



**ESCUELA SUPERIOR POLITECNICA DEL LITORAL**  
**FACULTAD DE INGENIERIA EN ELECTRICIDAD**

## **Tópico de Redes de Area Local**

**Informe presentado**  
**Previo a la Obtención del Título de:**  
**INGENIERO EN ELECTRICIDAD**  
**Especialización ELECTRONICA**

**Perteneciente a:**

**Freddy Vizquete M. - Zenen Delgado M.**  
**Fernando Mórtola B. - Marco Villalba D.**

**Tema:**

**Análisis, Programación e Implementación de**  
**una Red de Area Local "SERIAL"**

**Profesor: JAIME PUENTE PEREZ**

**1993 - 1994**

**Guayaquil - Ecuador**

## TABLA DE CONTENIDO

INTRODUCCION:.....	4
<b>CAPITULO I</b>	
1.1.BP-LAN BOSQUEJO Y PROGRAMACIÓN.....	6
1.2.CARACTERÍSTICAS DISTINTIVAS DE BP-LAN.....	8
1.3.DESVENTAJAS DE BP-LAN EN LA COMUNICACIÓN :	10
1.4.REQUERIMIENTOS.....	10
1.5.SERVIDORES DEDICADOS VERSUS NO DEDICADOS....	11
1.5.1.CARACTERÍSTICAS DEL SERVER EN MODO DEDICADO.....	11
1.5.2.CARACTERÍSTICAS DEL SERVER EN MODO NO DEDICADO.....	12
<b>CAPITULO II</b>	
2.1.BOSQUEJO DEL BIOS DE BP-LAN.....	13
2.1.1.LA CARTA DE INTERRUPCIONES DEL BIOS DE BP-LAN.....	15
2.1.1.1.Nivel de Interrupción, Interrupciones de Control (INT 80 H)..	15
2.1.1.2.Nivel de Interrupción , Interrupciones Físicas ( INT 81 H )...	15
2.1.1.3.Nivel de Interrupción, Interrupciones de Enlace de Datos ( INT 82 H ).....	16
2.1.2.COMUNICACIÓN SERIAL.....	17
2.1.3.TRANSFERENCIA DE DATOS.....	21
<b>CAPITULO III</b>	
3.1.TIPOS DE PAQUETES (COMANDOS):.....	22
3.1.1.COMANDOS DE MENOR ESTADO.-.....	22
3.1.1.1.COMANDOS DE MÚLTIPLE ESTADO.-.	22
3.1.1.2.SOLICITUD.-.....	22
3.1.2.SOLICITUDES AL FILE SERVER:.....	23
3.1.2.1.SOLICITUD COMANDO DE EJECUCIÓN REMOTA:.....	24
3.1.2.2.SOLICITUD DE FUNCIONES DE BAJO NIVEL: 21	
3.1.2.3.SOLICITUD DE ACCESO AL SEMÁ- FORO.....	25

## CAPITULO IV

4.1.COMPARTICIÓN DE DISCOS EN RED.....	26
4.1.1.MODO SECTOR.....	26
4.1.2.PROGRAMAS TERMINATE AND STAY RESIDENT (TSR).....	27
4.1.2.1.LOS DEVICE DRIVERS.....	28
4.1.2.2.BPMOUNT.SYS.....	29
4.1.3.COMPARTIENDO DISPOSITIVOS DE RED PARALELO Y SERIE.....	31
4.1.4.SOLUCIÓN DE BP-LAN PARA PROBLEMAS DE IMPRESIÓN.....	33

## CAPITULO V

5.1.UTILITARIOS DE BP-LAN.....	34
5.1.1.Comandos de ejecución remota:.....	34
5.1.2.Comandos de Impresión remota:.....	35
5.1.3.Mostrando y modificando definiciones de discos:.....	36
5.1.4.Hora del servidor:.....	36
5.1.5.Ejecución del correo electrónico:....	37
5.1.5.1.S - Enviar mensaje:.....	37
5.1.5.2.V - Ver un mensaje:.....	38
5.1.5.3.Q - Salir y grabar mensaje:...	38
5.1.5.4.X - Salir sin grabar mensaje:.	39

## CAPITULO VI

6.1.UTILITARIO AUTOMÁTICO DE INSTALACIÓN BP-LAN.	40
6.1.1.REQUERIMIENTOS GENERALES:.....	40
6.1.2.REQUERIMIENTOS DEL CLIENTE:.....	40
6.1.3.REQUERIMIENTOS EN EL SERVIDOR:.....	41
6.1.4.INSTALACIÓN:.....	41
6.1.4.1.PREGUNTAS GENERALES.....	42
6.1.4.2.PREGUNTAS DEL PUERTO SERIAL...	42
6.1.4.3.PREGUNTAS DE DRIVE COMPARTIDOS	43
6.1.4.4.PREGUNTAS DE CORREO ELECTRÓ- NICO.....	44
6.1.5.REMOCIÓN DE LA RED.....	45

<b>CAPITULO VII</b>	
7.1. Análisis de los programas.....	46
7.1.1. BPBIOS.ASM:.....	47
7.1.2. BPBIOSHD.MOD.....	47
7.1.3. MISC.MOD:.....	48
7.1.4. LOWLEVEL.MOD:.....	49
7.1.5. FILESERV.MOD.....	52
7.1.6. CONSOLE.MOD.....	55
7.1.7. BPTIME.ASM.....	55
 <b>CAPITULO VIII</b>	
8.1. PROBLEMAS Y SOLUCIONES DE HARDWARE.....	59
8.1.1. CABLEADO.....	59
 CONCLUSIONES Y RECOMENDACIONES.....	
	64
 ANEXO A.....	
	67
RECIBIR UN PAQUETE.....	68
TRANSMITIR UN PAQUETE.....	69
ENTRADA/SALIDA DE ARCHIVOS LEER SECTOR.....	70
ENTRADA/SALIDA DE ARCHIVOS ESCRIBE SECTOR.....	71
IMPRESION EN RED (LADO DEL SERVIDOR).....	72
IMPRESION EN RED (LADO DEL USUARIO).....	73
COMANDO DE EJECUCION REMOTA.....	74
BPINST.EXE.....	75
BPSEMAPH.EXE.....	76
 ANEXO B.....	
	77
LISTADO DEL PROGRAMA.....	78



## INTRODUCCION:

Nuestro conocimiento se compone de un sin número de experiencias, las cuales han llegado hasta nosotros por diferentes vías. Algunas de estas experiencias nos han sido pasadas por nuestros mayores, otras han sido fruto de la simple observación. Pero ninguna otra experiencia esta tan arraigada en nuestro conocimiento como el de las propias experiencias vividas.

En efecto, no hay mejor manera de entender un problema complejo, que el enfrentarse cara a cara con él, y sortear los obstáculos que aparecen a nuestro paso, sacando al final de la lucha conclusiones, fruto de nuestras propias experiencias.

Esta es justamente la filosofía con la que esta desarrollado este proyecto de **comunicaciones entre PC's**, llamado BP-LAN.

Este trabajo netamente didáctico nos lleva de la mano, mostrándonos los problemas que están involucrados en las comunicaciones dentro del entorno de una Red de Área Local (LAN), a la vez que nos muestra las soluciones y las razones que condujeron a tomarlas. De esta manera, muchos conceptos aprendidos en clase tienen entonces una base mas firme quedando las ideas sólidas y completas.

En este caso en particular, se trata de una red serial, es decir que para su implementación no hace falta ninguna tarjeta de red especial. Basta con que los computadores involucrados en la misma tengan uno o más puertos seriales, los cuales ya son considerados equipo estándar en los computadores. El caso un tanto excepcional es el servidor, el cual es el único que requiere un puerto serial para cada terminal, de tal manera que al desear conectar mas de dos terminales, hará falta el uso de una tarjeta multipuerto.

El nombre muy sugestivo, (Planificación de una Red) nos indica que el proyecto ha sido desarrollado para entender como trabaja una LAN y cuáles son los factores a considerar en su diseño.

Esta también detalla como el cableado de la red y hace un estudio de las diferentes posibilidades existentes al respecto. Tópicos discutidos incluyen técnicas de programación, BIOS DE BP-LAN, Drivers de comunicación, y compartición de recursos.

La principal motivación para el desarrollo de este trabajo es el de que se nos introduce en conceptos y técnicas básicas de programas residentes de memoria, dispositivos instalables y hasta cierto punto conceptos de sistemas operativos. Además de lo expuesto, influenció notablemente la facilidad en la implementación de esta red.

Uno quizás uno de los objetivos primordiales de este proyecto es la de introducir al estudiante a la programación de la computadora personal bajo este novedoso concepto de "Grupo", que significa una **Red de Computadoras**.

## EL TÍTULO

### **III. BP-LAN BOSQUEJO Y PROGRAMACIÓN**

El proyecto BP-LAN va a consistir de cuatro diferentes tipos de programas :

Bloques que manejan dispositivos	TSRs
Archivos de sistemas de operación de red	Aplicaciones

Todos los programas de BP-LAN interactúan a través de llamadas al Sistema Básico de Entrada / Salida de BP-LAN ( **SEDE DE BP-LAN** ) .

La función y valor de BP-LAN pueden ser mejor entendidos considerando sus objetivos de diseño .

**VELOCIDAD:** BP-LAN transmite y recibe información en velocidades de hasta 115.200 baudios.

**MODULARIDAD:** Tanto el código fuente como archivos ejecutables para BP-LAN deben ser altamente modulares, simplificando futuras adecuaciones y modificaciones del código .

**CONFIGURACIÓN:** La versión actual de BP-LAN fue completamente re-diseñada. Toda la información de configuración ha sido estandarizada de modo que ninguna reensamblación es necesaria para configuraciones estándar . La información de la configuración está ahora pasada a los diversos programas de BP-LAN ya sea en la línea de comando o a través de tablas en **SEDE DE BP-LAN** .

**ACcesIBILIDAD AL PROGRAMADOR:** El código fuente de BP-LAN es altamente modular. Cada función tiene una interface bien definida con el resto del código, permitiendo al programador modificar cualquier característica de BP-LAN sin requerir un conocimiento íntimo del programa completo .

**EL USO DE MEMORIA:** El uso de memoria de BP-LAN para programas residente en memoria ha sido minimizado permitiendo al administrador del sistema cargar solamente las características consideradas necesarias .

**COMPARTICIÓN DE ARCHIVOS:** Cualquier dispositivo de bloque ( lectura / escritura ) en el Servidor puede ser accedido de cualquier nodo en la red .

**COMPARTICIÓN DE IMPRESORA:** Cualquier impresora en el Servidor puede ser accedida como si fuera una impresora local del cliente , además se le provee de un mecanismo para evitar acceso simultáneo por otros clientes .

**EJECUCIÓN DE COMANDO REMOTOS:** Si un cliente inicia la ejecución de un comando de DOS en el Servidor , éste no responde a cualesquier solicitudes de clientes mientras esté ejecutando el comando.

**SEMÁFOROS:** La versión actual de BP-LAN no permite el BLOQUEO DE FICHERO . Si dos NO-APLICACIONES de BP-LAN ( cualquier programa no específicamente diseñado para correr en una red BP-LAN ) intenta acceso de escritura simultáneamente a un archivo , es probable que se corrompa . Las APLICACIONES de BP-LAN pueden utilizar opcionalmente semáforos para evitar el acceso a escritura simultáneo a un archivo. Un Semáforo es una variable en el servidor que corresponde al estado de un dispositivo, archivo, o grupo de archivos compartidos .

**TOPOLOGÍA:** Es de topología Estrella , en donde una computadora central es tanto el conector ( HUB ) de red como el administrador de recurso único . En una configuración más compleja , más de un administrador de recursos puede existir.



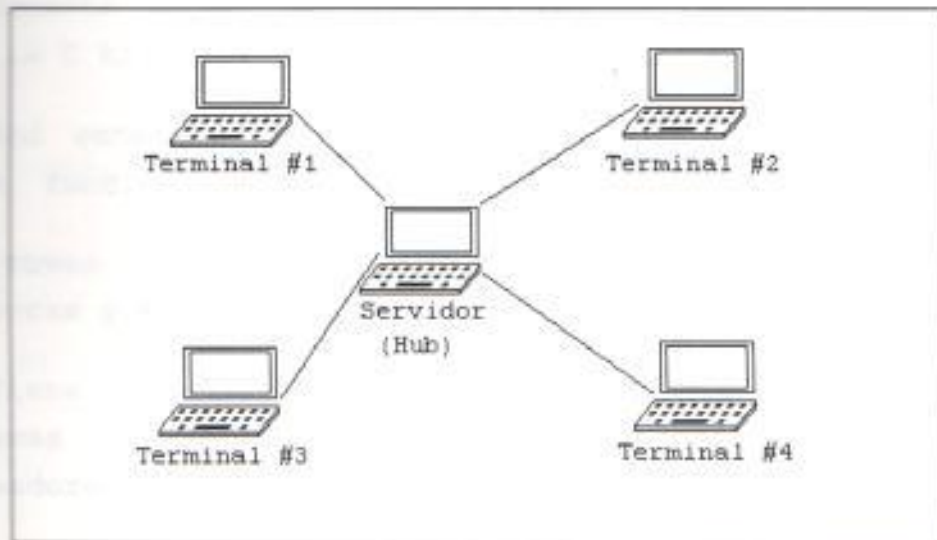


Fig. #1

## 1.2 CARACTERÍSTICAS DISTINTIVAS DE BP-LAN EN LA COMUNICACIÓN .

1. BP-LAN utiliza el RS-232C puerto de comunicación serie .
2. Un paquete de comunicaciones contiene muy pocos gastos ( overhead ) .
3. Soporta hasta 256 conectores ( dispositivos de red, incluyendo nodos de red, impresoras series, plotters, y modems ) .
4. El cliente puede acceder a un Servidor a través de un modem ( 2400 baudios o 9600 baudios )
5. La red puede ser anidada, así un Servidor en una red puede ser cliente en otra red .

## EN LA COMPARTICIÓN DE RECURSOS .

1. Soporta acceso de hasta 24 drives para compartición ( desde la C hasta la Z ) .
2. Red sensible , hasta aplicaciones , como utilitarios Norton, funcionan adecuadamente .
3. Provee a cada cliente acceso transparente a las impresoras y modem en el Servidor .
4. Tiene 256 semáforos disponibles para ser usados por programas de aplicaciones en dispositivos o archivos bloqueadores .
5. Un cliente puede iniciar la ejecución de un mando de DOS en el Servidor .

## MISCELÁNEOS .

1. El Servidor puede correr ya sea en modo dedicado (procesando comandos de red en primer plano ) o en modo no dedicado, es decir que puede ejecutar los mismos procesos de cualquier terminal.
2. Es provisto de un programa de instalación basado en ventanas.
3. Es provisto de un menú que maneja fácilmente correspondencia electrónica
4. El código fuente , está diseñado especialmente para simplificar la tarea de diseñar o modificar el código .

### 1.3. DESVENTAJAS DE BP-LAN EN LA COMUNICACIÓN :

1. A 115.200 baudios, la comunicación es casi 100 veces más lenta que los 10 Mhz , tasa de comunicación de las ultimas grandes redes .

### EN LA COMPARTICIÓN DE RECURSOS :

1. El bloqueo de ficheros no es soportado .
2. La seguridad de los archivos no es soportado .

### MISCELÁNEOS :

1. BP-LAN , como todas las Redes de Área Local, requieren al menos una persona con conocimientos de computación ,para instalar el hardware y configurar el software en una Red BP-LAN.

### 1.4. REQUERIMIENTOS

Al ser éste proyecto educativo , los requerimientos son pocos y de bajo costo, entre los que podemos nombrar los siguientes :

- La tarjeta adaptadora de red ( NIC ) es la tarjeta del puerto serial RS-232C.
- El servidor, que es sin duda el centro de la ESTRELLA deberá tener tantos puertos seriales RS-232C como estaciones deseara tener , de manera que al querer instalar más de dos terminales, hará falta el uso de una tarjeta multipuertos.
- En cuanto al cableado , se necesita un cable en cuyos extremos existan conectores DB-9 ó DB-25 según sean las computadoras que vayan a escoger como servidor y como estación. Estos conectores serán del tipo " hembra " .
- La conexión de los puertos RS-232C será del tipo MODEM NULO.

- Al ser una Red de baja performance por su característica educacional exclusivamente, está soportada por un software no muy complejo, que fácilmente se podrá manejar con las poderosas plataformas de hardware instaladas en las PC's en la actualidad, por lo que el hardware no es una limitante . Los requerimientos básicos de éste software son :

- 256K en RAM

- 256K espacio disponible en disco

- Un Driver y Disco Duro .

## 1.5. SERVIDORES DEDICADOS VERSUS NO DEDICADOS

BLUE PRINT LAN, puede trabajar en las dos formas, con servidor DEDICADO, y con servidor NO DEDICADO.

Un servidor DEDICADO significa que la computadora que está ejerciendo de servidor, sólo corre aplicaciones de red, es decir se concentra en atender a los usuarios de la red exclusivamente, procesando todos los requerimientos solicitados por ellos, y no puede atender un trabajo, como estación de la red, es decir no se puede accesar por medio del teclado a el, el servidor NO DEDICADO si lo permite, realizando solicitudes de red a la vez en el BACKGROUND , mientras actúa como estación en el FOREGROUND, esperando ser accesado por teclado.

### 1.5.1. CARACTERÍSTICAS DEL SERVIDOR EN MODO DEDICADO

- Procesa de una manera más rápida y confiable que uno dedicado, ya que no hay ninguna otra aplicación en competencia.
- Realiza en forma segura comandos de ejecución remota y de ingreso de archivo remoto.



- Es más costoso porque requiere de la compra de un computador extra, para tenerlo sólo como servidor.

#### 1.5.2. CARACTERÍSTICAS DEL SERVER EN MODO NO DEDICADO

- Es muy atractivo porque no requiere una computadora extra para usar como servidor, por lo tanto es económico.
- Tiene una calidad pobre y bajo rendimiento debido a que comparte tareas de estación de trabajo y de servidor.
- Sólo realiza en forma segura impresiones remotas, funciones de bajo nivel y semáforos, mientras que no son soportados o los realiza en forma insegura los comandos de ejecución remota e ingreso a archivo remoto.

## 2. CAPITULO

### **2.1. BOSQUEJO DEL BIOS DE BP-LAN.**

Todas las computadoras modernas vienen equipadas con un BIOS (sistema básico de entrada\salida) éste contiene una biblioteca de funciones de entrada\salida que pueden ser llamadas desde programas de aplicación . El BIOS esta almacenado generalmente, al menos en parte, en memoria ROM, hay EXTENSIONES al BIOS que pueden ser cargadas desde disco flexible o disco duro.

Las funciones de la biblioteca del BIOS son iniciadas a través de llamadas a interrupciones (instrucción INT) , por ejemplo INT 5 llama a una dirección que esta ubicada en la tabla 5 en ésta tabla están almacenadas los vectores de interrupción (el valor de las direcciones de las diferentes funciones), debido a esto podemos cambiar el código que desempeña una función simplemente cambiando la dirección en un vector de interrupción .

Ahora el objetivo es que BP-LAN provea una extensión al BIOS de una IBM PC, mediante un programa residente en un bloque reservado de memoria RAM (TSR) , que puede ser cargado dinámicamente, es decir que puede ser cargado en cualquier momento en una sesión de DOS . El propósito de un programa TSR es cargar funciones de remplazo o extensión al sistema de operación existente , de esta manera exteriorizamos las funciones de manipulación de recursos compartidos , logrando una integración de estas capacidades , al sistema de operación , entonces todas las aplicaciones pueden beneficiarse de estos recursos .

Por ejemplo si reemplazamos INT 17H, la función del BIOS para comunicar la impresora local conectada a la PC, con código que re dirige la salida deseada para el puerto paralelo local hacia un dispositivo en el Servidor, entonces podemos

utilizar esos dispositivos . Sin embargo, no es posible soportar dispositivos de, bloque, como drives duros compartidos, simplemente reemplazando llamadas al BIOS. En lugar de esto , es necesario utilizar dispositivo driver instalable , que es un programa que reside en un bloque reservado de RAM , diferenciándose con un TSR , en que debe ser cargado en tiempo de arranque del sistema ( Boot ) , Siendo así , éste dispositivo drivers tiene que ser especificado en el archivo CONFIG.SYS en el directorio principal del driver de arranque , cuyo formato es como sigue :

```
DEVICE = [PATH] FILENAME [/ PARAMETER(S)]
```

```
Ejemplo 1: DEVICE = C: \ DOS \ ANSI.SYS
```

```
Ejemplo 2: DEVICE = C: \BP-LAN \ BPMOUNT.SYS / C / O
```

Las extensiones provistas por BP-LAN al BIOS de IBM PC , son cargadas con el programa BPBIOS.COM , y las funciones pueden ser accedidas a través de INT 80 H hasta INT 82 H . La versión del BIOS del BP-LAN soporta solamente las dos capas más bajas del modelo de referencia ISO\OSI , LA CAPA FÍSICA que proporciona transmisión y recepción de bytes y LA CAPA DE ENLACE DE DATOS que es responsable de proporcionar detección y corrección de error . Todas las capas más altas son implementadas directamente en los programas de aplicaciones de BP-LAN .

Las interrupciones de capa física da funcionamiento a todo conector de entrada /salida . Un conector es cualquier dispositivo de hardware que puede desempeñar las siguientes funciones de E/S de la red :

1. Prueba si un byte ha sido recibido
2. Lectura de un byte

3. Prueba si un byte puede ser 4. Transmisión de un byte transmitido

La siguiente tabla documenta todas las funciones que son soportadas por el BIOS de BP-LAN .

### 2.1.1. LA CARTA DE INTERRUPCIONES DEL BIOS DE BP-LAN

#### 2.1.1.1. Nivel de Interrupción, Interrupciones de Control (INT 80 H)

INT 80H Función 00H- Obtiene número de conectores definidos

Llamar con : AH = 00H

Retorna : AL = Número de conectores definidos

INT 80H Función 04H- Instale conector en la siguiente posición disponible

Llamar con : AH = 04H

Retorna : Nada

INT 80H Función 05H - El punto del siguiente conector disponible

Llamar con : AH = 05H

Retorna : Nada

#### 2.1.1.2. Nivel de Interrupción , Interrupciones Físicas ( INT 81 H )

INT 81H Función 00H - Prueba si un byte ha sido recibido

Llamar con : AH = 00H

BL = Número de conector

Retorna : NZ Dato recibido

Z Dato no recibido

INT 81H Función 01H - Recibe un byte

Llamar con : AH = 01H

BL = Número de conector



Retorna : AL = Byte recibido

INT 81H Función 02H - Prueba si puede transmitir desde  
= conector

LLamar con : AH = 02H

BL = Número del conector

Retorna : NZ Si buffer de transmisión esta vacante

Z Si buffer de transmisión esta lleno

INT 81H Función 03H - Transmite un byte

LLamar con : AH = 03H

BL = Número del conector

AL = Byte a ser transmitido

Retorna : Nada

### 2.1.1.3. Nivel de Interrupción, Interrupciones de Enlace de Datos ( INT 82 H )

INT 82H Función 00H - Limpia el verificador de  
integridad de datos

LLamar con : AH = 00H

Retorna : Nada

INT 82H Función 01H - Calcula el verificador de  
integridad de datos

LLamar con : AH = 01H

AL = Byte a añadir al verificador de  
integridad de datos

Retorna : Nada

INT 82H Función 02H - Obtiene el verificador de  
integridad de datos

LLamar con : AH = 02H

Retorna : AX = Retorna el verificador de integridad  
de datos

INT 82H Función 03H - Transmite un paquete

LLamar con : AH = 03H

BL = Número del conector  
DS : DX = Dirección de transmisión del paquete

CX = Longitud del paquete a transmitir

Retorna : Nada

INT 82H Función 04H - Paquete Recibido

LLamar con : AH = 04H

BL = Número del conector

DS : DX = Dirección del paquete recibido

Retorna : C X = Longitud del paquete recibido.

### 2.1.2. COMUNICACIÓN SERIAL

En la red BP-LAN los nodos usaran el adaptador de comunicaciones asincrónicas RS 232-C para compartir información . La tabla siguiente ilustra la interfaz normalizada RS 232-C diseñada por IBM.

Terminal	Nombre Señal	Comentarios
1	Tierra-chasis	Unidades conectadas a tierra
2	Datos transmitidos (TD)	Salida de datos de la interfaz
3	Datos recibidos (RD)	Entrada de datos a la interfaz
4	Petición para enviar (RTS)	Define condición de dato, listo para envío
5	Limpia para enviar (CTS)	Señal de entrada que permite la transmisión
6	Datos colocados listos (DSR)	Entrada que notifica estado de disponibilidad

7	Señal de tierra	Señal común de tierra única
8	Detector portador de datos (CD)	Indica presencia de datos
20	Terminal de datos lista (DTR)	Señal de salida que indica disponible para transmitir
21	Indicador de alerta	Se lleva a cabo , debido a timbre de teléfono

En la IBM-PC ,el puerto RS 232-C es controlado con el INS8250 Semiconductor Nacional Transmisor Receptor Asincrónico Universal (UART) . Este chip convierte los datos paralelos que llegan del sistema del microprocesador a datos en serie . De manera similar convierte los datos en serie que llegan desde el RS 232-C a datos en paralelo que salen hacia el sistema microprocesador.

El INS8250 contiene 11 registros de 8 bits, de los cuales se pueden acceder a 10 de ellos por medio de mapeo de memoria. El BIOS de cada computadora contiene una tabla, en la cual se puede determinar la dirección base de cada puerto. Esto se lo realiza leyendo el contenido que se encuentra en 0040:0000. Aquí la primera palabra contiene la dirección base del puerto COM1, la segunda la del puerto COM2 y así sucesivamente, dependiendo de la cantidad de puertos instalados.

La velocidad de transmisión y recepción de datos ( en bits por segundo ) se la determina dividiendo la frecuencia del reloj de entrada al UART (1.8432 Mhz) para un valor de 16 bits leídos de las posiciones MSB y LSB del registro "Divisor Latch" y luego dividiendo para 16.

Para programar el UART a una velocidad determinada, se efectúa lo contrario a lo explicado anteriormente, es decir, se almacena un valor determinado en ese registro.

Estos son los pasos que efectúa el BIOS del computador cuando se lo invoca a inicializar al UART ( por lo general esto ocurre al encender el computador ) a través de la rutina de servicio del BIOS para comunicaciones asincrónicas, INT 14H, que contiene cuatro funciones :

1. Inicializa el puerto de comandos (AH=0)
2. Envía un carácter (en AL) fuera del puerto (AH=1)
3. Envía un carácter (establecido en AL) desde el puerto (AH=2)
4. Retorna el estado del puerto en AL (AH=3)

En el INS8250 la información del formato de datos se lo obtiene de los últimos 6 bits del registro "Line Control". Las designaciones del formato de paridad se hallan en los bits 3 al 5, el número de bits de parada se hallan en el bit #2, y un código correspondiente al número de bits de datos se encuentra en los bits 0 y 1. El bit mas significativo de este registro se lo conoce como "Divisor Latch Access Bit" (DLAB). Este es usado como un registro adicional para compartir dos puertos. Si el DLAB es 0 , entonces los puertos y la dirección base+1 corresponden a los registros de recepción / transmisión y al interrupt enable . Si DLAB es 1 , entonces los puertos están conectados lógicamente a los MSB y LSB del divisor latch.

El INS8250 UART solamente soporta velocidades de hasta 115.200 baudios , mientras que las tarjetas adaptadoras de LAN soportan tasa de comunicación de 10 Mbps(10.000.000 baudios).



Más que una baja velocidad, es la carga añadida al procesador central CPU. Las tarjetas adaptadoras de LANs transmiten y reciben paquetes completos de información independientemente del CPU, mientras el INS8250 UART puede solamente manipular un simple byte a la vez.

La información tiene que ser formateada en paquetes de comunicación antes de que sea pasado entre dos computadoras. Tales paquetes consisten generalmente de un encabezado descriptivo, un campo de información, y un campo de chequeo de error.

Antes de transmitir un paquete en BP-LAN, la computadora transmisora tiene que tener la entera atención de la computadora receptora. La computadora transmisora esta transmitiendo repetidamente el byte 01H, conocido como encabezado de inicio, o SOH. Los caracteres van a la computadora receptora hasta que la computadora receptora reconoce el encabezado de inicio con un byte 06H, conocido como un reconocimiento de caracter o ACK. La computadora transmisora escucha el eco del byte de reconocimiento, y transmite el paquete de comunicación.

La conexión entre los puertos de comunicación asincrónico RS-232C de las distintas computadoras se realiza mediante el uso de un cable de "modem nulo" (null modem) que se lo debe construir de la siguiente manera:

Una línea de  
transmisión  
de datos  
de control  
de flujo  
de control  
de flujo

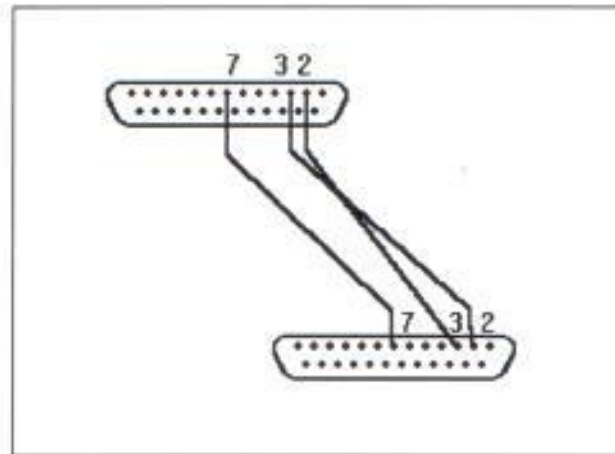


Fig. #2

Como se puede apreciar en el gráfico, se interconectan los pines de transmisión de un lado con los de recepción del otro lado y viceversa. Además de estos, se deberá conectar también entre ellos los pines que corresponden a la tierra de la señal.

### 2.1.3. TRANSFERENCIA DE DATOS

Toda la transmisión de datos se la realiza a través de un modelo determinado de datos llamados paquetes. Estos paquetes son totalmente transparentes al usuario, ya que los arma cada computador antes de transmitirlos, y los interpreta cada computador luego de haberlos recibido. Los paquetes no son más que una cadena de datos, la cual lleva un formato preestablecido y bien definido, para facilitar de esta manera los procesos de recepción y transmisión. Es así como, con solo interpretar la cabecera del paquete (identificador) se podrá identificar el proceso que se desea realizar, y se podrá que tipo de datos es el que le sigue.

## 3. CAPITULO

### 3.1. TIPOS DE PAQUETES (COMANDOS):

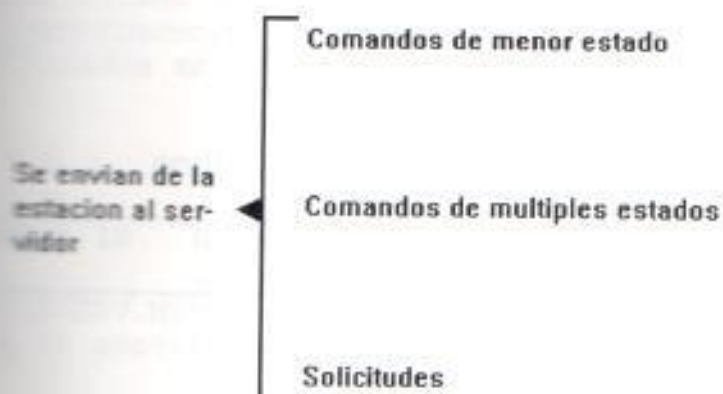


Fig. #3

#### 3.1.1. COMANDOS DE MENOR ESTADO.-

Cuando el usuario de la estación inicializa este tipo de comando, lo hace en el server, no espera retorno de ninguna señal de reconocimiento, es decir que hace uso de 1 sólo paquete, todo el proceso es realizado por el server, mientras que la estación puede encargarse de otras labores, un ejemplo de este tipo de comando, son los comandos de ejecución remota. ejm: llamada a impresora, llamada a las señales de I/O, llamada a comandos del DOS en el server.

#### 3.1.1.1. COMANDOS DE MÚLTIPLE ESTADO.-

Un ejemplo de este comando son los semáforos, ya que cuando ellos actúan, concentran la actividad de la red en el usuario que le solicitó primero, mientras que los demás esperan en la cola, como ejemplo si estamos realizando un RECORD LOCKING, para extraer un dato de un archivo, y en ese mismo instante otro usuario está cambiando los datos, actualizándolos, vamos a tener una lectura errónea, así que es conveniente que se bloquee a nosotros hasta que se complete la modificación.

### 3.1.1.2. SOLICITUD.-

A diferencia de los anteriores, retorna un estatus de información u otro dato luego de realizar su tarea, algunos siempre requieren el cambio de 2 paquetes por comando, por lo tanto son menos eficientes que los comandos de múltiple y menor estado que requieren un sólo paquete.

A continuación presentamos los paquetes de comandos utilizados en BP-LAN que utilizan solicitudes:

### 3.1.2. SOLICITUDES AL FILE SERVER:

Packet ID	Drive #	Sector #
3	1 Byte	2 Bytes

(FILESERV.MOD)

Lee un sector determinado en el drive especificado.

Packet ID	Drive #	Sector #	Sector de datos
4	1 Byte	2 Bytes	sobre 1024 Bytes

(FILESERV.MOD)

Escribe un sector de datos para el drive especificado y sector en el server.

Packet ID
5

(FILESERV.MOD)

Indica que el comando escribir en el sector terminó.

Packet ID	Drive #
6	1 Byte

(FILESERV.MOD)

Chequea si media en drive especificado del server tuvo que ser modificada.

### 3.1.2.1. SOLICITUD COMANDO DE EJECUCIÓN REMOTA:

Packet ID	Comando de DOS terminado con retorno
7	Sobre los 512 Bytes

(CHILD.MOD)

Ejecuta comando especificado en el DOS en el server.



Packet ID	Device #	Character
F	1 Byte	1 Byte

(PRINSERV.MOD)

Ejecuta int 17H, impresora de linea llama al BIOS en el server.

Packet ID	Función	Device #	Character
S	1 Byte	1 Byte	0 a 1 Bytes

(PRINSERV.MOD)

Ejecuta int 14H, I/O serial llama al BIOS en el server.

### 3.1.2.2. SOLICITUD DE FUNCIONES DE BAJO NIVEL:

Packet ID	Dirección Offset	Dirección segmento	Longitud
E	2 Bytes	2 Bytes	2 Bytes

(LOWLEVEL.MOD)

LEE EL CONTENIDO DE MEMORIA DE UN RANGO DE DIRECCIONES ESPECIFICADAS EN EL SERVER.

Packet ID	Offset	Segment	Length	Data
w	2 Bytes	2 Bytes	2 Bytes	> 2 Bytes

(LOWLEVEL.MOD)

ESCRIBE DATOS PARA UN RANGO DE DIRECCIÓN ESPECIFICADA EN EL SERVER.

Packet ID	I/O Direcciones
i	2 Bytes

(LOWLEVEL.MOD)

LEE (ENTRADA) BYTE DESDE DIRECCIONES ESPECIFICADAS DE I/O EN EL SERVER.

Packet ID	I/O Direcciones	Datos
e	2 Bytes	1 Byte

(LOWLEVEL.MOD)

ESCRIBE

(SALIDA) BYTE LA DIRECCIÓN I/O ESPECIFICADA EN EL SERVER.

Packet ID	Código de maquina a ser ejecutado
a	Sobre los 512 Bytes

(LOWLEVEL.MOD)

EJECUTA INSTRUCCIONES DE CÓDIGO PASADAS EN EL PAQUETE PREVIO EN EL SERVER.

Packet ID	Offset	Segmento	Longitud	Llenar	Caracter
1	2 Bytes	2 Bytes	2 Bytes	1 Byte	

(LOWLEVEL.MOD)

LENA UN RANGO DE DIRECCIONES ESPECIFICADAS EN EL SERVER CON UN BYTE ESPECIFICADO

Packet ID	Offset1	Seg. 1	Offset2	Seg. 2	Longitud
2	2 Bytes	2 Bytes	2 Bytes	2 Bytes	2 Bytes

(LOWLEVEL.MOD)

MUEVE UN RANGO DE DIRECCIONES ESPECIFICADAS EN MEMORIA EN EL SERVER.

### 3.1.2.3. SOLICITUD DE ACCESO AL SEMÁFORO:

Packet ID	Numero del Semáforo
3	1 Byte

(SEMAPHOR.MOD)

CIERRA EL SEMÁFORO ESPECIFICADO EN EL SERVER, SOLAMENTE SI ESTE ES CORRIENTEMENTE NO APROPIADO.

Packet ID	Numero del Semáforo
3	1 Byte

(SEMAPHOR.MOD)

ABRE EL SEMÁFORO ESPECIFICADO EN SERVER, SOLAMENTE SI ESTE ES APROPIADO PARA EL CLIENTE.

Packet ID	Numero del Semáforo
3	1 Byte

(SEMAPHOR.MOD)

COPIA EL SEMÁFORO ESPECIFICADO EN EL SERVER SIN HACER CASO DE SU ESTADO CORRIENTE.

En muchos de los programas que componen el software de BP-  
LMS, es común el uso de las letras de los paquetes de  
identificación, he aquí su gran importancia en conocerlos.

## 4. CAPITULO

### **4.1. COMPARTICIÓN DE DISCOS EN RED**

Los discos de red compartidos pueden ser accedidos en dos modos:

- MODO SECTOR
- MODO ARCHIVO

En modo sector la computadora del usuario realiza todo el manejo de funciones de disco que implican el drive ingresado.

En modo archivo la computadora del server es responsable de manejo de las funciones de disco.

En el software de BP-LAN se utiliza el modo sector para acceder a discos de red compartidos.

#### **4.1.1. MODO SECTOR**

Los datos están almacenados en sectores, en los dispositivos de acceso aleatorio tales como, floppy drives, hard drives y drives RAM.

Un sector es el bloque de datos direccionable más pequeño en un disk drive. A fin de acceder datos en modo sector, el sistema operativo debe conocer, la organización física del medio de almacenaje (por ejemplo, ¿ cuántas cabezas ?, cilindros y sectores y ¿ cuántos bytes por sector ?).

El dato es escrito y leído un sector a la vez. Si un programa de aplicación requiere un sólo byte desde un archivo, el sistema operativo, debe determinar, ¿ cuál sector contiene aquel byte y luego leer el sector entero.

Compartir un disco de red en modo sector requiere una cantidad muy pequeña de código para implementar. El modo

sector se accesa a menudo de una manera más eficiente que el modo archivo, en terminos del número de paquetes que deben ser transmitidos para igual cantidad de datos.

En modo **archivo** , los datos son a menudo transmitidos, en pequeños paquetes, a menudo conteniendo un sólo caracter, escrita de eso requieren algunos bytes para el header del paquete e información del chequeo del error.

En modo **sector**, el dato es transmitido en paquetes más grandes, así genera menos transmisión de gasto u OVERHEAD. Los discos en modo sector son compatibles con casi todo tipo de software, incluyendo utilitarios, defragmentación del disco duro, ordenamiento de directorios, ordenamiento de directorios, DOS, programa CHKDSK (chequeo de disco) y utilitarios NORTON.

#### 4.1.2. PROGRAMAS "TERMINATE AND STAY RESIDENT" (TSR)

Los programas capaces de extender el sistema operativo, es decir sus funciones ya existentes; como su nombre lo indica una vez que se corren quedan residentes en un bloque reservado de memoria de la RAM del computador.

Estos programas fueron creados con la finalidad de suplir deficiencias tales como las que se presentan al usar un EMULADOR DE TERMINAL ,el que suministra un usuario con acceso completo a los recursos de un computador remoto, pero una vez finalizada la aplicación,todo el acceso a estos recursos acaba y no pueden beneficiarse las demás aplicaciones de tales recursos.

Los TSR reemplazan o complementan las llamadas al DOS o BIOS con nuestras propias llamadas a función acostumbradas,



entonces se está añadiendo soporte de recursos compartidos donde es necesario.

Como ejemplo si reemplazamos int 17H (función de imprimir caracter del BIOS, con un código que redireccione la salida destinada para los puertos locales paralelos a un dispositivo en el server, entonces podemos utilizar aquel dispositivo.

Se debe tener mucho cuidado de añadir algún programa TSR nuevo a BP-LAN debido a que esta fue creada para un determinado número de programas TSR, tales como aquellos con los que BP-LAN se conecta por sí sola a algunos vectores de interrupción tales como int 14H (RS232C I/O), int 17H (salida a impresora de línea) int 21H (funciones del DOS) e int 28H (DOS desocupado), algún nuevo TSR podría reemplazar una o más de estas interrupciones.

**BLOCK DEVICE** .- Es un dispositivo que corre a una letra drive, en el server o en el computador hub, puede ser accesado desde algún nodo en la red, un dispositivo puede ser añadido como un volumen de lectura/escritura o lectura solamente.

#### 4.1.2.1. LOS DEVICE DRIVERS

Los "installable device drivers" son programas que tienen la misión de reemplazar a los TSR's, ya que ellos además de entender rápido la funcionalidad del DOS soportan los **block devices**, cosa que no lo realizan los TSR's.

No se puede compartir un hard drive simplemente con una llamada al BIOS como lo harían los TSR's.

Reside en un bloque reservado de memoria de la RAM, al igual que los TSR's. Es cargado en el momento del booteo, mientras que los TSR's lo hacen en una sesión de DOS.

El device driver debe ser especificado en el drive de booteo, por una entrada especifica en el CONFIG.SYS, el siguiente es su formato.

```
DEVICE = [ path ] filename [ /parameter (s) ]
```

Ejemplo 1 : DEVICE = C : \ DOS \ ANSI.SYS

Ejemplo 2 : DEVICE = C : \ BP-LAN \ BPMOUNT.SYS

#### 4.1.2.2. BPMOUNT.SYS

Es un installable MSDOS **device driver** que permite acceder a un sector desde una estación a un drive compartido especifico en el server. Este **block device driver** es instalado en la cabeza del DOS device driver enlazando listas para una entrada en el archivo CONFIG.SYS file, colocando así al block device como el próximo drive disponible.

El código fuente para BPMOUNT.SYS reside en 4 archivos:

BPMOUNT.ASM (es el principal archivo fuente).

REDDESK.MOD (es un archivo encabezador conteniendo varias definiciones red relacionada).

UTIL.MOD (es un módulo conteniendo varios macros en lenguaje ensamblador y subrutinas de propósito general).

CONSOLE.MOD (contiene rutinas, las cuales manejan la salida por pantalla y entradas por teclado en el cliente).

El **BPMOUNT.SYS** se ejecuta de la siguiente forma:

```
MS-DOS BPMOUNT;
```

```
LINK BPMOUNT;
```

```
DIRBIN BPMOUNT BPMOUNT.SYS
```

El **BPMOUNT.SYS** se encarga de configurar el primer paso de una estación de BP-LAN como es el de añadir todos los drives compartidos. Este debe ser cargado usando la palabra clave **DEVICE**. La sintaxis para lograr esto es :

```
DEVICE = BPMOUNT.SYS / shared_drive / socket_number /  
write_protect_flag / max_sector_length
```

**shared\_drive** es la letra que el server usa para hacer referencia a el drive, por ejemplo si la estación añade drive C desde el server, entonces en el formato va **C**.

**socket\_number** como su nombre lo indica en este espacio va el número del socket que puede ser un número de **0** a **255** y que se coloca de acuerdo al orden en que fue definido, por ejemplo socket 0 el primero, socket 1, el segundo, etc.

**write\_protect\_flag**, en este espacio va **0** para indicar lectura/escritura y **1** para lectura solamente.

**max\_sector\_length** se refiere al número máximo de bytes por sector para este drive, usualmente **512**, otras versiones de **MS-DOS**, versión 3.30 en particular usan longitudes de 1024 a 2048 bytes, este valor no debe ser excedido en los local hard drives.

### 4.1.3. COMPARTIENDO DISPOSITIVOS DE RED PARALELO Y SERIE

BP-LAN tiene incorporado en su sistema, las opciones de compartir los dispositivos ya sean paralelos o seriales. La idea de compartir uno de estos dispositivos, es la de que teniendo solo uno conectado en la red, cualquier usuario desde cualquier estación lo pueda utilizar. Generalmente se asocia a dispositivos paralelos las impresoras, y a dispositivos seriales los modems o las impresoras seriales. Si se refiere a compartir dispositivos seriales, estos pueden estar configurados como COM1, COM2, COM3 o COM4, y la cooperación se la realiza mediante el uso del BPCOMDEF, que es un programa residente de memoria que intercepta todas las interrupciones 14H de DOS, que son las que corresponden a los dispositivos seriales.

En cuanto a dispositivos paralelos, nos referimos a impresoras. Básicamente el proceso que realiza esta parte es idéntico al de los dispositivos seriales, con la diferencia que aquí lo que se intercepta son la interrupciones 17H de DOS, que son las que están relacionadas con la impresión.

Al igual que en todos los procesos de BP-LAN, aquí también se hace uso de los paquetes, y para el caso de por ejemplo una impresión remota, se enviara un paquete con el siguiente formato:

Packet ID	Device #	Character
0	1 Byte	1 Byte

Se puede usar este paquete para imprimir un caracter, pero se presentan ciertos problemas, como cuando el dispositivo está



agotado o esta fuera de papel, entonces entra a un estado que se llama **bit bucket** que significa irremediablemente perdido. Para solucionar esto, vamos a tener un paquete llamado RETURN SERVER, cada vez que la estación desea imprimir un caracter en el server, este construye y transmite el paquete especificado anteriormente, y luego espera por un paquete estatus, el cual indica el éxito o falla de la tarea de impresión.

Despues sobreviene otro problema, que sucede si dos usuarios intentan imprimir simultáneamente, los caracteres se imprimen en el servidor en el orden en que llegan si son enviados simultáneamente, la salida será un revoltijo de los dos.

Una manera para prevenir este revoltijo es permitir el acceso exclusivo de impresión para el primero de los usuarios, y que despues libere la impresora para que el segundo usuario imprima, esto se logra sensando el marcador de fin de archivo, pero por lo general la información que se imprime es información captada al vuelo, por lo tanto carente de marcador de fin de archivo, y entonces o no se imprime por completo o nunca se libera la impresora, algún mecanismo debe liberar la impresora cuando el usuario está obviamente inactivo.

Una solución es controlar el tiempo de trabajo de la impresora, si se la detecta inactiva, el server puede liberarla sin problema, si se realizan impresiones demasiado largas, el server puede interrumpir la impresión con cierto retardo, para que no monopolice la red y luego volverla a activar.

#### 4.1.4. SOLUCIÓN DE BP-LAN PARA PROBLEMAS DE IMPRESIÓN

En TSR BPLPTDEF.COM en el usuario ,recibe las int 17H, el IBM PCBIOS llama a imprimir un caracter para una impresora paralela, construye el paquete de impresión, y espera por un paquete estatus enviado por el server , este paquete es creado por un módulo de BPSERVER.ASM llamado PRINSERV.MOD. Si la impresora es apropiada para otro usuario, el usuario corriente espera 2 seg y luego reinicia, para asignar a otro usuario, el server detecta impresora inactiva por 30 seg, y luego asigna a otro usuario.

En caso de impresiones remotas seriales, que usan los puertos de comunicación serial predefinidos (COM 1, COM 2, COM 3, O COM 4),se ejecutara el programa BPCOMDEF.COM que se encarga de elaborar el paquete necesario conteniendo toda la información de int 14H.

Esta versión de BP-LAN no soporta dispositivo de cierre (bloqueo) para dispositivos seriales accesados, el uso de semáforos para prevenir acceso simultáneo para usuarios múltiples es recomendado.

En el lado del server PRINSERV.MOD ,se encarga de recibir las solicitudes de impresión remota y procesarlas para seleccionar la más indicada y luego mandar el paquete de estatus.

## 5. CAPITULO

### 5.1. UTILITARIOS DE BP-LAN

#### 5.1.1. Comandos de ejecución remota:

Determinadas funciones son altamente ineficientes al ser ejecutadas en una red, lo cual se puede ilustrar en el siguiente ejemplo:

Un cliente trabajando en un terminal desea copiar un archivo largo de un lado del disco de red a otro lado en el mismo disco. Esto implica que el archivo debe pasar inicialmente del servidor al terminal, para luego regresar del terminal al servidor y almacenarlo. Como se aprecia claramente, este proceso involucra un gran trafico innecesario en la red.

Para eliminar problemas de este tipo, se han implementado los comandos de ejecución remota. Este proceso es mucho mas sencillo, ya que el cliente envia solo la información del nombre del archivo y el lugar donde deberá ser copiado. Luego de esto, la estación ya no se involucra en el proceso, ya que ahora el SERVIDOR hará todo el proceso de copia directamente, eliminando de esta manera el trafico innecesario.

Este tipo de ejecuciones se la realiza con el programa BPREMOTE que lleva los siguientes parámetros:

```
BPREMOTE /numero_socket/comando_DOS
```

```
Por ejemplo: BPREMOTE /0 /COPY C:\WP51\LISTADO.TXT C:\
```

En este ejemplo se indicara el servidor que deberá copiar el archivo LISTADO.TXT que se encuentra en el subdirectorío WP51 al directorío raíz.

El parámetro numero\_socket se refiere al numero de socket que conecta al terminal con el servidor.

### 5.1.2. Comandos de Impresión remota:

En los capítulos anteriores se explico el porque de la necesidad de poder compartir impresoras remotas seriales o paralelas.

Para compartir una impresora remota se usara el siguiente comando:

```
SLPTDEF /numero_socket /dispositivo_cliente /dispositivo_servidor
```

Notese que el DISPOSITIVO para LPTn = n - 1

De tal manera que si deseamos imprimir los datos que deberían salir en el puerto LPT1 en el terminal, en el puerto LPT2 del servidor, y la comunicación se realiza a través del numero\_socket = 3, deberíamos ejecutar el comando:

```
SLPTDEF /3 /0 /1
```

Para impresoras seriales se deberá ejecutar el comando SPCOMDEF con parámetros idénticos al caso anterior.

Notese que el DISPOSITIVO para COMn = n - 1

En el mismo ejemplo anterior, pero en lugar de puertos paralelos usaremos puertos seriales, la notación será:

```
SPCOMDEF /3 /0 /1
```



### 3.1.3. Mostrando y modificando definiciones de discos:

Los discos de BP-LAN son especificados en el CONFIG.SYS a la hora de arrancar el computador. Pero estos dispositivos pueden ser modificados para una sesión en particular mediante el uso de BPDIVES. Este comando permite cambiar el acceso a los discos, el acceso a escritura y el numero de socket con el cual están conectados cliente y servidor.

```
BPDIVES /disco_cliente /disco_server /derechos_acceso  
/numero_socket
```

Disco\_cliente se refiere a la letra del disco virtual en el terminal.

Disco\_server se refiere a la letra del disco del server con la cual se hace referencia a dicho disco

Derechos\_acceso indica si en el disco del servidor solo se puede leer o hacer el proceso completo de lectura-escritura.

### 3.1.4. Hora del servidor:

La hora del servidor se puede leer a cualquier momento desde cualquier terminal. Esto se lo usa para sincronizar todos los relojes a la misma hora del servidor.

```
NETIME /numero_socket
```

### Correo electrónico:

El correo electrónico es un mecanismo para intercambiar datos en forma de texto entre dos usuarios de una red.

Estos datos son convertidos a mensajes de correo, mediante la anteposición de la información tanto del que envía, cuanto la del que recibe.

### 5.1.5. Ejecución del correo electrónico:

BP-LAN utiliza las siguientes variables de ambiente de DOS: BPDIR, BPEDITOR, BPSMAILDIR, BPUSER y COMSPEC. BPDIR es el directorio donde se encuentran los archivos ejecutables del correo electrónico. BPEDITOR es la ruta de acceso completa del editor de texto definido por el usuario. BPSMAILDIR es el directorio en el server, en el cual se almacenaran los distintos mensajes. BPUSER es el nombre del usuario, tanto para el que envía, como para el que recibe mensajes. COMSPEC es la ruta completa de acceso al COMAND.COM. En caso de haber una de estas variables no definidas, el programa pedirá al usuario que las defina.

Ejemplo:

```
SET BPSMAIL=D:\BPSMAIL (Especifica como directorio el BPSMAIL
en el disco D)
SET BPEDITOR=C:\DOS\EDLIN.COM (Especifica como editor al
EDLIN que esta en el disco C y en el directorio DOS)
```

El correo electrónico se ejecuta mediante el comando BPSMAIL desde el DOS. En caso de no tener correo el usuario, se mostrara un mensaje indicando ese hecho; Caso contrario se mostraran todos los correos que el usuario haya recibido. Luego de haber leído algún mensaje, se mostrara un menú de cuatro opciones: S - enviar un mensaje; V - leer un mensaje; G - salir y grabar mensaje; X - salir sin grabar mensaje.

#### 5.1.5.1. S - Enviar mensaje:

Al oprimir la "S" el programa permitirá crear un archivo de mensaje y enviarlo a algún usuario. Inicialmente se pedirá el nombre del usuario y una breve descripción del mensaje.

Seguido se pedirá especificar el origen del ingreso del mensaje, esto puede ser: K - para escribir el mensaje; E - para ejecutar el editor de línea especificado y F - para un archivo ya existente.

#### **5.1.5.2. V - Ver un mensaje:**

Esta opción permitirá leer el contenido de un mensaje. Los mensajes son mostrados por partes de 16 líneas a la vez. Oprimiendo la barra espaciadora se mostraran las siguientes 16 líneas. Con la tecla "T" se podrá regresar al inicio del mensaje.

Con la flecha hacia abajo seleccionaremos el siguiente mensaje y con la flecha hacia arriba seleccionaremos el mensaje anterior.

Hay la opción de almacenar algún mensaje específico como archivo de tipo texto. Esto se logra oprimiendo la tecla "F" y luego habrá que ingresar el nombre que deseamos dar al archivo.

Si se desee obtener una copia impresa del mensaje, deberemos oprimir la "P", y se imprimirá en el puerto LPT1.

La opción de borrado de mensajes se la realiza con la tecla "D". Pero esto solamente le pone una marca al mensaje, y este será borrado al salir del programa. En caso de algún arrepentimiento, se podrá desmarcar el mensaje mediante la tecla "U".

#### **5.1.5.3. Q - Salir y grabar mensaje:**

Esta opción sirve para terminar la ejecución del programa y guardar todos los mensajes leídos.

#### 5.1.5.4. X - Salir sin grabar mensaje:

Presionando "X" terminaremos con la ejecución del programa sin almacenar o actualizar los cambios realizados. Tampoco serán borrados los archivos marcados para tal efecto.



## 6. CAPITULO

### **6.1. UTILITARIO AUTOMÁTICO DE INSTALACIÓN BP-LAN**

Para una fácil instalación de la LAN SERIAL, es necesario correr el programa automático de instalación BPINST.EXE, el cual nos provee una interface muy amigable con el usuario, que en este caso es el administrador de la Red.

Este programa, cuyo código fuente está escrito en TurboPascal 5.0, de una manera interactiva y a través de preguntas sencillas modifica el CONFIG.SYS y AUTOEXEC.BAT de la computadora apropiadamente.

#### **6.1.1. REQUERIMIENTOS GENERALES:**

- Un IBM PC o compatible
- Como mínimo 256K en RAM
- 256K disponible de espacio en disco
- Dos Drives o un drive y un Disco Duro.
- El disco de instalación conteniendo los archivos:
  - BPINST.EXE
  - BPBIOS.COM
  - BPSERIAL.COM
  - BPMOUNT.SYS
  - BPMAIL.EXE
- Impresora Opcional.

#### **6.1.2. REQUERIMIENTOS DEL CLIENTE:**

-Tener disponible un puerto serial RS-232C para cada servidor.

- Por lo menos 1 drive
- Disco duro opcional
- 256k en RAM

#### 6.1.3. REQUERIMIENTOS EN EL SERVIDOR:

- Tener disponible un puerto serial RS-232C para cada cliente.
- Por lo menos 1 drive
- Disco duro opcional, pero recomendado.
- 256K en RAM
- Impresora opcional pero recomendada.

#### 6.1.4. INSTALACIÓN:

El programa de instalación automática BPINST.EXE puede correrse desde cualquier disco de instalación previamente preparado con todos los archivos ejecutables o por lo menos los que se indican inicialmente. Esto es especialmente recomendado si no se va a usar frecuentemente la RED.

Si el uso de la red es muy frecuente, se recomienda bootear desde el disco duro.

Todo el procedimiento de instalación se lo realiza a través de ventanas:

- Ventana de TRANSCRIPCIÓN
- Ventana de AYUDA
- Ventana de PREGUNTA

#### 6.1.4.1. PREGUNTAS GENERALES

1. La primera pregunta trata de que drive contiene el Disco de instalación.

A: o B:

2. La segunda pregunta trata de cuál es el boot drive

A: o C:

3. La tercera pregunta es respecto al sendero y nombre completo del directorio en donde se desea almacenar los archivos de BP-LAN, por ejemplo:

C:\BP-LAN

4. Por ultimo, el programa automático de instalación pregunta al administrador de la red si todas las respuestas a este bloque de preguntas están correctas

Y/N/E (Si, No, Exit)

#### 6.1.4.2. PREGUNTAS DEL PUERTO SERIAL

1. La primera pregunta al respecto de los puertos seriales es cuántos sockets de red (RS-232C) desea instalar.

Aquí se pueden definir un máximo de 256 sockets que es el máximo número posible en un byte (socket 0 a socket 255),

pero sabemos que esto es teórico y se aleja de toda realidad práctica.

Cabe anotar que si el caso es de más de dos puertos seriales, es decir más de dos clientes, habrá que instalar una tarjeta multipuerto en el servidor.

El programa preguntará cuál es la dirección base hexadecimal de cada socket, generalmente 3F8, 2F8, 3E8, etc.

La siguiente pregunta es respecto a la rata de baudios (baudrate) de cada socket, donde velocidades de 57600 baudios para las más lentas y 115200 baudios para las más rápidas.

#### 6.1.4.3. PREGUNTAS DE DRIVE COMPARTIDOS

El programa pregunta cuántos drives compartidos desea montar.

Si es en el servidor la respuesta debe ser "0" y las siguientes preguntas no se responderán ya que el concepto es que el servidor no tiene drives compartidos.

Si el caso es en los clientes, se deberá responder el número de drives del servidor que desearía compartir el cliente, puede ser disco duro, floppy drives, etc.

La siguiente pregunta es con respecto al drive de mas alta prioridad para la instalación de BP-LAN que no está en red.

Si el sistema tiene dos drives la respuesta puede ser b:

Si el sistema es de disco duro la respuesta puede ser c:



El utilitario también pregunta por cada socket conectado a cada cliente para cada drive compartido.

Así por ejemplo: socket # 0 para el drive compartido D:

La cuarta pregunta se refiere a que drive en el servidor corresponde a tal drive compartido, por ejemplo:

El drive C: en el server corresponde al drive compartido D en el mismo servidor.

#### 6.1.4.4. PREGUNTAS DE CORREO ELECTRÓNICO

Como la red tiene un correo electrónico es necesario en el proceso de instalación la inclusión de un bloque de preguntas destinado al efecto.

Si el proceso de instalación se lo está haciendo en el servidor, no se deberá responder a las preguntas de este bloque, sino que se las parará presionando RETURN.

Como paso previo debemos crear manualmente un directorio donde almacenar los mensajes del correo electrónico en el server y en un drive compartido, ejemplo: D:\mail

Se preguntará por el sendero del directorio donde se ha decidido previamente almacenar los mensajes: D:\mail

Después preguntará por el sendero del cliente donde está el procesador EDLIN.COM (editor de texto ASCII), ejemplo:

c:\dos\edlin.com

La última pregunta se refiere al único nombre o identificación del usuario. El nombre de usuario corresponde

un archivo en el directorio de correo por lo que se deberá escoger un nombre compatible, por ejemplo:

PEDRO, PEDRO\_1, etc.

El utilitario copia todos los archivos que comienzan con "P" del drive fuente al directorio de la red o destino.

También el AUTOEXEC.BAT y CONFIG.SYS en el drive de booteo son copiados como AUTOEXEC.OLD y CONFIG.OLD respectivamente en el directorio de destino de la red. Un archivo tipo BAT llamado BPRUN.BAT, el cual setea la variables del DOS, incluyendo el sendero, y configura todos los sockets de red, es creado.

Una línea que llama a BPRUN.BAT es añadida al final del AUTOEXEC.BAT en el drive de booteo. Los comandos de los dispositivos drivers a ser montados en los drives compartidos son añadidos al final del CONFIG.SYS en el drive de booteo.

#### 4.1.5. REMOCIÓN DE LA RED

Para remover la red del sistema es necesario seguir los siguientes pasos:

1. Copiar el CONFIG.OLD del directorio destino de la red a CONFIG.SYS en el drive de booteo.
2. Copiar el AUTOEXEC.OLD del directorio destino de la red a AUTOEXEC.BAT en el drive del booteo.
3. Borrar todos los archivos de la red del directorio de destino de la misma.
4. Remover el directorio destino de la Red.

## 7. CAPITULO

### **7.1. Análisis de los programas**

Para poder implementar la red por medio del BP-LAN, hay que cargar inicialmente unos programas residentes de memoria, de los cuales se hablo hace unos cuantos párrafos.

Para la implementaron de este tipo de programas hay algunas alternativas. Los que están implementados en este sistema son los mas sencillos, ya que solo hacen uso de la INT 27h.

Decimos que es la manera mas sencilla, ya que lo único que se hace es apuntar con bx al segmento que deseamos mantener residente. Hay otro tipo de programas residentes en memoria, que utilizan otras técnicas. Por ejemplo se suele reservar un bloque de memoria, que es en donde se cargara el programa. Este método tiene la ventaja de que se puede hacer una verificación, para determinar si ya se cargo el programa como residente o no. Ademas estos por lo general también se los puede desinstalar.

En nuestro caso no se los puede desinstalar ni tampoco se puede determinar si es que ya ha sido instalado.

En BP-LAN tenemos dos principales programas residentes. Son BPSIOS.COM y BPSERIAL.COM y son los básicos para su operación.

BPSIOS es el que redefine los bios y le incluye las nuevas interrupciones que serán usadas por esta red.

BPSERIAL es el que especifica el puerto serial, la velocidad de transmisión, el tipo de paridad, la longitud de datos y el número de bits de parada.

### 7.1.1. BPBIOS.ASM:

En este modulo se especifican las nuevas interrupciones y se establece que parte de programa se debe ejecutar al ser invocada una de ellas. Esto se lo realiza mediante la INT 21H con AH=25H, que es SET INTERRUPT VECTOR. En AL se carga el numero de la interrupción y en DX se carga el offset al puntero del manejador de interrupciones. Asi tenemos por ejemplo:

```
mov ah,25h
mov al,phys_int
mov dx,offset physical_interrupt
int 21h
```

En este ejemplo se ve claramente como se cargan los distintos registros (Nótese que phys\_int esta definido como 81H. El offset physical\_interrupt apunta a ese procedimiento que se halla en PHYSICAL.MOD).

El mismo procedimiento se hace para las otras interrupciones definidas.

### 7.1.2. BPBIOSHD.MOD

Es un programa en el que se definen muchas constantes de operación, por lo que debe estar incluido en la mayoría de los programas.

Aquí se definen constantes de uso general como son las de cancelación, inicio de cabecera, reconocimiento y negación del reconocimiento.

Aparte se definen las interrupciones para cada capa. Se nota que hay capas que no están implementadas en este tipo de red, pero que se han dejado definidas sus interrupciones para implementaciones futuras.



Sin duda que las partes de mas importancia la constituyen los macros de BPBIOS, GET\_PACKET y PUT\_PACKET.

**BPBIOS:** Al invocar este macro, debemos incluirle una serie de parámetros, para que se pueda realizar su ejecución. Estos parámetros son:

**intnum:** Se especifica el numero de interrupción que se desea ejecutar.

**funcnum:** Se indica la función que se desea realizar. Esto es, ya que una misma interrupción puede ejecutar varias funciones distintas. Estas son cargadas en AH.

**portnum:** Se establece a través de que puerto de esta realizando la comunicación.

Parte de estas hay otras de menor importancia, y que para el análisis que se va a realizar no intervienen.

**GET\_PACKET:** Este macro y el de PUT\_PACKET son idénticos. Se reciben una serie de parámetros, los cuales son asociados con sus respectivos registros.

Los distintos parámetros que se establecen son los siguientes:

**portnum:** Se especifica el puerto de comunicación.

**packet\_length:** longitud del paquete.

**packet\_address:** Dirección del paquete.

Una vez realizado esto se ejecuta INT LINK\_INT. Esto es una interrupción concerniente a la capa de enlace. Se intercepta esa ejecución y ejecutan procesos definidos en DATALINK.MOD. Es en ese modulo que se realizan las operaciones de transmitir o recibir paquetes.

### 7.1.3. MISC.MOD:

Este modulo esta incluido en la gran mayoría de los programas, ya que incluye funciones sencillas pero de uso

general. Aquí hallamos macros y procedimientos como los siguientes:

**pushall:** Hace un push de todos los registros

**popall:** Hace un pop de todos los registros, pero manteniendo consistencia con el pushall

**get\_opt:** Aquí se determinan los parámetros que han sido ingresados en la línea de comando, y que son necesarios para la ejecución de un programa.

**get\_dec:** Es llamado en caso de que el parámetro este ingresado en forma hexadecimal. Esto puede ocurrir al ingresar posiciones de memoria correspondientes a los distintos puertos que se usaran.

**get\_hex:** Idéntico al caso anterior, pero es llamado en caso de que los datos estén en forma hexadecimal.

**get\_ticks:** Esta es la manera básica de obtener la hora de un computador. Especifica mente en la posición 0000:046h se almacena un valor que corresponde al numero de "TICKS" ocurridos desde media noche.

**want\_a\_sec:** Es un pequeño retardo de aproximadamente un segundo.

**slash:** En la línea de comando, al ingresar los parámetros, estos son ingresados precedidos de un "/". Entonces este programa trocea la línea carácter por carácter y cuando encuentra un "/" sabe que lo que sigue es un parámetro u opción del programa.

**uppercase:** Este convierte los caracteres minúsculas de AL en mayúsculas.

Además de estos, hay algunos macros para el movimiento de datos entre registros y segmentos.

#### 7.1.4. LOWLEVEL.MOD:

Aquí se encuentran todas la funciones y procedimientos de bajo nivel, necesarios para el proceso de los paquetes.

Muchas de estas funciones son llamadas principalmente mediante el **PROCESS\_PACKET** del **BPSERVER.ASM**

Todos los paquetes utilizados por BPLAN llevan en la cabecera un carácter que determina el tipo de paquete.

Estos caracteres pueden ser los siguientes:

**r:** Indica que el paquete lleva un requerimiento de lectura de un rango específico de memoria del servidor. Es ejecutado por medio de peekmem.

**w:** Es lo contrario del caso anterior, es decir que escribe en un rango de memoria dado. Lo ejecuta pokemem.

**i:** Lee un byte de una dirección específica de I/O del server. Es procesado por portin.

**o:** Escribe un byte en una dirección específica de I/O del server. (portout).

**e:** Ejecuta un código de instrucciones, pasados en el paquete anterior, en el servidor.

**f:** Llena una dirección específica del servidor con un byte específico (fillmem).

**m:** Mueve un rango determinado de memoria del servidor a otro rango de memoria del mismo servidor.

De todas estas funciones, ahora nos dedicaremos a analizar una de ellas que por el momento las consideramos las más importantes. Estas son:

1.- **portin:** El formato del paquete que llega es el siguiente:

Packet ID.	I/O Address
" i "	2 Bytes

Mediante la interpretación de la cabecera del paquete, determinamos que corresponde a un ingreso de un byte por un



puerto (Ya que es "i"). Así mismo podemos determinar la dirección del puerto que se encuentra en los 2 bytes que siguen en el paquete. Estos se hallan en PACKET\_BUFFER, y son cargados al registro DX. Ahora se hace un IN.

Los siguientes pasos son un tanto confusos, ya que no se ajustan a los diagramas de flujo indicados, y así mismo crean un nuevo tipo de paquete (Con cabecera "D" y PACKET\_LENGTH = 2), del cual no se hace ninguna mención en el libro.

Una vez que ha leído el byte del puerto, envía el paquete por medio de put\_packet y los siguientes parámetros: socket\_num, packet\_length, offset packet\_buffer. A su vez carga BL con portnum, CX con packet\_legth, DX con packet\_address y AH con transmit\_packet. Este último corresponde a una interrupción del Data Link Layer, para luego hacer una int link\_int. Esta interrupción es interceptada por el bios, que es una especie de "interpretador". Le podemos decir así, ya que interpreta la interrupción y determina si es una interrupción normal del DOS o es una de las ampliadas por BPLAN. En vista de que ya está instalado un nuevo vector de interrupciones, lo que se ejecuta con esa orden es un proceso llamado out\_packet que se encuentra en DATALINK.MOD. Aquí es donde se utiliza el transmit\_byte, de manera que el DATALINK sabe que se debe ejecutar el out\_packet.

**2.-PORTOUT:** Este al contrario del anterior, pone un dato en el puerto para ser enviado. Al recibir el paquete, este contiene en la cabecera una "o", seguido de 2 bytes de la dirección de I/O y seguido de un byte de DATA. Con esto carga la dirección del puerto en DX, y en AL carga el dato a ser enviado. Para enviar el dato se ejecuta la función OUT.



### 7.1.5. FILESERV.MOD

Este modulo es el que se encarga de los requerimientos de lectura, escritura y chequeo de MEDIA. Los paquetes van encabezados con cualquiera de las siguientes opciones:

- R: Para ejecutar una acción de lectura
- W: Para realizar una escritura
- D: Para hacer una revisión al MEDIA

**ESCRITURA:** En caso de que la cabecera del paquete lleve una "W", se ejecutara el proceso llamado "wrisec". Lo que se hace inicialmente es cambiar una bandera, indicando de esta manera que el media ha cambiado. El media se refiere propiamente dicho al disco duro, su estructura, tipo y contenido. El objetivo de cambiar esta bandera es el indicar que el contenido que tenia el disco ha cambiado. Esto se hace principalmente debido a que los servidores están provistos de una memoria cache para el disco duro; de tal manera que, antes de buscar la información en el disco, lo hará primero en el cache. La idea de hacer esto es la de incrementar la velocidad, ya que en la memoria cache no se involucran movimientos mecánicos, que son los que en definitiva retrasan los tiempos de proceso.

Entonces, al recibir un requerimiento de lectura de disco, se verificara primero la bandera de MEDIA. Si la bandera indica que hubo un cambio, se accederá directamente al disco, caso contrario se accederá al cache, evitando así datos incorrectos o desactualizados.

En el proceso normal de recibir un paquete con cabecera "W", luego de cambiar la bandera, se creara un nuevo paquete con cabecera "D". Esto al igual que en el caso de portin es un paquete desconocido, del cual no se hace mención en el libro. Seguido se puede leer del paquete la información del disco, de los bits MSB y LSB del sector inicial y la longitud.

Packet ID	Drive #	Sector #	Length of Data
" R "	2 Bytes	2 Bytes	2 hasta 1024 Bytes

Dentro del programa se inicializa otra variable llamada "funcion" que especifica a otros segmentos del programa si se trata de lectura (si funcion=0) o escritura (si funcion=1). En BX serán almacenados los datos que deberán ser escritos y luego se ejecutara el modulo "sector\_io".

En este caso se inicia CX=1 para trabajar con 1 sector y en DI se especifica el sector inicial. En BX serán cargados los datos y luego se ejecutara la INT 26h que es la de escritura absoluta de sector.

En este punto se termina con la ejecución de sector\_io y se retorna a wrisec, donde leemos un nuevo paquete y repetimos el proceso hasta encontrar un paquete con cabecera "T", que indica fin de los paquetes.

Según encontramos otra inconsistencia, ya que el servidor de archivos debería ser el que genere un paquete con cabecera "T", para que de tal manera la estación sepa que el proceso de escritura ha sido realizado exitosamente, pero, siguiendo la secuencia del programa, se interpreta que uno de los paquetes que envía el terminal lleva cabecera "T". Según la tabla de la pagina 81 del libro guía, entre los paquetes del server se halla uno que es indicador de que la escritura de un sector se ha ejecutado, lo que se contradice con la secuencia del programa.

**LECTURA:** La lectura se produce casi idénticamente a la de la escritura. El paquete lleva el siguiente formato:

Packet ID.	Drive #	Sector #
" R "	1 Byte	2 Bytes

Se reciben los parámetros necesarios para la lectura de un sector específico del disco. Nótese que en estas circunstancias no será necesario modificar la bandera de MEDIA, ya que la lectura no altera en nada el contenido del disco. Es por esto que directamente se ejecuta la lectura por intermedio de la INT 25h. Los registros que se cargan son:

DI= Numero del disco

DI= Offset de dirección de transferencia

DI= Numero de sectores a leer

DI= Sector inicial

Una vez realizada la lectura, se creara un nuevo paquete con cabecera "E". Aquí encontramos un nuevo problema, ya que un paquete de este tipo es de ejecución remota en el servidor. En realidad no se entiende el porque de este tipo de paquete, el cual será ensamblado y enviado por put\_packet. De esta manera se termina el proceso de lectura de disco.

Vale anotar que la interrupciones 25H y 26H corresponden a lectura y escritura absoluta de disco. Es decir que en este tipo de acceso a disco se pierden las ventaja de usar la INT 21H que provee manejo de directorios y bloqueo y desbloqueo de archivos.

Debido a que INT 26H y 25H tratan todos los registros como una longitud de sector, se debe implementar un propio sistema de bloqueo y desbloqueo de archivos. Esta operación accesa directamente un sector o bloque de sectores.

Dentro de este modulo también se determina la cantidad de espacio libre en disco y se lo hace mediante:

mov ah,36h

mov dl,drive

int 21h



Retorna:

AX= sectores por cluster

BX= # de clusters disponibles

CX= # de bytes por sector

DX= # de clusters por drive

#### 7.1.6. CONSOLE.MOD

Contiene una serie de procedimientos de muestreo y manejo de pantallas así como de lectura de teclado. Forma parte de BSMOUNT.SYS.

Inicialmente se define BACKGROUND=0, indicando que la operación se realiza en FOREGROUND.

Contiene procedimientos como por ejemplo:

**ESCTST:** Este sensa si se ha oprimido la tecla ESC. Para esto usa la INT 16h con AH=01 para leer el status. Si no hay carácter disponible se enciende la bandera de ZERO. Caso contrario el valor de carácter se carga en AL. En caso de contener AL el numero 27, que es el asociado con la tecla ESC, se sabe si ha sido o no oprimido esa tecla.

**ESCAPE:** Este proceso hace uso de ESCTST para verificar si la tecla fue o no presionada. En caso afirmativo de ejecuta INT 20h que es para finalizar el programa (Program Terminate).

**CLS:** Proceso para limpiar la pantalla.

**CHAROUT:** Aquí se hace escritura en pantalla en modo teletipo, ya que se usa AH=0eH e INT 10H. Al contiene el carácter que deseamos mostrar.

#### 7.1.7. BPTIME.ASM

Este modulo sirve como ejemplo para explicar un poco mejor algunas de las operaciones realizadas por LOWLEVEL.MOD. Este modulo se ejecuta en las estaciones de trabajo, y su función



es la de obtener información de la hora del servidor. Esto se hace, debido a que por lo general las estaciones de trabajo suelen ser computadoras muy limitadas, es decir, muchas veces hasta carecen de una tarjeta que retenga la información de la hora y fecha. Esto implica que cada vez que se enciende el computador, haya que actualizar manualmente la fecha y hora. Entonces, al ejecutar este programa, se la fecha del servidor, que es una computadora mas compleja que si esta provista de ese tipo de tarjeta. Ademas de esta manera se unifica la hora de todas las computadoras conectadas. Este programa arma un paquete de la siguiente manera:

PACKET ID.	ADDRESS OFF.	ADDRESS SEG.	LENGTH OF DATA
"r"	046CH	0000	0004

El paquete lleva esta forma debido a que es un proceso de lectura ( "r" ) de un valor de memoria del servidor.

Se leerá la posición 0000:0046CH que es donde todas las computadoras almacenan la información de la hora. Y se especifica una longitud de 4 debido a que esa es la longitud de la hora.

La notación para la ejecución es la siguiente:

ERTIME / numero\_de\_socket

Donde numero\_de\_socket determina a través de que socket se realizara la comunicación.

Se empieza por hacer las siguientes definiciones:

**time\_request:** Esto corresponde al requerimiento que deseamos hacer, que en este caso es "r" para hacer la lectura en el server.

**time\_addoff:** Aquí se especifica el offset de la dirección de la memoria a leer que es 046CH.

**time\_addseg:** Se especifica el segmento (0000)  
**time\_length:** Longitud del dato a leer. En este caso es 4.  
Inicialmente se hace una verificación de que la línea de comando incluya el numero\_de\_socket. Luego se ejecutan dos funciones que se definen en BPBIOSHD.MOD que son las de put\_packet que pone el paquete en el medio de transmisión para que el servidor lo interprete. Luego el servidor devuelve el paquete y se lo recibe con get\_packet. Una vez recibido el paquete, pone los valores en sus registros correspondientes y de esta manera se actualiza la hora. El programa finaliza con INT 20H de terminación de programa.

Este es un programa que viene incluido en la red, aunque no tenga nada que ver con la red propiamente dicho. Este solo sirve de ayuda para poder determinar si los puertos de comunicación están bien definidos.

Este programa simplemente muestra lo digitado en un computador en la pantalla del otro y viceversa. Si se establece correctamente la comunicación, no deberían haber problemas posteriormente.

Lo interesante de este programa es que utiliza el BPBIOS y el SERIAL como programas de soporte.

El modo de operación de este programa es muy sencillo y opera de la siguiente manera:

Inicialmente se establecen los parámetros que han sido ingresados en la línea de comando, específicamente del número de puerto.

Luego se lee el estatus del teclado mediante INT 16H AH=1.

Esto se lo hace para determinar si ha sido oprimida alguna tecla. Si se ha detectado la presencia de alguna tecla, se determina cual fue, mediante la INT 16H AH=0. El valor de la tecla que fue oprimida se cargara en AL.

Se verifica que la tecla presionada no haya sido ESC, ya que en este caso se terminara con la ejecución del programa.

Una vez que se tiene cargado el registro AL, se invoca al BPBIOS con los sgtes parámetros: PHYS\_INT, TRANSMIT\_BYTE, PORTNUM. Con PHYS\_INT se especifica que queremos realizar una interrupción a la capa física. TRANSMIT\_BYTE indica que deseamos transmitir el byte que tenemos en AL, y PORTNUM especifica el numero del puerto.

En caso de que no se detecte la presencia de ninguna tecla, se hace un BPBIOS PHYS\_INT, RECIEVE\_STATUS, PORTNUM. Con esto se hace una verificación al puerto de comunicación, para determinar si hay algún caracter presente. En caso negativo se repite todo el proceso, pero en caso positivo se lee el caracter mediante BPBIOS PHYS\_INT, RECIEVE\_BYTE, PORTNUM. Luego de leer el caracter del puerto se lo muestra con INT 10H y AH=0EH.

## 8. CAPITULO

### 8.1. PROBLEMAS Y SOLUCIONES DE HARDWARE

#### 8.1.1. CABLEADO

Cerca del 95% de los problemas en una Red de Computadores son ocasionados por los cables. Estos problemas se pueden identificar y solucionar siguiendo los items de chequeo que a continuación se relata:

1.- Corresponde el cable al tipo de conexión recomendada?

El tipo de conexión recomendada es la de Modem Nulo donde se contemplan líneas de recepción, transmisión y de tierra.

2.- Es el cable de la longitud adecuada?

La longitud del cable debe minimizarse por razones económicas y de operación, éstas últimas ocasionadas por la degradación de la señal en este medio de transmisión.

Puede ser necesario reducir el baud rate para comunicaciones en cables de longitud mayor a 50 metros.

3.- Pasa el cable con por lo menos 6 pies de cualquier equipo generador de fuertes campos electromagnéticos?

Tales como: lámparas fluorescentes, motores de neveras.

Estos campos electromagnéticos inducen corrientes en los cables, degradando la señal a transmitir.

4.- Tiene el cable torceduras?

Una torcedura puede indicar rotura interna de alambres, por lo que será un factor a chequear.



5.- Está algún alambre del cable roto?

Use continuamente el multímetro u otro tipo de probador para comprobar que ninguna conexión haya discontinuidad eléctrica.

6.- Está algún cable cortocircuitado?

Verificar ésto, ya que se produce frecuentemente en el mismo conector.

7.- Están los cables atornillados en el lugar correcto?

Muchos cables podrían atornillarse antes de verificar que sea la conexión correcta.

8.- Ha probado reemplazando un cable bajo sospecha de falla con uno que se tenga la seguridad de una buena operación?

La única manera de asegurarnos de que tal o cual cable es el que falla es reemplazándolo con uno ya probado como bueno.

## **BIERTO RS-232C**

Una vez asegurados de que los cables no presentan falla, los siguientes puntos a revisar son los relativos a los puertos RS-232C, a continuación enumerados:

1.- Está la tarjeta ajustada firmemente en el slot?

Hay que asegurarse que este ajustada y atornillada.

2.- Si la tarjeta del puerto está en el slot 8 de una IBM XT-compatíble?

Estas tarjetas deben configurarse de manera diferente en el slot 8 de una IBM XT-compatíble, si la instalación manual no

proporciona la información de configurar la tarjeta para el slot 8, no lo instale en ese slot

3.- Está alguno de los slots de la tarjeta dañado?

Slots sucios, con polvo, o torcidos son fallas potenciales de las tarjetas.

4.- Ha probado mover la tarjeta hacia otro slot?

Algunas veces las tarjetas madre están combadas y no ofrecen un buen contacto en algún slot o sino algún punto de soldadura está roto o partido.

5.- Están dos puertos RS-232C mapeados en el mismo lugar?

Si éste es el caso ninguno de los dos puertos funcionará adecuadamente.

6.- Está el puerto RS-232C siendo manejado por interrupciones?

Muchas tarjetas de puerto RS-232C soportan ser manejadas por interrupciones de entrada/salida. Las comunicaciones de alta velocidad BP-LAN pueden fallar cuando a la tarjeta se le permite generar interrupciones. Usualmente es posible deshabilitar esta facultad removiendo el jumper el cual selecciona una línea de solicitud de interrupción.

7.- Ha probado el puerto RS-232C con el programa BPTERM.COM de BP-LAN?

Conecte el cable entre el cliente y el servidor. Instale el BP-LAN BIOS tanto en el cliente como en el servidor ejecutando BPBIOS.COM. Configure el socket 0 tanto en el cliente como en el servidor con BPSERIAL.COM, por ejemplo:

```
BPSERIAL /$3F8 /3 /1
```

esto configura COM1 para 115.200 baudios.

Escriba BPTERM /0 para inicializar el programa de terminal tanto tanto en el cliente como en el servidor. Escriba un mensaje para cada computadora.

a) El mensaje no tiene eco en ninguna pantalla?

Esto puede ser un indicativo de que no funcionan los puertos seriales o una impropia configuración en el puerto. Ambos puertos deben estar seteados a la misma velocidad (baudios) para que la comunicación ocurra.

b) Los mensajes tienen eco pero aparecen montados.

Puede ser un indicativo de que se ha configurado incorrectamente la longitud de la palabra, la paridad o el bit de parada.

c) Están los mensajes teniendo eco en la misma pantalla mientras se son digitados?

Esto es un indicativo de que existe un corto entre las líneas de transmisión y recepción del cable. Algunas veces la señal que transmite le UART induce una corriente en la línea de recepción .

d) La computadora transmite mensajes apropiadamente pero no los recibe de la misma manera.

La línea de recepción puede estar rota, o algún programa TSR puede estar interceptando los caracteres que llegan. Tanto el MOUSE.COM como MOUSE.SYS pueden ser la causa de éste problema.

e) La computadora recibe los mensajes apropiadamente pero no los transmite de la misma manera.

Se puede asumir que puede haber una rotura en la línea de transmisión. Chequear la línea con un puerto que se tenga la seguridad que reciba correctamente.

f) Falla el acceso a la red a pesar de que ambos puertos RS-232C funcionan adecuadamente con BPTERM.COM?

El BPTERM.COM transmite por caracteres, así, la transmisión en red consiste en transmitir un bloque de caracteres en secuencia rápida, dando un muy pequeño espacio de tiempo para procesar cada caracter y recibir el siguiente. Reducir el baud rate en ambos puertos RS-232C incrementa el tiempo de procesamiento de cada caracter, pudiendo ésto ser la solución a nuestro problema.



## CONCLUSIONES Y RECOMENDACIONES

El sistema BP-LAN tiene un gran valor didáctico, pero en la práctica es un sistema muy limitado comparado con los sistemas que se venden en el mercado. Es limitado principalmente en cuanto a niveles de seguridad se refiere. No cuenta con ningún sistema de asignación de claves para los usuarios, lo que lo hace muy vulnerable en ese sentido.

El sistema tal como está implementado, (con acceso de modo sector) nunca será un sistema eficiente, ya que como se ha mencionado antes, no hay posibilidad de bloqueo de registros o archivos. Esto lo hace poco seguro para usarlo.

La velocidad de comunicación de un puerto serial es mucho menor que la de los actuales sistemas de comunicación y es mucho más limitada, ya que no fue diseñada para ese propósito específico.

El programa solo funcionará eficientemente en discos de capacidad máxima de 30Mb. Esto se debe principalmente a que fue diseñado para trabajar solo con FAT de 12 bits y no en FAT extendida de 16 bits. Esto hace que el programa presente grandes problemas al tratar de trabajar con versiones de D.O.S. mayores a la 3.30, ya que esas versiones ya incluyen FAT de 16 bits para poder soportar discos de gran capacidad. Por el hecho de realizarse la comunicación a través de cables multipares y al sencillo código de transferencia de datos, este es un sistema que no se presta para abarcar grandes distancias.

Se ha podido interpretar y comprender la idea de los programas residentes de memoria, sobre todo para interceptar los vectores de interrupción. El sistema de enrutamiento de trabajos de impresión se basa en este esquema y es muy similar al de otro tipo de redes comerciales. Lamentablemente

no tiene previsto la compartición de equipos que no estén conectados al servidor.

Incluye un programa de correo electrónico que es muy completo, y el cual se lo podría adaptar para que funcione con otras marcas de redes, ya que es sencillo de interpretar y manejar.

La forma modular en que está diseñado este programa permite hacer futuras ampliaciones y modificaciones al mismo, para obtener un mejor rendimiento y mayor confiabilidad. Mas aún tomando en cuenta que se basa en el modelo OSI e implementa las dos capas inferiores para su funcionamiento. Dentro del mismo programa se ha tenido el cuidado de dejar listas las especificaciones e interrupciones para la implementación futura de las siguientes capas superiores.

Incluye un programa de ejecución remota para disminuir el tráfico en la red. Este es muy útil, ya que para determinados procesos se recargaría de un tráfico intenso e innecesario a la red.

Debido a la manera en que se implementan los programas residentes de memoria en este programa, es casi imposible poder cargar algún otro que no forme parte de la red. Esto es, ya que los programas no reservan partes de memoria ni la bloquean. Entonces al cargar otro programa residente, se podría cargar encima de alguno de los de la red, e interferir de esta manera con las funciones básicas de la red.

En definitiva podríamos decir que el programa nos ha ilustrado y enseñado los pasos necesarios para la implementación de sistemas operativos, programas residentes en memoria y los dispositivos instalables.

Se ha podido apreciar los distintos niveles de seguridad que se deben implementar para la correcta transmisión y recepción de datos. Definitivamente, en esto nos ayuda en gran mayoría

el uso de los paquetes para la transmisión de datos, ya que de antemano se puede establecer que tipo de dato se espera. Con los conocimientos que nos ha proporcionado este trabajo, podríamos pensar en futuras modificaciones, una de las principales que deberíamos tener en cuenta es la de cambiar el modo de acceso al disco. Apesar de que esto complicaría notablemente el programa, no habría la limitación de trabajar con FAT de DOS, obteniendo de esta manera un programa que pueda trabajar en cualquier versión de sistema operativo y disco duro. Una vez logrado esto, podríamos reemplazar el sistema de semaforos por sistemas de bloqueo de registros y archivos.