteres en el buffer de entrada
<code>Sem sout,sok;</code>	Para el control de inserción o eliminación de caracteres en el buffer de salida
<code>byte eventos;</code>	Máscara de eventos, entre ellos: bit 0: Envío del último caracter en una sesión bit 1: Hubo time out por no recibir respuesta bit 2: Hubo time out por fin de sesion bit 3: El puerto que se atiende se ha deshabilitado bit 4: Se a desocupado un puerto por el cual espera el usuario al que se atiende.
<code>byte pqe;</code>	Guarda el puerto que ha quedado libre y se esperaba por él.
<code>Buffer bout,bin;</code>	Buffers circulares de entrada y salida
<code>Tmr tempor,tsesion;</code>	Temporizadores de respuesta y sesión
<code>char *str;</code>	Para almacenar un string obtenido del buffer de entrada y procesarlo posteriormente.
<code>} complex;</code>	
<code>complex tty[NUMUSART];</code>	Lista de USART's que atienden las solicitudes

Una vez inicializada la siguiente estructura por el usuario maestro, todos los usuarios que se den de alta posteriormente, tendrán atribuida esta máscara de acceso a puertos, por default.

```
byte ini_ports[NTJTAS]; /*DEFAULT de mascara de acceso a puertos al
dar de alta un usuario */
```

Estructura fundamental para la definición de puertos

<code>struct {</code>	
<code>byte activo;</code>	Está en uno cuando un puerto permite el diálogo con un usart y en cero en otro caso
<code>byte atencion;</code>	Está en uno cuando si se toman en cuenta las variaciones del DTR de un puerto como señal de solicitud y en cero en otro caso
<code>int usuario;</code>	Campo destinado al usuario que ocupa el puerto, guardando su posición en la tabla de usuarios

byte paridad;	Tipo de paridad de un puerto, codificada bajo los siguientes números: 0 : paridad par 1 : paridad impar 2 : sin paridad
byte eco;	Está en uno cuando el puerto tiene eco, cero en otro caso
byte veloc;	Actualmente este byte permite indicar hasta 8 velocidades de transmisión distintas, asociadas con un número entero del 0 al 7, estas son: 0: 150 1: 300 2: 600 3: 1200 4: 2400 5: 4800 6: 9600 7: 19200
byte queue[TAMQPORT];	Cola de Espera del puerto
} Lst_port[MAXPORT];	Lista de 32 puertos

Estructura para la definición de grupos de puertos por el usuario maestro

```

struct {
    char nomgpo[6];      Nombre del grupo
    byte puertos[2];     Puertos o rango de puertos que comprende,
                        almacenando uno, dos o un rango de puertos
} Grupo[NUMGPOS];      Lista de grupos

```

Estructuras requeridas para la modificación de fecha y hora

```

struct time *tmpo;
struct date *hoy;

```

Semáforo con el que se controla la inserción de solicitudes en el buffer circular b_petic:

Sem peticion; Indicador de alguna solicitud en el buffer b_petic

Buffer b_petic; Buffer circular en el que se almacenan las solicitudes de puertos

int cont[NUMUSART] = {0,0,0,0}; Controladores del cursor en las ventanas

Procesos en el Sistema:

Desc evencom; Proceso para la simulación de interrupciones por llegada de caracteres

Desc per_proc; Proceso periódico que se encarga de:
 - Vaciar las colas de salida de los usart's
 - Detectar cuando ha ocurrido una variación de DTR e insertar el puerto en el buffer de solicitudes (b_petic)

Desc usart[NUMUSART]; Procesos USART's

La tabla de palabras reservadas está especificada por la siguiente estructura, si se desea cambiar el nombre de algun comando, o añadir nuevos nombres, únicamente bastará con modificar esta tabla y la constante que indica su tamaño, si es necesario.

LPAL_RSV 40 /*Tamaño de la tabla de palabras reservadas*/

```
char *Plb_Rsrv[LPAL_RSV] = {
    "ON","OFF","SET","RST","LST","DATE","TIME",
    "CNXE","DNXE","CNC","SIM","PW","QE","CI",
    "JR","DU","PU","SU","US","PT","GP","CE",
    "AU","TM","ALL","T","R","V","P","E","DEFAULT",
    "CFG","MS","DF","PT","STAD","CONNM","DEF_PORT",
```

```
"N","S");
```

IV.2 Funciones básicas

IV.2.1 Funciones de inicialización

Las funciones que a continuación se presentan son esenciales para el buen funcionamiento del sistema:

Inicialización de la matriz de conmutación, partiendo con todos los puertos desconectados.

```
void inicia_matriz() /* inician todos los puertos como desconectados */
{
    int i;

    for(i=0;i<MAXPORT;i++)
        matriz[i] = NOCONECT;
}
```

Inicialización de las señales DTR y DSR en la tarjeta de puertos. Se inicializa también el arreglo para la detección de cambios en DTR's.

```
void iniciatjtas() /* pone en cero a DTR(INPUT) y a DSR(OUTPUT) de cada
                    puerto */
{
    int i;

    for(i=0;i<NTJTAS;i++)
        tjta[INPUT][i] = tjta[OUTPUT][i] = 0;
    for(i=0;i<MAXPORT;i++)
    {
        DTR[new][i] = 0;
        DTR[!new][i] = 0;
    }
}
```

Creación del buffer de solicitudes de atención, y de los buffers circulares de entrada y salida para cada Usart, así como su inicialización.

```
void crea_buff()
{ int i;
```

```

b_petic=mkbuff(TAMBUFF);
clrbuff(b_petic);
for(i=0;i<NUMUSART;i++)
{
    tty[i].str = espace[i];
    tty[i].bin = mkbuff(TAMBUFF);
    tty[i].bout = mkbuff(TAMBUFF);
    clrbuff(tty[i].bin);
    clrbuff(tty[i].bout);
}
}

```

Creación de los semáforos para el control de las operaciones de entrada y salida, inicializándolos adecuadamente, así como el semáforo que indica la aparición de una solicitud.

```

void Crea_Semaforos()
{
    int i;
    peticion = creates(0,"peticion-port");
    for (i=0;i<NUMUSART;i++)
    {
        tty[i].sout = creates(TAMBUFF,"Sem-Sal");
        tty[i].inok = creates(TAMBUFF,"Sin-U0");
        tty[i].sinp = creates(0,"Sem-Ui0");
        tty[i].sok = creates(0,"Buff_Out");
    }
}

```

Descripción y consideraciones de inicialización:

- sout: Indica espacio disponible en el buffer de salida
Al inicio su valor es el tamaño del buffer
- inok: Indica espacio disponible en el buffer de entrada
Al inicio su valor es el tamaño del buffer
- sinp: Indica si hay un elemento en el buffer de entrada
Al inicio su valor es cero

sok: Indica que se ha transmitido el último carácter a algún dispositivo
Al inicio su valor es cero

peticion: Indica cuándo hay una solicitud de atención en el buffer de solicitudes.
Su valor al inicio es cero

Inserción de un usuario en el primer espacio disponible de la tabla de usuarios, asignándole el acceso a los puertos defaults, establecidos por el usuario maestro.

```
void ins_usuario(char *rfc,char *pasw)
{ int i=0,j,exito = 0;

while ((i<NUMUSUAR) && (!exito) )
{
    if (*usuarios[i].RFC == '\0')
    {
        strncat(usuarios[i].RFC,rfc,10);
        *usuarios[i].password = '\0';
        if (*pasw != '\0') strncat(usuarios[i].password,pasw,5);
        exito = 1;
    }
    i++;
}
}
```

Inicialización de la tabla de usuarios, insertando de antemano el que va a distinguirse como usuario maestro (posición cero), concediéndole la mayor prioridad (jerarquía = 0) y acceso a todos los puertos y servicios del sistema.

```
void inicia_usuarios()
{ int i,j;

for(i=0;i<NUMUSUAR;++i)
{
    *usuarios[i].RFC = '\0';
    *usuarios[i].password = '\0';
    for (j = 0; j < 4;j++)
```



```

usuarios[i].ptomsc[j]=0;
usuarios[i].srvmsc = 0;
usuarios[i].jerarq = 10;
usuarios[i].dfault[0] = NOCONECT;
usuarios[i].dfault[1] = NOCONECT;
}
ins_usuario("ROAG651214","");
usuarios[0].ptomsc[0]= 240; /* Los cuatro puertos primeros son para USART*/
for (j = 1;j < NUMUSART;j++)
    usuarios[0].ptomsc[j]= 255;
usuarios[0].srvmsc = 255;
usuarios[0].jerarq = 0;
usuarios[0].dfault[0] = NUMUSART;
usuarios[0].dfault[1] = 31 | 128;
serv_negados(CNXACTV,0); /* el Usuario maestro debe activar los servicios
                           de conexión inmediata */
}

```

Con la siguiente rutina, se efectuará la inicialización de puertos, bajo la configuración default siguiente: usuario nulo, tipo activo, se toman en cuenta los cambios de DTR, velocidad = 9600, sin paridad, con eco y cola de espera vacía.

```

void inicia_puertos()
{ int i,j;

for(i=0;i<NUMUSART;++i) /* PUERTOS ASIGNADOS A USARTS */
{
    Lst_port[i].usuario = NUMUSUAR;
    Lst_port[i].activo = 0;
    Lst_port[i].atencion = 1;
    Lst_port[i].veloc = 4;
    Lst_port[i].paridad = 2;
    Lst_port[i].eco = 0;
    for(j=0;j<TAMQPORT;j++)
        Lst_port[i].queue[j] = NOCONECT;
}
for(i=NUMUSART;i<MAXPORT;++i)
{
    Lst_port[i].usuario = NUMUSUAR;
    Lst_port[i].activo = 1;
    Lst_port[i].atencion = 1;
    Lst_port[i].veloc = 4;
}
}

```

```
Lst_port[i].paridad = 0;
Lst_port[i].eco = 1;
for(j=0;j<TAMQPORT;j++)
    Lst_port[i].queue[j] = NOCONNECT;
}
assign_port_usu(ini_ports,0);
}
```

Inicialización de la tabla de grupos de puertos, en donde los dos campos destinados para contener los puertos del grupo, serán puestos al mismo valor que la constante NOCONNECT, como una señal de campos vacíos.

```
void inicia_grupos()
{ int i;

  for(i=0;i<NUMGPOS;++i)
  {
    *Grupo[i].nomgpo = '\0';
    Grupo[i].puertos[0] = 255;
    Grupo[i].puertos[1] = 255;
  }
}
```

La siguiente función se encarga de llamar a cada uno de los procedimientos anteriores.

```
void inicializa()
{
  int i;

  new = 1;
  inicia_matriz();
  iniciatjtas();
  crea_buff();
  listpuert(RENG_PTO);
  inicia_usuarios();
  inicia_puertos();
  inicia_grupos();
  desp_mat_conm();
}
```

A parte de las funciones principales de inicialización, se construyeron otras también básicas que son de utilidad general, y que a continuación se muestran

```
void desp_mat_conm(); /* Se encarga del desplegado de la matriz de
                        conmutación*/
```

```
void limpia_Qport(byte numport); /* Se eliminan todas las ocurrencias del
                                   puerto numport en todas las colas de espera */
```

```
byte DesQ_port(byte numport); /* Esta función regresa el puerto que sigue
                                dentro de la cola de espera del puerto numport */
```

```
byte EnQ_port(byte numport, byte portq); /* Se inserta un puerto al final de la
                                           cola de espera del puerto numport, regresando
                                           los siguientes valores:
                                           1 : Inserción correcta
                                           2 : portq ya estaba formado anteriormente
                                           3 : Ya no hay espacio */
```

```
Int Sup_Us(int usar); /* Regresa 1 si usar es el usuario maestro,
                        checando que ocupe la posición cero de la tabla
                        de usuarios, si regresa 0 se trata de cualquier otro
                        usuario*/
```

```
int busca_Us(char *token); /*Recibe como parámetro un identificador de
                             usuario (RFC) y lo busca en la tabla de usuarios,
                             regresando la posición donde lo encuentra ( de 0
                             a NUMUSUAR-1), o bien el tamaño de la tabla
                             (NUMUSUAR), como indicador de búsqueda sin
                             éxito */
```

```
int serv_permitidos(int usar, int serv); /* Regresa 1 cuando un usuario
                                           específico tiene acceso a un determinado servicio,
                                           denominando a cada usuario por su posición y a
```

cada servicio por la constante que lo define, en otro caso regresa 0 */

```
int port_permitido(int usar,int puerto); /* Regresa 1 cuando un usuario específico tiene acceso a un determinado puerto, denominando a cada usuario por su posición y enviando el número de puerto, en otro caso el valor regresado será 0 */
```

```
void serv_asignados(byte serv, int usu); /* Esta es la rutina que modifica la máscara de acceso a puertos de el usuario usu, poniendo en uno el bit correspondiente al servicio autorizado*/
```

```
void port_asignados(byte pto, byte *ptomsc) /* Se reciben como parámetros el puerto autorizado, junto con una máscara de acceso a puertos, como esta última se integra de 32 bits, se almacenan en un arreglo de 4 byte's. La rutina modifica dicha máscara encendiendo el bit que identifica al puerto pto, como señal de autorización */
```

```
void serv_negados(byte serv, int usu);/* Esta rutina modifica la máscara de acceso a puertos de el usuario usu, poniendo en cero el bit correspondiente al servicio negado*/
```

```
void port_negados(byte pto, byte *ptomsc);/* Se reciben como parámetros el puerto autorizado, junto con una máscara de acceso a puertos, como esta última se integra de 32 bits, se almacenan en un arreglo de 4 byte's. La rutina modifica dicha máscara apagando el bit que identifica al puerto pto, como señal de prohibición */
```

```
si_evento(int argc, byte NumEvento); /* Esta función activa un evento
                                     específico, dentro de la máscara de eventos del
                                     usart argc, referenciándolo mediante el número de
                                     bit correspondiente*/
```

```
no_evento(int argc, byte NumEvento); /* Esta función desactiva un evento
                                     específico, dentro de la máscara de eventos del
                                     usart argc, referenciándolo mediante el número de
                                     bit correspondiente*/
```

```
void enviacadena(int argc, int msge, char *prm1); /* Esta es la función para el
                                                  envío de mensajes a algún usuario, si se quieren
                                                  hacer correcciones o cambios de idioma, bastará
                                                  con modificarla*/
```

IV.2.2 Funciones del conmutador

```
void conecta(byte orig, byte dest);/* Conexión unidireccional del puerto orig
                                     hacia el puerto dest */
```

```
void desconecta(byte puerto); /* Desconexión unidireccional */
```

```
void enlaza(byte orig, byte dest); /* Conexión bidireccional entre los puertos
                                     orig y dest */
```

```
void desenlaza(byte puerto); /* Desconexión bidireccional */
```

```
int habilitado(int pto_dest); /*Regresa cierto si un puerto tiene su señal
                                DTR encendida */
```

```
void leeDTR(); /* Debe leer puertos de hardware para checar la
               señal DTR, en la simulación no hace nada */

void activa_dtr_tjta(byte pto, int señal); /* Se activa la señal DTR o DSR de
      algún puerto en la tarjeta de puertos, si la señal a
      activar es DTR, entonces el parámetro señal
      deberá ser INPUT, si se trata de un DSR, el
      parámetro señal deberá ser OUTPUT */

void desact_dtr_tjta(byte pto, int señal); /* Se desactiva la señal DTR o DSR
      de algún puerto en la tarjeta de puertos, si la
      señal a activar es DTR, entonces el parámetro
      señal deberá ser INPUT, si se trata de un
      DSR, el parámetro será OUTPUT */

void Detecta(); /* Detecta cuándo hubo una variación de DTR en
                algún puerto y lo inserta en el buffer de solicitudes
                para ser atendido posteriormente*/

void period_proc(); /* Procedimiento periódico que se encargará de
                    detectar los cambios de DTR en los puertos, al
                    llamar a la rutina Detecta */
```

IV.2.3 Funciones del analizador lexicográfico

La función `lexx` recibe como parámetro un string (comando), una variable tipo entero y otro string en el que almacenará el token obtenido. Después de llamarla, el comando se modificará al eliminar su primer token más a la izquierda, quedando con una longitud menor a la inicial. La variable tipo regresará uno de los siguientes valores, según sea el caso:

tipo	Token
3	REAL
4	ENTERO
5	PAL_RES
6	IDENT

7	NO_DEFIN
8	OP_ASSIGN
9	INTERVALO
10	SEPARADOR
46	PUNTO
92	B_SLASH

```
char *lexx(char *comando,int *tipo, char *token);
```

Suponiendo que tenemos un comando y se quiere determinar si su primer token es una palabra reservada, habrá que llamar a lexx como sigue:

```
comando = lexx(comando,&tipo,token);
```

la función deberá regresar el valor de tipo igual a PAL_RES. Si se necesita saber de qué palabra reservada se trata, mediante la función busca se obtiene la posición en la tabla de palabras reservadas (Plb_Rsrv) del token obtenido.

```
pos = busca(token);
```

Si se busca un token que no se encuentra en la tabla de palabras reservadas, entonces la posición que se regresará es la longitud de la tabla.

IV.3 Servicios para usuarios

En esta sección se describen las funciones relacionadas con los servicios del Sistema, haciendo referencia a los diagramas que se encuentran en la parte final de éste capítulo, ver también apéndice C (LED).

Algoritmo Usart

El algoritmo principal que sigue un usart al atender las solicitudes, se representa básicamente mediante la figura 4.1. El bloque de inicialización se encarga de limpiar las variables de control, incluidas en el byte de máscara de eventos, tales como: Bandera que indica la ocurrencia de un time-out de respuesta, bandera que indica la ocurrencia de un time-out de sesión, bandera que indica si el puerto que se está atendiendo se ha apagado, otra cuando hay un puerto libre por el cual el usuario que atiende espera, junto con la variable

que contiene, en tal caso, el puerto que se desocupó, y por último la bandera que indica el envío del último carácter en una sesión.

En la espera de una solicitud, el usart quedará bloqueado. Cuando ocurre alguna, el usart la toma y determina si es un puerto con el que se puede establecer comunicación (activo), en caso afirmativo si su señal DTR se encendió, se concede una sesión, si la señal no está encendida implica una desconexión. Cuando un dispositivo se apaga, hay que borrar todas sus ocurrencias de las colas de espera en que aparecía. Si se trata de un puerto pasivo, únicamente se procesa su desconexión.

El usart ignora los siguientes casos (ver fig. 4.1):

- Cuando hay una señal DTR encendida y el puerto está conectado
- Cuando hay una señal DTR apagada y el puerto está deshabilitado

Algoritmo de Desconexión por deshabilitación de un puerto

El algoritmo se ilustra en el diagrama de la figura 4.2. Este procedimiento realiza la desconexión de un puerto que se ha deshabilitado o apagado, mandando un mensaje de desconexión al puerto con el que interactuaba, si acaso este es un puerto de tipo activo. El puerto que se ha apagado debe desaparecer de todas las colas de espera en que se encontraba, y el que queda libre revisará si hay alguien en su cola de espera para conectarse, este procedimiento PROC_QE se aprecia mediante el diagrama de la figura 4.5.

Algoritmo de Sesión

Cuando un usart entabla una sesión con algún usuario, su primera acción es enlazarse con el puerto que ocupa y poner en marcha el temporizador de sesión. A continuación se pide su RFC y clave de acceso, checando contra la tabla de usuarios. Si los datos son correctos y dicho usuario tiene acceso al servicio de conexión inmediata, trata de conectarlo, si el usuario no acepta la conexión, entonces procede a recibir comandos y ejecutarlos, mandando mensajes en caso de error de sintaxis o servicios y comandos no permitidos. El algoritmo se representa mediante el diagrama de la figura 4.3.

Una sesión terminará en los siguientes casos:

- Cuando el tiempo de sesión termine
- Cuando el usuario ha establecido una conexión con algún otro puerto, en cuyo caso se eliminará de las colas de espera en que aparezca.

- Cuando ha ocurrido un time-out por la no recepción de caracteres
- Cuando se apague la terminal

Algoritmo de confirmación de cola de espera(CONFIRMA QE)

Este algoritmo se representa mediante el diagrama de la figura 4.4, y su función es la de confirmar con el usuario la conexión con un puerto por el cual esperaba, si el usuario acepta, se establece la conexión y la sesión termina, en otro caso el usart tratará de conectar al que sigue en la cola de espera del puerto si es que hay alguien más esperando. Si hay algún puerto en cola de espera que está interactuando con un usart, se le indicará a tal usart mediante el evento de cola de espera, así como el puerto que está libre.

Función HAZ_CNX

Esta función establece la conexión entre dos puertos, mandándoles mensajes de conexión efectuada, en caso de que éstos sean de tipo activo. Una vez conectados, elimina todas sus ocurrencias en las colas de espera de otros puertos, figura 4.6.

Función de Conexión inmediata (CNX_INM)

Cuando un usuario puede acceder el servicio de conexión inmediata, tiene posibilidad de usarlo o no. En caso de tenerlo activado, en cada sesión con un usart, se le permitirá conectarse directamente a alguno de los puertos definidos por él mismo en su grupo default de puertos. Si ninguno está libre o disponible, el procedimiento terminará como si no hubiese pasado nada, en otro caso, se pedirá la confirmación al usuario antes de realizar la conexión, figura 4.7.

Función de Conexión (CNC)

El algoritmo de conexión se aprecia mediante el diagrama de la figura 4.8. Cuando un usuario quiere conectarse con algún puerto X, se checarán los siguientes aspectos:

- Que X sea un puerto permitido

- Si se trata de un puerto del cual se toman en cuenta las variaciones de su DTR, debe estar habilitado

Si el puerto X cumple con los requisitos mencionados y no está conectado, se establecerá la conexión sin algún problema. Sin embargo, cuando el puerto X cumple con esos requisitos y está conectado con un puerto Y, entonces quedan 3 posibilidades:

- Si el usuario goza del servicio de Desalojo y su prioridad es mayor que la de X y Y, se desenlazarán el puerto X para después conectarse con el puerto que ocupa el usuario.
- Si el usuario goza del servicio de cola de espera y hay cupo dentro de la cola de espera del puerto X, se insertará en ella hasta que sea su turno.
- Se enviará al usuario un mensaje avisándole de la conexión infructuosa.

No se podrá establecer una conexión con algún puerto que esté en sesión con algún usart.

Función de Desalojo de usuarios de baja prioridad (DESLOJA)

Esta función determina si un usuario tiene prioridad mayor que cualquiera de otros dos que forman una conexión, si esto se cumple, dicha conexión podrá deshacerse, avisando a los usuarios de tal hecho si es que los puertos son de tipo activo, y estableciendo a continuación la conexión solicitada por el usuario de mayor prioridad.

El puerto que queda libre, será procesado para revisar su cola de espera, figura 4.9.

Función de Conexión externa (CNXE)

En esta función se checa que el usuario tenga permitido el uso de este servicio, en seguida se valida que los puertos que quiere conectar cumplan con lo siguiente:

- Que sean puertos ajenos al que ocupa el usuario
- Que ninguno de los dos esté conectado
- Que, en caso de importar las variaciones de sus DTR, éstos estén habilitados
- Que caigan dentro del rango de puertos válidos (5/32)

Si son válidos, entonces se procede a conectarlos, mandando a cada uno un mensaje de conexión establecida en caso de ser de tipo activo. Finalmente se avisa al usuario que solicitó la conexión externa, si tuvo lugar o no tal petición, figura 4.10.

Función de Desconexión externa (DNXE)

La función DNXE, checa que el usuario que la solicita tenga permitido el acceso a este servicio, a continuación realiza la siguiente validación referente al puerto X a desconectar:

- Que sea ajeno al puerto que ocupa el usuario
- Que esté conectado
- Que pertenezca al rango de puertos válidos(5-32)

Cuando se cumplen estos requerimientos, se mandan mensajes de desconexión al puerto X y al que está conectado, en caso de que sean de tipo activo. Después se efectúa el desenlace y se avisa al usuario que solicitó el servicio, del resultado de la operación, figura 4.11.

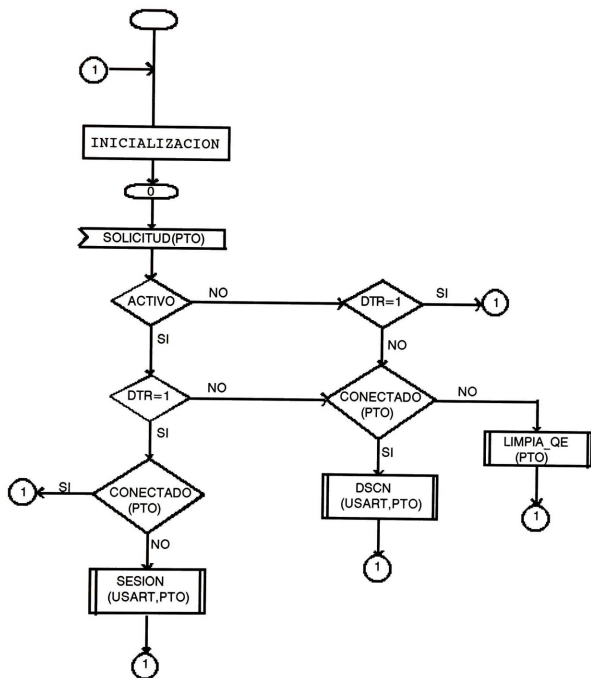


Fig. 4.1

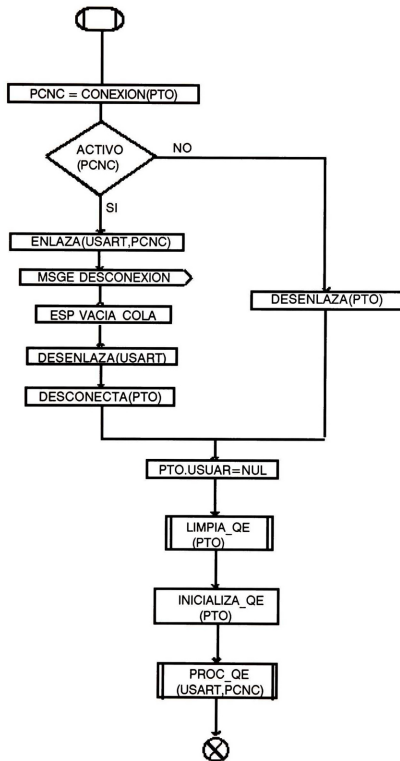


Fig. 4.2

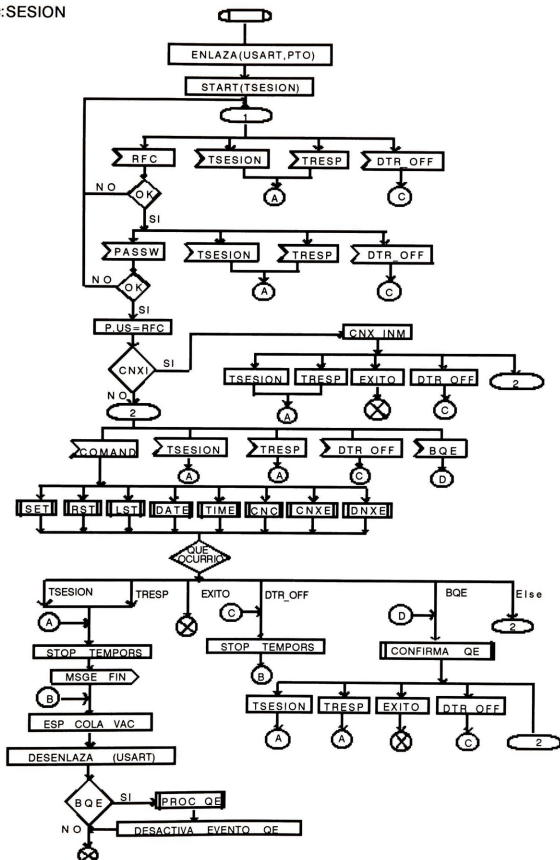


Fig. 4.3

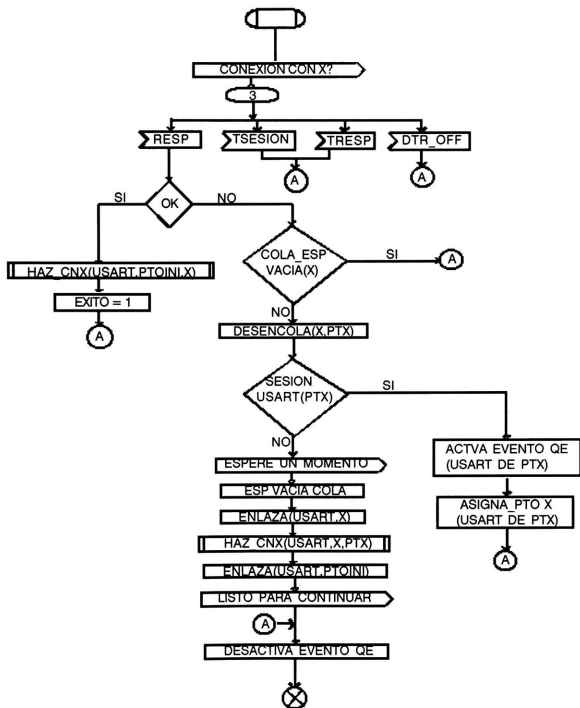


Fig. 4.4

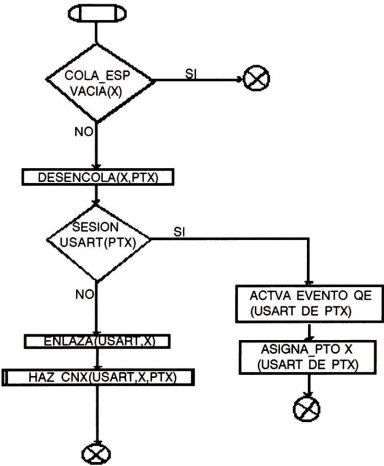


Fig. 4.5

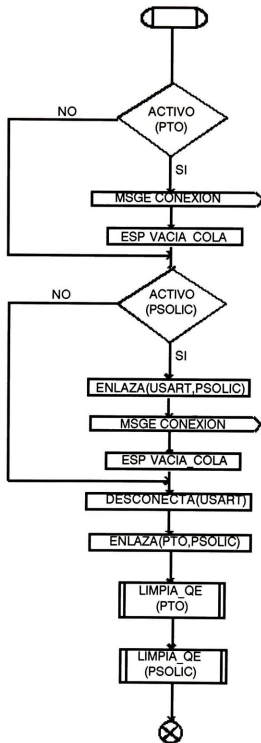


Fig. 4.6

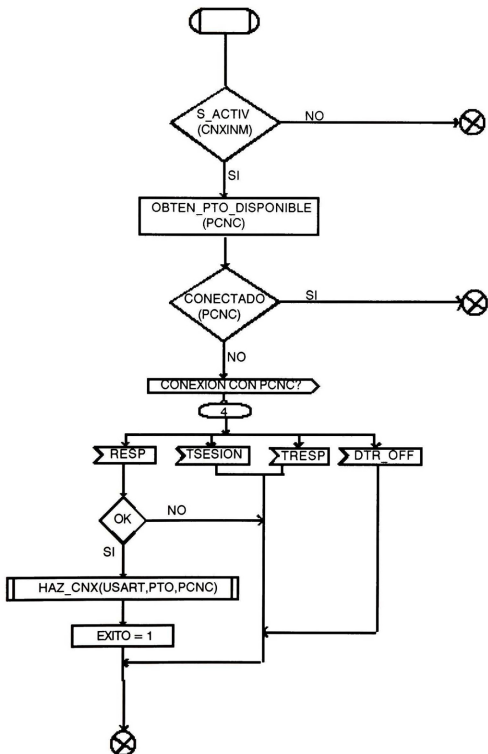


Fig. 4.7

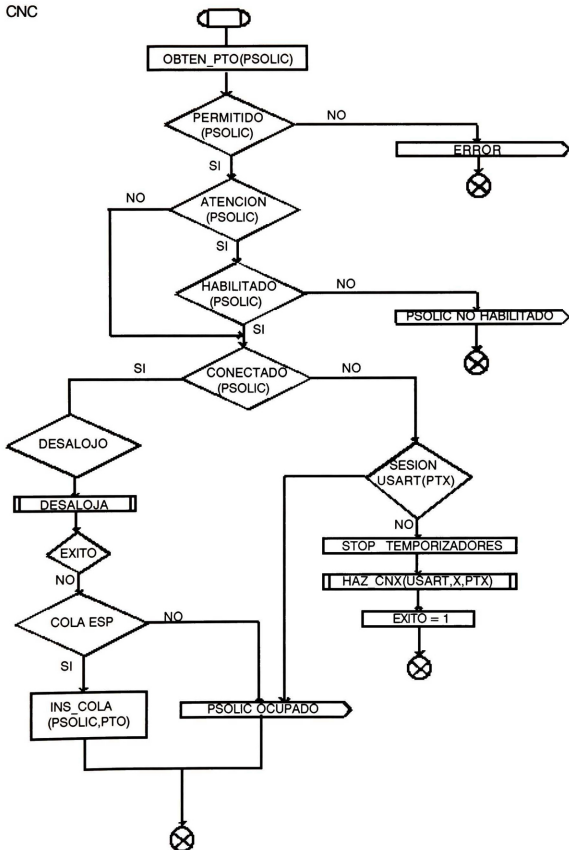


Fig. 4.8

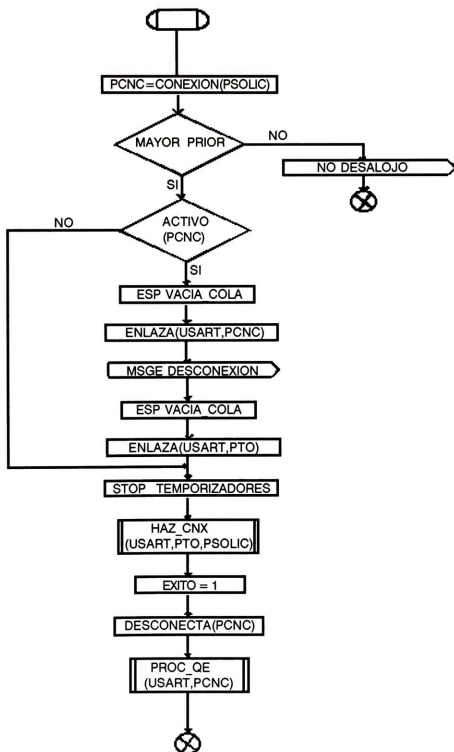


Fig. 4.9

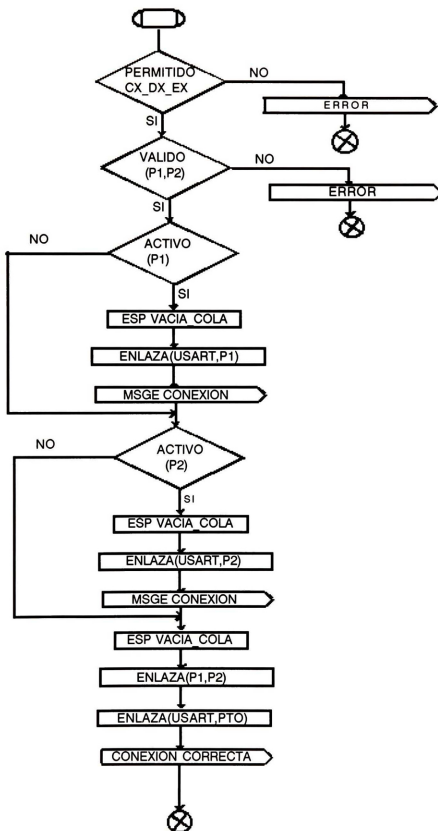


Fig. 4.10

Proc: DNxE

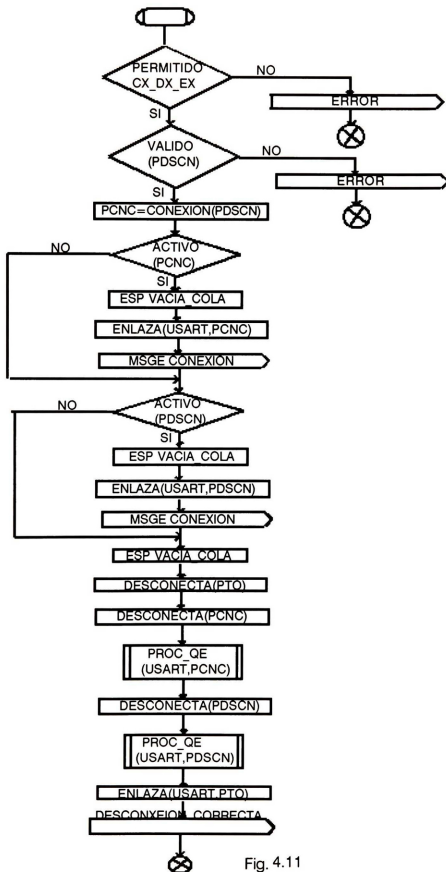


Fig. 4.11

V. MANEJADOR DE E/S

Para el manejo de E/S se han implantado dos buffers circulares de bytes, uno para la recepción de caracteres y otro para la transmisión, controlados mediante el empleo de semáforos. En las siguientes dos secciones se describe su implantación.

V.1 Funciones de entrada

Las dos funciones principales, están dedicadas a obtener un caracter del buffer de entrada y en base a ésto, obtener un string completo para su procesamiento. Estas son referidas como tgetc y tgets.

Función tgetc:

La función tgetc recibe como parámetros, el número de usart, y una variable que indica si se le hará eco o no al caracter recibido.

Su función es la de obtener un caracter del buffer circular de entrada de un usart, ésto se controla mediante el semáforo tty[argc].sinp, indicando si hay algún caracter dentro del buffer, de no haberlo, se producirá un bloqueo hasta que tal caracter llegue.

Para evitar el bloqueo permanente, en caso de no llegar algún caracter en el buffer de entrada, cada vez que se ejecuta la función, se activa un temporizador referente al tiempo máximo de respuesta (tty[argc].tempor), que se encarga de mandarlo, indicando evento de Time-Out.

Si el caracter se recibe sin problemas, se detiene inmediatamente el temporizador activado y se da un signal en el semáforo que indica espacio disponible en el buffer de entrada (tty[argc].inok), procediendo a darle eco según sea el caso.

A los caracteres de control como '\0' y HHELP no se les da eco.

Las funciones tgetc y las relacionada con temporizadores de Time-Out de caracter y de sesión, se muestran a continuación.

```
byte tgetc(int argc,byte eco) /*Operacion de ENTRADA del USART argc,  
    retirando un caracter del buffer circular de entrada */
```

```

{
    byte carac;

    startt(tty[argc].tempor);
    wait(tty[argc].sinp);
    carac = getbuff(tty[argc].bin);
    stopt(tty[argc].tempor);
    signal(tty[argc].inok);
    if ( eco )
    { if ( (carac == '\0') || (carac == (byte) HHELP) )
        tputc(argc, '\n');
        else
            tputc(argc, carac);
    }
    return(carac);
}

/* Time-Out por sesión*/
void t_sesion(word argc)
{
    insbuff(tty[argc].bin, '\0');
    si_evento(argc, BSESSION); /* se indica el evento */
    signal(tty[argc].sinp);
}

/* Time-Out por caracter */
void t_in_us(word argc)
{
    insbuff(tty[argc].bin, '\0');
    si_evento(argc, BTEMPOR); /* se indica el evento */
    signal(tty[argc].sinp);
}

```

La función tgets :

Esta función recibe como parámetro el número de Usart, y una variable de indicación de eco. Su función es la de regresar un comando completo para ser procesado por el sistema.

Básicamente, su tarea es la de ir concatenando caracteres, obtenidos al invocar a la función tgetc, hasta que obtenga un fin de línea. Esta concatenación puede ser suspendida debido al surgimiento de eventos tales como :

- Ocurrencia de un Time-Out por falta de caracteres en el buffer de entrada
- Fin de sesión
- El dispositivo se ha apagado (DTR = 0)
- Hay solicitud de ayuda

En tales casos, siempre que se llame a la función `tgets`, deberá chequearse la aparición de alguno de estos eventos para tomar las correspondientes medidas.

`void tgets(int argc,byte eco) /* obtiene una cadena de caracteres del buffer de entrada de un usart */`

```
{
    int i=0;
    char car;

    espace[argc][0] = '\0';
    car = tgetc(argc,eco);
    while ( (car != '\0') && (!hay_evento(argc,BTEMPOR)) && (car != HHELP) &&
        (!hay_evento(argc,BDTR)) && (!hay_evento(argc,BSESSION)) )
    {
        if ( ((car == 8) || (car == 180)) && (i>0) ) i--;
        else
        if (car != '\n')
        {
            espace[argc][i] = car;
            i++;
        }
        car = tgetc(argc,eco);
    }
    if (car == HHELP)
    {
        espace[argc][i] = HHELP;
        i++;
    }
    espace[argc][i] = '\0';
    tty[argc].str = espace[argc];
}
```

El procedimiento que simula el envío de caracteres de parte de algún dispositivo, al buffer de entrada de un usart se controla con los siguientes

semáforos: **inok** para checar espacio libre en el buffer de entrada y **sinp** para avisar que hay caracteres listos en el buffer.

```
void ParaUsart(char car)
{
    byte pto;

    if ( matriz[Opcion-1] != NOCONNECT )
    {
        wait(tty[Opcion-1].inok);
        insbuff(tty[Opcion-1].bin,car);
        signal(tty[Opcion-1].sinp);
    }
    else { *comando = '\0';
           Rutina_error(7,'\0');
        }
    }
}
```

V.2 Funciones de salida

Las funciones relacionadas con la operación de salida, se encargan de enviar un caracter por el buffer destinado a la salida de datos de un Usart, y en base a ésto, mandar un string completo. Tales funciones son tputc y tputs.

Función tputc:

Esta función recibe como parámetros el número de usart de quien se trata y el caracter a insertar en el buffer circular de salida.

Primero espera a que haya un espacio disponible, ayudándose del semáforo dedicado a la detección de espacio libre en el buffer circular de salida. Cuando esto ocurre inserta el caracter en el buffer, deshabilitando interrupciones.

```
void tputc(int argc, char c)    /*Operacion de SALIDA del USART argc,
                                insertando un caracter en el buffer circular de
                                salida */
{
    wait(tty[argc].sout); /* al menos un lugar vacio */
    beginat();
    insbuff(tty[argc].bout,c);
    endat();
}
```

```
}
```

Función tpuc:

Recibe como parámetros el número de usart, y una cadena de caracteres para enviar.

Básicamente hace una sucesión de llamadas a la función tputc, para cada uno de los caracteres de la cadena.

```
void tputs(int argc, char *str)
/* envia una cadena de caracteres al buffer de salida de un usart */
{
    while(*str != '\0')
    {
        tputc(argc,*str);
        str++;
    }
    tputc(argc,*str);
}
```

El procedimiento vacia_colas_usart(), se encarga de simular la recepción de mensajes del sistema en cada dispositivo, desplegándolos en la ventana correspondiente. El procedimiento general es detectar que cada buffer de salida de un usart no esté vacío, para poder sacar un byte, desplegándolo según sea el caso. A continuación, dará un signal al semáforo que indica espacio disponible en el buffer de salida del usart i (signal(tty[i].sout). Otra función es la de hacer un signal al semáforo **sok** de cada usart, para avisarle que se ha mandado el último caracter a un dispositivo, esto será posible cuando los mismos usarts se lo indiquen al encender la bandera **bok**.

A continuación se muestra dicho procedimiento

```
void vacia_colas_usart()
{
    int i=0;
    byte car;
    for (i=0;i<NUMUSART;i++)
    {
        while (!emptybuff(tty[i].bout))
        {
            setcurw(3,cont[i],w[i]);
        }
    }
}
```

```
car = getbuff(tty[i].bout);
if ( (car == 8) || (car == 180) )
{ if (cont[i] > 0)
  { cont[i]--;
    setcurw(3,cont[i],w[i]);
    printfw(w[i], " ");
  }
}
else
if (car != '\0')
{
  printfw(w[i], "%c", car);
  cont[i]++;
}
if (car == '\n') cont[i] = 0;
signal(tty[i].sout);
}
if (hay_evento(i,BOK)
{ no_evento(i,BOK);
signal(tty[i].sok);
}
}
}
```

VI. CONCLUSIONES

Se puede decir que el software de servicios desarrollado, ha cumplido con los objetivos establecidos, añadiendo los siguientes comentarios que pueden considerarse como aspectos concluyentes de este trabajo.

Durante la evolución del proyecto, hubo la necesidad de hacer la simulación del hardware del conmutador, para que sobre éste, se implantaran las rutinas de servicios, tal simulación demandó el conocimiento de las características fundamentales del conmutador, para apegarse más a la realidad, implicando sólo unos cambios en determinados puntos específicos, al momento de implantación. Las dificultades que surgieron, se debieron al establecimiento y especificación de estructuras de datos adecuadas, de tal modo que reflejaran transparentemente los componentes del hardware. También la especificación de procesos periódicos, temporizadores y procesos Usart, necesarios para el buen funcionamiento del conmutador. La ocurrencia de interrupciones por llegada de algún carácter, se simuló mediante el teclado. Este módulo fué también complejo, ya que se requería saber si lo obtenido por el teclado era para algún buffer de entrada de un determinado Usart, o bien era un evento externo como la modificación de señales DTR, para esto tuvieron que incluirse variables globales de control y uso de ventanas. Para apreciar y checar la simulación, se despliegan la matriz de conmutación y el estado de la señal DTR para cada puerto.

Planeando un posible crecimiento del sistema, y como algunas funciones no se ven tan simples, se expresaron los algoritmos mediante la representación gráfica LED (Lenguaje de especificación y descripción para algoritmos de conmutación y telecomunicaciones, basado en un modelo de máquina de estados finitos ampliada). De esta manera, se puede visualizar en general el funcionamiento de ciertos módulos importantes, haciendo de una manera más fácil la inclusión de alguna otra rutina si se desea posteriormente.

Dentro de la lista de servicios, se pusieron en marcha la mayoría de ellos, exceptuando los siguientes: Comunicación multipunto, correo electrónico, deshabilitación de la interrupción del enlace por otros usuarios, estadísticas y temporización del uso de puertos. Considerándolos como de más alto nivel, y que por tanto, se pueden implementar de manera aislada, usando las funciones más primitivas, necesitando algunos de ellos, de una cierta cantidad de memoria para su desarrollo. No obstante se han establecido las rutinas fundamentales de comunicación y operaciones de E/S, en base a las cuales se podrían diseñar. Lo único que se pide es el espacio en memoria necesario para asignar a cada usuario la capacidad de recepción y envío de mensajes. Igualmente con los puertos, cada uno debería tener su propio temporizador para restringir su uso respecto al tiempo.

Software de servicios

Los mensajes enviados a un usuario, son factibles de modificación, pues pensando en esto, se agruparon en un determinado módulo.

En cuanto al módulo de ayuda, los desplegados hacen referencia a la tabla de palabras reservadas, así que si se hacen cambios o correcciones de nombres de comandos en ella, el módulo de ayuda funcionará normalmente.

Otro posible cambio a futuro, es la implementación de una dinámica de mensajes, es decir proporcionar un esquema donde se disponga de ellos en varios idiomas, seleccionables mediante algún parámetro adicional.

En general puede decirse, que el software desarrollado ofrece los servicios básicos para cubrir las necesidades de comunicación digital, de una manera fácil y transparente entre los usuarios, sin requerir de un costo considerable.

Apéndice A

MANUAL DE USUARIO

Cuando un usuario desea conectarse con algún dispositivo, mediante el conmutador, deberá encender su terminal. A continuación se entablará una conversación con el sistema, pidiéndole antes de iniciar cualquier acción de conexión o algún otro servicio, el registro federal de causantes del usuario, que deberá estar dado de alta previamente.

En seguida, el usuario deberá proporcionar su clave de acceso o password, la cual no hará eco en el monitor, por cuestiones de seguridad.

Cuando el RFC y la clave de acceso son correctos, el usuario podrá entrar en plena sesión con el sistema, accediendo los puertos y servicios que le fueron otorgados.

Cada vez que se quiera hacer uso de algún servicio, o acceder un puerto, se checará el derecho sobre ellos, dependiendo del usuario que los invoque. Si no se cuenta con tal derecho, el sistema le advertirá mediante el mensaje correspondiente.

Otros tipos de mensaje que un usuario puede recibir son los siguientes:

- Error de sintaxis en los comandos especificados
- Mensaje de operación exitosa
- Mensaje de operación infructuosa
- Mensaje de fin de sesión
- Mensaje de abortamiento de sesión

Error de sintaxis

Los mensajes de sintaxis, aparecen cuando se escribe un comando con errores ortográficos, que son tomados como parámetros erróneos, de igual manera, cuando ocurre la falta o exceso de ellos. Otra causa puede ser debida a la omisión de operadores adecuados de asignación o de indicación de intervalos, en los comandos que se requieren.

Los siguientes son una muestra de esta clase de mensajes enviados por el sistema:

"Parámetro incorrecto "
"Comando invalido "
"Demasiados parámetros"

Software de servicios

"Operador de asignacion invalido"

"Separador invalido "

"Error. primero password actual"

"Error. falta password nuevo"

"Operador de intervalo invalido"

Operación exitosa

Cuando se ha solicitado un servicio o un comando, y la operación resultó exitosa, los mensajes más comunes son:

"Usuario X dado de alta"

"Usuario X dado de baja"

"Tabla de Usuarios vacia"

"Conexion P1-P2 Establecida"

"Desconexión efectuada"

"Conexión efectuada"

"En espera del puerto"

Si hay que confirmar una operación, el sistema enviará mensajes como los que se muestran a continuación, y dependiendo de la respuesta se procederá adecuadamente:

"Acepta conexion con X (S/N)?"

"Acepta Cola de espera de X (S/N)?"

En algunas ocasiones, el sistema suspende la sesión con un usuario durante unos instantes, debido al procesamiento de colas de espera, para ello, manda los siguientes mensajes.

"Espere un momento"

"Listo para continuar"

Para el desplegado del estado de puertos, usuarios, conmutador, colas de espera, servicios y puertos permitidos, el formato de salida será el siguiente:

En el desplegado de status de puertos, se mostrará en forma de columna, el número de puerto del cual se trata, su tipo (1=activo o 0=pasivo), si se consideran los cambios de la señal DTR (1=si , 0=no), su paridad (0=par,

Software de servicios

1=impar, 2=sin paridad), si es un puerto con eco (1=si, 0=no), y por último la velocidad con la cual transmite. El encabezado es el siguiente:

"Port Type DTR Parit Echo Speed"

Cuando se muestra el estado del conmutador, se presentan en forma de columna los siguientes datos: Número de puerto, RFC del usuario que lo ocupa, puerto con el que está conectado y el RFC del usuario lo que ocupa.

Si algún puerto es de tipo pasivo o es un puerto al cual no se le toman en cuenta los cambios de DTR, el campo referente al RFC aparecerá vacío. El encabezado es el siguiente:

"-Pto--Usuario- -Cnx--Usuario-"

En el despliegado de colas de espera, se muestra una relación entre un puerto y los que están en espera de conexión. Si el puerto no tiene elementos en su cola de espera, no aparecerá en esta lista. Encabezado:

"-Pto- -ColaEspera"

Cuando se solicita el despliegue de usuarios para apreciar algunas de sus características, éstas se aparecerán en forma de columna, mostrando los atributos como RFC del usuario, clave de acceso (Passw), cola de espera (QE=1 si está permitido o QE=0 en otro caso), conexión inmediata (CI=1, si está permitido y CI=0 en otro caso), jerarquía (JR), desalojo de usuarios (DU=1, si está permitido y DU=0 en otro caso), conexión inmediata activada (en el caso de que un usuario tenga activo este servicio CXA=1, de otro modo CXA=0), y por último conexión y desconexión externa (CDEX=1 si está permitido y CDEX=0 en otro caso).

" Usuario Passw QE CI JR DU CXA CDEX"

Operación infructuosa

Cuando una operación no ha tenido lugar, los mensajes más comunes son:

"RFC no existe"

"RFC invalido "

Software de servicios

"Password incorrecto"

"Puerto invalido "

"Paridad invalida, 0=PP 1=PI 2=SP"

"Grupo no definido "

"Password actual incorrecto"

"Usuario ya existe"

"Usuario no existe"

"Superusuario no accesible"

"Comando no permitido"

"Servicio no permitido "

"La fecha no es valida"

"Tabla de usuarios llena"

"Servicio no permitido"

"Grupo ya existente "

"Espacio insuficiente en tabla de gpos"

"Puerto(s) ocupado(s) / Inaccesible(s)"

"Puerto destino X deshabilitado"

"Puerto destino X desconectado"

"Puerto inaccesible"

"Puerto ya desconectado"

"Sigue en espera del puerto "

Fin de sesion

Mensaje de fin de sesion:

"Sesion terminada"

Esta es la terminación normal, sin embargo, una vez que el usuario se conecta con algún otro dispositivo, la sesión se da por terminada.

Suspensión de sesión

Mensaje de abortamiento de sesión por falta de respuesta en un determinado intervalo de tiempo:

"Timeout, activese nuevamente"

Si al terminar una sesión el usuario no pudo conectarse o bien no completó sus objetivos, quedan dos posibilidades: la primera, en caso de estar en cola de espera de un puerto, esperar su turno. Con ésta opción se corre el riesgo de

Software de servicios

esperar un periodo de tiempo muy largo, o bien que otro usuario en el sistema pueda conectarse con su puerto, tomando en cuenta que está libre. La otra posibilidad, muchas veces más factible, es la de apagar y encender el dispositivo, con fines de obtener una nueva sesión.

Otra de las facilidades que se ofrece al usuario es un programa de ayuda, que se describirá en el apéndice B.

Apéndice B

MODULO DE AYUDA

Cada vez que el usuario tenga duda referente a algún comando, podrá llamar mediante la pulsación de la tecla F6, al módulo de ayuda. Este módulo desplegará los posibles comandos que puede ejecutar en determinado momento.

La ayuda se realiza por niveles, es decir, los comandos básicos son SET, RST, LST, CNXE, DNXE y CNC, si el usuario no sabe cómo empezar y pulsa la tecla F6, la ayuda será la siguiente:

Comm: SET | RST | LST | CNXE | DNXE | CNC

Una vez que elige alguno de ellos, podrá solicitar nuevamente la ayuda, al escribirlo y pulsar enseguida la tecla F6. De esta manera, para cada uno de éstos obtendrá su respectiva descripción, como a continuación se muestra:

Si se escribe SET y F6, el mensaje serán todos los posibles comandos relacionados con el comando SET:

SET US|QE|CI|JR|DU|PU|SU|PW|PT|AU|GP|DEF_PORT

Nuevamente el usuario podrá teclear el comando SET seguido de algún otro comando seleccionado y pulsar F6, obteniendo finalmente la especificación sintáctica del comando completo, mediante una especie de ejemplo.

Para la invocación de ayuda en el caso SET, se tienen los siguientes mensajes:

SET PW F6:
 SET PW RFC clave
 SET PW clvold clvnew

SET QE F6:
 SET QE RFC 0|1

SET CI F6:
 SET CI RFC 0|1

SET JR F6:
 SET JR RFC 0|1

Software de servicios

SET DU **F6**:

SET DU RFC 0|1

SET PU **F6**:

SET PU RFC p1, p3/p7, p5,...

SET PU RFC ALL

SET SU **F6**:

SET SU RFC 1, 3/5,...

SET SU RFC ALL

SET US **F6**:

SET US RFC RFC RFC

SET PT **F6**:

SET PT TYP:1 REQ:1 ECH:1 PAR:1 SPD:600

SET GP **F6**:

SET GP nomgpo p1 | p1,p2 | p1/p2

SET GP p1 | p1,p2 | p1/p2

SET AU **F6**:

SET AU QE:1 CI:1 DU:0 JR:5 PU:8,9/12,20 SU:1,5

SET DEF_PORT **F6**:

SET DEF_PORT p1, p4/p8, p3, p9/p12,...

SET DEF_PORT ALL

Si se elige RST y **F6**, el mensaje serán todos los posibles comandos relacionados con el comando RST:

RST PU|SU|US|GP|DEF_PORT

Nuevamente el usuario podrá teclear el comando RST seguido de algun otro comando seleccionado y pulsar F6, obteniendo finalmente la especificación sintáctica del comando completo, mediante una especie de ejemplo.

Para la invocación de ayuda en el caso RST, se tienen los siguientes mensajes:

Software de servicios

RST PU **F6**:

```
RST PU RFC p1, p4/p8, p3, p9/p12,...  
RST PU RFC ALL
```

RST SU **F6**:

```
RST SU RFC 5, 4, 1/3,..  
RST SU RFC ALL
```

RST US **F6**:

```
RST US RFC1, RFC2, RFC3,..  
RST US ALL
```

RST GP **F6**:

```
RST GP nom1, nom2,... | ALL  
RST GP
```

RST DEF_PORT **F6**:

```
RST DEF_PORT p1/p4, p6,..  
RST DEF_PORT ALL
```

Si se elige LST y **F6**, la ayuda mostrará todos los posibles comandos relacionados con el comando LST:

LST PW|QE|JR|PU|SU|US|PT|GP|CONM|DEF_PORT

Nuevamente el usuario podrá teclear el comando LST seguido de algún otro comando seleccionado y pulsar F6, obteniendo finalmente la especificación sintáctica del comando completo, mediante una especie de ejemplo.

Para la invocación de ayuda en el caso LST, se tienen los siguientes mensajes:

LST PW **F6**:

```
LST PW RFC | LST PW ALL
```

LST QE **F6**:

```
LST QE
```

LST JR **F6**:

```
LST JR
```

Software de servicios

LST PU **F6:**

LST PU RFC,RFC,... | LST PU ALL | LST PU

LST SU **F6:**

LST SU RFC,RFC,... | LST SU ALL | LST SU

LST US **F6:**

LST US

LST PT **F6:**

LST PT ALL | LST PT p1, p3/p7,...

LST GP **F6:**

LST GP ALL | LST GP nom1,nom2, | LST GP

LST CONM **F6:**

LST CONM

LST DEF_PORT **F6:**

LST DEF_PORT

Para el resto de comandos, la ayuda será la siguiente:

CNXX **F6:**

CNXX port1, port2

DNXX **F6:**

DNXX port1

CNC **F6:**

CNC port1 | CNC port1/port2

Cuando se pide la ayuda referente a un comando no permitido o propio del usuario maestro, se envía el siguiente mensaje:

"Comando propio del usuario maestro"

Apéndice C

SIMBOLOGÍA DE LED

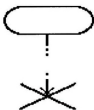
El Lenguaje de Especificación y Descripción del CCITT, conocido por la sigla LED, se ha diseñado teniendo presente la especificación del comportamiento de los sistemas de conmutación para telecomunicaciones. La parte a la cual se refiere éste apéndice, es a un subconjunto de la representación gráfica LED, empleado en los diagramas presentados en este trabajo.

Proceso

Un proceso es una máquina de estado finito ampliada que define el comportamiento dinámico de un sistema. Básicamente, en un estado el proceso está esperando señales. Al recibir una señal, responde efectuando las acciones que están especificadas para cada tipo de señal que puede recibir. Una vez que han tenido lugar todas las acciones asociadas a la recepción de una determinada señal, se pasa a un nuevo estado y el proceso espera otra señal.

Su representación gráfica es una secuencia de símbolos que están conectados por arcos dirigidos (en forma similar a un diagrama de flujo), entre éstos símbolos se encuentran los de estado, señales de entrada, señales de salida, procedimientos y conectores.

Los símbolos de arranque y de parada para un proceso son los siguientes:



**Símbolos de arranque y
de parada para un proceso**

Estado

Un estado es un punto en el proceso en el que no se ejecutan acciones sino que se espera la llegada de señales entrantes, la llegada de alguna señal particular hará que el proceso deje este estado y ejecute una determinada secuencia de acciones.

En LED un estado se representa por medio de un símbolo de estado, que contiene el nombre del estado y está conectado a símbolos de entrada.

En la siguiente figura se presenta un ejemplo.

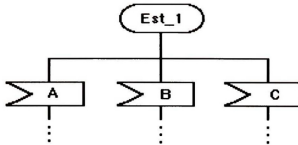
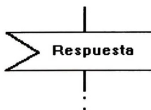


Diagrama de un estado conectado a tres símbolos de entrada

Señal de entrada

Un símbolo de entrada asociado a un estado X, implica que si la señal inscrita en el símbolo de entrada llega cuando el proceso se encuentra en X, habrá que interpretar la transición que sigue a dicho símbolo. Cuando una señal provoca la interpretación de una transición, deja de existir y se dice que ha sido consumida.

Su representación gráfica se construye mediante un símbolo de entrada que contiene uno o más identificadores de señal. La siguiente figura muestra un símbolo de señal de entrada en LED, con el identificador de señal "respuesta".



Ejemplo de símbolo de entrada

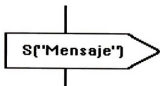
Señal de salida

Una salida es el envío de una señal de un proceso a otro proceso o así mismo. Desde el punto de vista del proceso emisor, una salida puede considerarse a menudo como una acción instantánea que, una vez completada,

Software de servicios

no ejercerá influencia en el proceso emisor, el cual no tendrá por qué conocer directamente el destino de la señal.

Su representación gráfica se construye por medio de un símbolo de salida, que contiene la especificación de señales, como se aprecia en la siguiente figura.

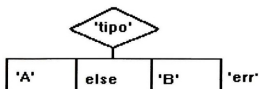


La salida envía un mensaje

Decisiones

Una decisión es una acción, dentro de una transición, en virtud de la cual se hace una pregunta respecto al valor de una expresión. El proceso continúa por uno de los dos o más trayectos que siguen a la decisión, de acuerdo con la respuesta.

La pregunta de una decisión puede ser una expresión o un texto informal. Las respuestas están representadas por uno o mas valores posibles, obtenidos por la evaluación de la expresión en la pregunta o por uno o más textos informales, como se muestra en la siguiente figura.



Ejemplo de decisión

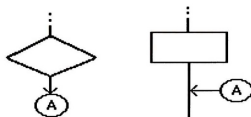
Uniones y conectores

Las uniones permiten transferir el control de un punto a otro en un cuerpo de procedimiento.

Gráficamente, las uniones corresponden a conectores (conectores de salida y conectores de entrada). Se utilizan para dividir los diagramas, debido a la falta

de espacio , o para evitar el cruce de las líneas de flujo que podrían hacer los diagramas algo confusos.

Cualquier línea de flujo puede ser interrumpida por un par de conectores asociados, considerándose que el flujo va del conector de salida al conector de entrada. Cada símbolo de conector contiene un nombre, los conectores asociados tienen el mismo nombre. Para cada nombre existe solamente un conector de entrada pero puede haber varios de salida. La siguiente figura muestra un ejemplo de conectores.



Referencias de conectores

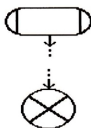
Procedimientos

Los procedimientos en LED son similares a los procedimientos en lenguajes de programación, donde sus propósitos son los de permitir la estructuración de un proceso en varios niveles de detalle, así como la compactación, permitiendo que una reunión compleja de ítems que pueden tomarse como aislados, se represente por uno solo, favoreciendo también su utilización repetidamente.

Gráficamente un procedimiento es similar a un proceso, con las siguientes diferencias:

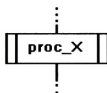
- Termina su interpretación con un retorno, en lugar de una parada
- El símbolo de arranque del procedimiento es ligeramente distinto al de un proceso
- Se puede usar la construcción de unión, pero nadamás para referir una etiqueta dentro del mismo procedimiento.

Los símbolos de arranque y retorno son los siguientes:



**Símbolos de arranque y
retorno de un procedimiento**

Las llamadas a procedimientos pueden producirse desde un proceso, o desde un procedimiento, mediante el siguiente símbolo.



Llamada al procedimiento proc_X

Bibliografía

[TAN 81] Tanenbaum, Andrew S.: "Computer Networks", Prentice Hall, USA., 1981.

[HUT88] Hutchinson, David: "Local Area Networks Architectures", Addison-Wesley, U.K., 1989.

[45RG86] Rodolfo Rosado, Manuel E. Guzmán, "DataCon 30: Un conmutador de datos de baja capacidad", "Informe técnico número 45, Serie amarilla, Dpto. de Ing. Eléctrica del Centro de Investigación y de Estudios Avanzados del IPN", México D.F., Mayo de 1986.

[24RG86], Rodolfo Rosado, Manuel E. Guzmán, "Red de conmutación de DataCon 30", "Informe técnico No. 24, Serie Azul, Dpto. de Ing. Eléctrica del Centro de Investigación y de Estudios Avanzados del IPN", México D.F., Mayo de 1986.

[24RG86], Rodolfo Rosado, Manuel E. Guzmán, "Memoria de Comando de DataCon 30", "Informe técnico No. 25, Serie Azul, Dpto. de Ing. Eléctrica del Centro de Investigación y de Estudios Avanzados del IPN", México D.F., Junio de 1986.

[24RG86], Rodolfo Rosado, Manuel E. Guzmán, "Señales de sincronía de DataCon 30", "Informe técnico No. 26, Serie Azul, Dpto. de Ing. Eléctrica del Centro de Investigación y de Estudios Avanzados del IPN", México D.F., Junio de 1986.

Andrés Vega, "Procesador de control y programa para un conmutador de datos", Tesis profesional, IPN Escuela Superior de Ingeniería Mecánica y Eléctrica, Febrero de 1988.

Andrés Vega, "Núcleo concurrente MULTITASK", Manual de Usuario, Dpto. de Ing. Eléctrica del Centro de Investigación y de Estudios Avanzados del IPN", México, D.F.

Anexo D a la recomendación Z.100, directrices para el usuario del LED, Comité Consultivo Internacional Telegráfico y Telefónico, Tomo X, Fascículo X.2, Libro Azul, Noviembre de 1988.



CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL I.P.N.

Ave. Instituto Politécnico Nacional 2508 Col. San Pedro Zacatenco
México, D.F., C.P. 07300

EL JURADO DESIGNADO POR LA SECCION DE COMPUTACION DEL DEPARTAMENTO
DE INGENIERIA ELECTRICA, APROBO EL DIA 17 DEL MES DE
AGOSTO DEL AÑO DE 1992 EL TRABAJO DE TESIS

SOFTWARE DE SERVICIOS PARA UN CONMUTADOR DE DATOS

DESARROLLADO POR EL

ALUMNO: GRACIELA ROMAN ALONSO

M en C. Andres Vega García

M en C. Jorge Buenabad Chávez

Dr. Sergio V. Chapa Vergara

CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL
INSTITUTO POLITECNICO NACIONAL

BIBLIOTECA DE INGENIERIA ELECTRICA
FECHA DE DEVOLUCION

El lector está obligado a devolver este libro
antes del vencimiento de préstamo señalado
por el último sello.

DEVOLUCION

AUTOR ROMAN ALONSO, G.

TITULO SOFTWARE DE SERVICIOS PARA UN CONMUTADOR DE DATOS

CLASIF.	XM	RGTR.	BI
	92.5		13148

[illegible]

