

## INTRODUÇÃO

Na década de 80, o normal era não se possuir um modem, pois era um item dispendioso e de utilidade muito restrita; a velocidade era de 300bps. Aparelhos de fax eram analógicos e raríssimos, a forma de comunicação “rápida” entre empresas era o Telex, que necessitava de uma linha especial (não se comunicavam através da linha telefônica), não permitiam acentuação, caracteres minúsculos, etc. A velocidade de transmissão era de 50bps (com apenas 5 bits por caracteres ao invés dos 8 empregados normalmente). Um modem atualmente pode transmitir 500 caracteres no mesmo tempo que um Telex necessitava para enviar um único caracter.

Em 1990 poucas empresas e “micreiros” (normalmente engenheiros e técnicos em eletrônica.. possuíam um modem. Nessa época os modems possuíam uma velocidade de 1.200 bps e alguns poucos privilegiados possuíam modems de 2.400 bps. A instalação era difícil, qualquer ruído na linha telefônica gerava caracteres aleatórios na tela. Agora, passados apenas seis anos, os modems pelo mesmo custo são acima de dez vezes mais velozes (14.400 e 28.800 bps e possuem compressão de dados), se tornaram mais seguros pois possuem os protocolos de correção de erros e não costumam apresentar defeitos tão freqüentemente, chegando a oferecer garantia total de até 5 anos, enviam e recebem fax, e com os programas mais modernos são de fácil instalação (plug and play), etc.

## ÍNDICE

### I. MODEMS

[1.0] O que é e como funciona um modem?

[1.1] Quais as diferenças entre bps, baud, cps, etc?

[2.0] O que é CCITT? E ITU-T?

[3.0] O que são Protocolos?

[3.1] O que são os protocolos de modulação?

[3.1.1] Quais são os protocolos de modulação?

[3.1.2] Qual a diferença entre os protocolos V.Fast, V.34 e V.FC?

[3.2] O que e quais são os protocolos de correção de erros?

- V.42 e MNP 2-4

[3.3] O que e quais são os protocolos de compressão de dados?

- V.42bis e MNP 5

- Diferenças entre V.42bis e MNP5

[3.4] O que são modems RPI e por que são mal vistos?

- Softwares que são RPI-compatibles

[3.5] O que são protocolos de transferência de arquivos?

[3.5.1] Quais são os protocolos de transferência mais comuns?

- Xmodem

- Xmodem-CRC

- Xmodem-1k

- Ymodem

- Ymodem-G

- Zmodem

- ZedZap

- Protocolos Bidirecionais

- ASCII

- Kermit

[3.5.2] Quais são os melhores e os piores protocolos de transferência de arquivos?

[4.0] O que são comandos? Como usá-los?

- O que é padrão Hayes?

- ATD

- ATZ

- +++

- ATA

- Como fazer uma conexão de modems durante uma ligação VOZ?

## II. CONFIGURAÇÃO

- [1.0] Velocidade serial x Velocidade modem (diferenças entre DTE e DCE).  
- Por que não existe a opção de velocidade “14400” no meu programa de comunicação? Ela pula de 9600 para 19200...  
- O que tem o V.42 e V.42bis a ver com isso?
- [2.0] O que é 8N1 e 7E1?  
- Qual usar?
- [3.0] O que é flow control (RTS/CTS - Xon/Xoff)?  
- Qual configuração usar?
- [4.0] O que são portas de comunicação e como configurar meu modem?
- [5.0] O que são terminais de comunicação?  
- Como funciona o terminal ANSI?  
- Que outros terminais existem?  
- Qual usar?
- [6.0] O que colocar no item Dial String?  
- ATDP ou ATDT

## III. ESPECIFICO SOFTWARE

- [1.0] Quais softwares de operação remota de microcomputadores existem e quais as diferenças entre eles?  
- Lap Link 6.0 for Windows  
- PC Anywhere for Windows  
- Doorway 2.22 for DOS  
- Carbon Copy 6.0 for DOS
- [2.0] Video Texto
- [3.0] O que é FOSSIL?

## IV. ESPECIFICO HARDWARE

- [1.0] O que é um chip UART?  
- 8250  
- 16450  
- 16550 (Não “A”)  
- 16550A, 16550AF e 16550AFN  
- Onde está esse chip em meu micro?  
- Por que eu precisaria de um UART 16550?  
- E o que tem UART 16550 a ver com Windows? E com o OS/2?
- [2.0] Qual é a configuração ideal para modems USR Sportster 14.4 kbps?
- [3.0] Qual é a configuração ideal para modems USR Sportster 28.8 kbps?
- [4.0] Quais as diferenças entre o USR Sportster V.34 e o Courier V.34?
- [5.0] Qual é a configuração ideal para modems Zoltrix NON-RPI?
- [6.0] Existem comandos que são bons de se usar em qualquer configuração? Quais?

[7.0] O que é Caller-ID? O que é BINA? O que relaciona ambos?

[8.0] O que é resposta adaptativa?

[9.0] Modem interno ou externo?

[10.0] Por que não consigo conexões a 9600 bps com meu modem que diz ser 2400/9600 bps?

## V. TENDÊNCIAS FUTURAS

[1.0] WebTV

[1.1] Como funciona?

[2.0] Cable Modem

[3.0] Network Computers

[4.0] Modem Celular

[5.0] O Modem de 56 kbps

[5.1] Caminhos assimétricos

[5.2] Problemas de controle de fluxo

[5.3] Problemas de instalação

## I. MODEMS

Essa parte apresenta os tópicos básicos que devem ser conhecidos sobre como funcionam os modems e o jargão técnico que é usado nesse contexto.

Ela apresenta os seguintes temas:

1.0 Funcionamento de um modem

1.1 Conceitos de velocidade (bps, baud, cps)

2.0 CCITT e ITU-T

3.0 Protocolos

3.1 Protocolos de modulação

3.2 Protocolos de correção de erros

3.3 Protocolos de compressão de dados

3.4 Modems RPI

3.5 Protocolos de transferência de arquivos

4.0 Comandos Hayes

I.[1.0] O que é e como funciona um modem?

Sabemos que um modem está ligado à linha telefônica e ao computador. Sabemos também que ele serve para fazer a comunicação entre o micro e um outro micro através da linha telefônica.

Surge então a questão, por que um computador não pode transmitir dados diretamente pela linha telefônica para outro computador. Para que é necessário o modem?

- O telefone

Vejamos como funciona um telefone comum:

O microfone do bocal converte as ondas de som que vem de sua boca em sinais elétricos. Do outro lado, o alto falante no

fone de ouvido converte novamente esses sinais elétricos em ondas sonoras. Os sinais elétricos trafegam pela linha telefônica por meio de oscilações de voltagem, podendo assim representar as ondas sonoras em sua frequência e altura (amplitude). Esses sinais são chamados de ANALÓGICOS, pois são uma analogia elétrica do som de sua voz. Pela linha telefônica somente este tipo de sinal pode trafegar.

#### - O computador e a porta serial

Já o computador se comunica com seus periféricos por meio de bits e bytes. Um bit é a menor unidade computacional, e aceita dois valores, ZERO ou UM. Um conjunto de oito bits forma um BYTE. Através das portas SERIAIS (COMx, normalmente usadas para mouses, modems) o micro se comunica bit a bit com o periférico, e através das portas PARALELAS (LPTx, normalmente usadas para impressoras), byte a byte.

Uma porta serial utiliza sinais elétricos. São voltagens positivas e negativas (normalmente +12V e -12V) para representar o zero e o um. Alternando as voltagens, ela pode se comunicar com periféricos externos bit a bit. Este sinal é um sinal DIGITAL, com o qual o computador tem grande facilidade de trabalhar.

#### - Os modems

Os modelos de modems mais antigos fazem o seguinte: Transformam essas voltagens positivas e negativas que vêm do micro em tons audíveis. Uma voltagem negativa (representando um bit 1) é convertida em um tom de determinado pitch; uma voltagem positiva (representando um bit 0) em um tom de pitch um pouco mais baixo. Esses sons são transmitidos pela linha telefônica da mesma forma como a voz. O modem receptor por sua vez converte esses sons em sinais digitais e os transfere para o micro, que os interpreta.

Desse processo vem a palavra MODEM, que é a sigla de MODulador / DEModulador. Um lado modula os sinais digitais em sinais analógicos, enquanto o outro lado demodula esses sinais analógicos novamente para sinais digitais.

#### - A modulação

Na verdade o termo MODULAÇÃO envolve bem mais do que somente isso. O uso de dois tons visto acima é chamado de modulação FSK. Com ele, se tem o limite de 300 bits por segundo numa transmissão de dados.

Modems atuais usam mais do que os dois “estados” possíveis (no exemplo anterior, cada tom representa um “estado”). Esses modems mais avançados podem combinar estados (como por exemplo a amplitude) com mudança de estados (por exemplo a mudança de fase) afim de representar grupos de dois, três, quatro ou mesmo mais bits.

### I.[1.1] Quais as diferenças entre bps, baud, cps, etc?

O BAUD é uma unidade de medida representando a quantidade de mudanças de fase por segundo que podem acontecer (por exemplo numa linha telefônica). As conexões de telefone atuais podem ser usadas de maneira confiável afim de transmitir um sinal que muda de estado até 2400 vezes por segundo. Tal conexão está operando a uma taxa de 2400 bauds.

Mas com técnicas de modulação mais complexas, podemos não ter apenas dois estados, mas sim muitos estados. Com oito estados diferentes, podemos usar cada estado para representar um dos oito grupos possíveis de três bits (000, 001, 010, 011, 100, 101, 110 e 111). Dessa forma, em uma taxa de sinalização de 1600 bauds (por exemplo), é possível transmitir 4800 bits por segundo (bps). Modems mais rápidos e mais comuns hoje em dia usam 64 estados, que podem representar todos os valores possíveis de um grupo de seis bits. Assim, a 2400 bauds, podem ser transferidos até 14400 bits por segundo (2400 bauds x 6 bits = 14400 bps).

A diferença entre BAUD e BPS se torna evidente. BAUD é a unidade de um valor que alcança no máximo algo em torno de 3000 em linhas telefônicas comuns no Brasil. BPS (bits por segundo) é a velocidade real do modem nessa linha telefônica, sendo que os modems mais velozes hoje em dia podem transmitir até 28800 bps (bits por segundo). Mas para conseguir esses 28800 bps, é necessária uma taxa de sinalização de aproximadamente 3200 bauds. Como este é quase o topo máximo que nossas linhas telefônicas alcançam, é possível que mesmo tendo um modem 28800 bps não se alcance essa velocidade, mas sim alguma velocidade inferior (26400 bps, 24000 bps ou 21600 bps).

O último termo que ainda não foi abordado é o de CPS. Significa “caracteres por segundo” e poderia ser chamado também de “byte por segundo”. Esse último termo não é muito usado, pois a sigla deste seria igualmente BPS, confundindo ela com o “bit por segundo”.

Um byte possui normalmente 8 bits. Em transmissões pela porta serial, são necessários dois bits adicionais, o START e o STOP bit, totalizando 10 bits. Assim sendo, numa conexão a 14400 bits por segundo, são transmitidos 1440 cps (caracteres por segundo). Veremos mais adiante porque esse valor normalmente é ainda maior.

1440 caracteres por segundo é uma velocidade bastante considerável. Uma pessoa com dotes de digitação médios consegue digitar no máximo a uns 10 caracteres por segundo, quando muito.

Essa taxa é comumente apresentada em telas de DOWNLOAD e UPLOADS de programas de comunicação, informando assim quantos BYTES estão chegando ou saindo por segundo.

## I.[2.0] O que é CCITT? E ITU-T?

Vimos que em modems de alta velocidade (como são chamados os modems a partir com velocidades de 9600 bps e acima) em um baud são modulados mais de um bit. Agora a forma como esses bits são modulados, precisou de alguma forma ser padronizado. Se não houvesse essa padronização, um modem não entenderia o que o outro estivesse falando, ou seja, um não conseguiria demodular o que outro modulou.

Quem fez grande parte dessa padronização foi um órgão chamado CCITT, sigla para Comitê Consultivo Internacional de Telegrafia e Telefonia. Este comitê definiu uma série de padrões para a telecomunicação, que são os padrões chamados de V.xx, onde xx é o número da padronização.

Desde o começo do ano de 1994 o CCITT não existe mais. Ele foi substituído ITU-T (International Telephone Union - Setor pra padrões na telefonia), órgão pertencente à ONU (Organização das Nações Unidas).

As atividades da ITU incluem a regularização, padronização, coordenação e desenvolvimento da telecomunicação internacional. A ITU é basicamente organizada em 3 setores que refletem suas atividades principais (Padronização, Radiocomunicação e Desenvolvimento). O que importa para usuários de modems é o Setor de Padronização da Telecomunicação (ITU-T). Essa agência cria entre outros as recomendações de padrões para a comunicação de dados.

Com essa padronização, que tem âmbito e validade MUNDIAL, os modems fabricados pelos mais diversos fabricantes passam a poder se comunicar entre si sem maiores dificuldades, já que “falam” a mesma “língua”.

Existem ainda outros padrões que não foram definidos pela CCITT (hoje ITU-T). São por exemplo os padrões Bell, definidos por uma companhia telefônica americana (Bell), ou o protocolo HST (High Speed Technology) criado pela USRobotics (nesse caso, só pode haver conexão de um modem com esse protocolo com outro que também tenha esse protocolo, no caso, um outro USRobotics HST). Esses protocolos não têm validade mundial, e portanto não são encontrados em todos os modems. São os chamados protocolos proprietários.

Além desses protocolos que definem a modulação para as diversas velocidades, existem outros tipos de protocolos da ITU-T que interessam ao universo dos modems, conforme veremos mais adiante (I.[3.2] e I.[3.3]).

Outros protocolos que não são padrões mundiais (da ITU-T), mas sim proprietários (criados por uma determinada empresa) eram os protocolos MNP. Mas esses protocolos entraram em domínio público (a especificação do protocolo se tornou propriedade pública) e hoje são encontrados na maioria dos modems de alta velocidade.

## I.[3.0] O que são Protocolos?

Hoje em dia os modems não só são mais rápidos como também são repletos de novos aspectos, como controle de erros e compressão de dados. De repente, você é confrontado com todas essas siglas: V.32, V.32bis, V.42, V.42bis, MNP5, LAP-M, etc. O que cada uma significa? O que que cada uma significa para você?

Para tirar o máximo de proveito de um modem de alta velocidade você precisa entender três diferentes tipos de protocolos e suas relações. São eles:

- Protocolos de modulação
- Protocolos de controle e correção de erros
- Protocolos de compressão de dados

### I.[3.1] O que são os protocolos de modulação?

As técnicas específicas para modular os bits digitais em sinais analógicos são chamadas de protocolos de modulação. Os vários protocolos de modulação definem o método exato dessa codificação e a velocidade da transferência resultante. Na realidade, você não pode ter um modem sem protocolo de modulação. Um modem normalmente suporta mais de um protocolo de modulação.

A velocidade crua (sem compressão de dados) de um modem é determinada pelo protocolo de modulação. Este é negociado entre dois modems na hora da conexão por envio e recepção de sinais característicos (os famosos “beeb-bbeebe-beebe” que se ouve na hora da conexão). Modems de alta velocidade são modems que suportam protocolos de modulação de 9600 bps ou acima (bps é a abreviatura de “bits por segundo”, ou seja, quantos bits podem ser transferidos por segundo).

#### I.[3.1.1] Quais são os protocolos de modulação?

- até 2400 bps

Um modem de 2400 bps compatível com o Hayes normalmente suporta os seguintes protocolos de modulação:

- Bell 103 (padrão americano para 300 bps)
- Bell 212A (padrão americano para 1200 bps)

ITU-T V.22 (padrão mundial para 1200 bps)  
ITU-T V.22bis (padrão mundial para 2400 bps)

Alguns modems de 2400 bps também suportam os seguintes protocolos:

ITU-T V.21 (padrão mundial para 300 bps)  
ITU-T V.23 (padrão europeu para 1200/75 e 75/1200 bps.  
Esse padrão é usado também pelo sistema de Video Texto no Brasil)

- acima de 2400 bps

Existem hoje três protocolos de modulação que são padrões mundiais para modems de alta velocidade: V.32, V.32bis e V.34. Os dois primeiros foram estabelecidos pela antiga CCITT e o último já pela ITU-T.

- V.32

Esse é o padrão para modems de 9600 bps (e 4800 bps). V.32 foi adotado como padrão pela CCITT em 1984. Mas o mercado para esses modems demorou um pouco para crescer. Um modem com V.32 custava por volta de US\$ 2000 no ano de 1988. Atualmente modems até esse protocolo (que não tenham nenhum protocolo para velocidades superiores) estão praticamente obsoletos e não existem em grandes quantidades no mercado. Nessa época surgiram alguns outros protocolos que não eram padrão mundial, mas proprietários de certas empresas fabricantes de modems, como a USRobotics, a Telebit, a Hayes e CompuCom, que criaram os modems com o chamado “dual standard” (padrão duplo) que suportavam tanto o V.32 e seu próprio padrão.

- V.32bis

V.32bis, estabelecido no começo de 1991 é o padrão da ITU-T para modems a 14400 bps. Um modem com V.32bis possui também capacidade de conectar a velocidades menores (“fall back”): 12000, 9600, 7200 e 4800 bps. No V.32bis está incluso o V.32.

Diferente da época dos modems de 2400 bps onde um só protocolo de modulação (V.22bis) era suportado por todos os fabricantes de modems, na época do V.32bis surgiram vários protocolos de modulação proprietários (não aprovados pela ITU-T) criados por fabricantes de modems. Os melhores exemplos são os protocolos HST da USRobotics e o PEP da Telebit.

- V.34

Atualmente o protocolo de modulação mais veloz para modems. Suporta conexões de até 28800 bps, com um “fall back” inteligente para velocidades inferiores, caso a linha não tenha condições de agüentar a alta velocidade de 28800 bps: 26400, 24000, 21600 e 19200 bps. Possui um método de negociação inteligente, que se adapta à qualidade e condição da linha telefônica.

I.[3.1.2] Qual a diferença entre os protocolos V.FAST, V.34 e V.FC?

- V.FAST

V.FAST foi o “codinome” dado ao protocolo para comunicação a 28.8 kbps (kbps significa kilobits por segundo) antes dele ser discutido e aprovado pela ITU-T. Ou seja, até junho de 1994 falava-se do protocolo V.FAST, em desenvolvimento, até aparecer o nome definitivo para o mesmo, que seria V.34. Ou seja, um é nome do protocolo na fase de desenvolvimento, não ainda o nome oficial. Muitos chamaram o V.FAST também de V.LAST, pois diziam que as linhas telefônicas comuns não agüentariam mais que isso, por isso seria o último (=LAST) dessa geração.

- V.FC

Existe ainda o V.FC, protocolo que o fabricante de chips para modems Rockwell criou antes do V.34 ser regularizado pela ITU-T. É um protocolo proprietário (não é padrão mundial da ITU-T) para comunicação a 28.8 kbps e que teve muitos seguidores. Era a época de desenvolvimento do V.34 (final de 1993) e a Rockwell achou que a outorgação do mesmo estava demorando muito. Até a USRobotics a seguiu e o implementou em seus modems.

- Desenvolvimento até o V.34

Até setembro de 1994 o padrão para 28.8 kbps era praticamente o V.FC, pois o V.34 ainda não estava totalmente regularizado (já estava aprovado pela ITU-T, mas ainda devia ser aprovado pelos países membros da ITU por votação). Muitos fabricantes de modems de 28.8 kbps com o V.FC prometeram um upgrade (atualização) para o padrão V.34 quando este estivesse regularizado. Alguns exigiram troca de hardware, outros upgrade via software.

Assim em setembro de 1994 o padrão V.34 foi finalmente aprovado em definitivo, e começou a ser fabricado e usado mundialmente.

Hoje em dia um modem 28.8 kbps que não tenha o protocolo V.34 está praticamente obsoleto. Existem modems com os dois protocolos V.34 e V.FC (caso do USR Courier 28.8 kbps dual standart), mas o V.FC certamente cairá em desuso.

### I.[3.2] O que e quais são os protocolos de correção de erros? (V.42 e MNP 2-4)

Em transmissões a altas velocidades, não é raro que as chamadas “sujeiras na linha” atrapalhem a conexão. Essas sujeiras nada mais são do que conexões telefônicas “sujas”, que evitam que o mesmo som que um modem produziu seja recebido pelo outro, ou seja, quando o lado que recebe for demodular o que chegou, não será o que o outro lado transmitiu, o que é demonstrado na conexão com a chamada “sujeira na tela”. Em modems de baixa velocidade a tolerância para sujeiras é maior, mas em alta velocidade, é necessária uma precisão muito maior para resultados adequados.

E é por isso que foram desenvolvidos protocolos que cuidam de monitorar a transferência de dados e que conseguem filtrar fora essa sujeira, que se manifesta na conexão com caracteres aleatórios na tela.

Os protocolos V.42 e MNP 2-4 cuidam da linha telefônica filtrando essas sujeiras. Quando dois modems estabelecerem conexão usando um protocolo de correção de erros, a conexão se dará de forma totalmente limpa, sem erros na tela. Note que a sujeira na linha continua presente, só que não transparece na conexão (na tela).

O processo de filtro usado pelo V.42 e MNP 2-4 é um esquema de correção de erros baseado em algoritmos sofisticados para garantir que os dados que chegam são os mesmos que foram enviados pelo outro lado. Caso os dados não correspondam, o bloco é reenviado. É por isso que algumas vezes a conexão a altas velocidades é interrompida brevemente: É a correção de erros em ação.

O protocolo V.42 utiliza o LAP-M (Link Access Procedure for Modems) como esquema primário de correção de erros e inclui o MNP 4 como esquema secundário. Um modem com V.42 automaticamente possui MNP 4, e é capaz de conectar com um modem dos dois tipos.

É portanto altamente recomendado deixar a correção de erros ativada durante uma conexão.

### I.[3.3] O que e quais são os protocolos de compressão de dados?

#### - V.42bis e MNP5

Estes protocolos permitem uma compactação de dados antes do envio, e a descompactação do outro lado, tudo “on-the-fly”, ou seja, ao mesmo tempo em que está sendo enviado.

A utilização ou não desses protocolos é negociado na hora da conexão e estes permanecem válidos até a posterior desconexão.

Com o protocolo V.42bis pode-se atingir uma compactação de até 4:1 (4 para 1) se o arquivo transmitido for altamente compactável. Neste caso, a taxa de transferência em modems 14400 bps passaria dos normais 1440 cps para até 5760 cps.

Como arquivos altamente compactáveis entendem-se arquivos-texto, planilhas, executáveis (não compactados), bancos de dados, etc.

#### - Diferenças entre V.42bis e MNP5

Mas o que acontece quando se transmite algum arquivo que já esteja compactado? Um protocolo que tem que atuar “on-the-fly” logicamente não conseguirá comprimir este arquivo mais ainda. Aí está a grande diferença entre MNP5 e V.42bis: No MNP5, ele não percebe nada, e tenta compactar mesmo assim, fazendo com que a transmissão seja mais lenta do que o normal. Com o V.42bis, ele percebe que os dados já estão compactados, e se auto-desativa. Com isso não se perde nada em performance.

### I.[3.4] O que são modems RPI e por que são mal vistos?

Normalmente os protocolos V.42 e V.42bis (correção de erros e compressão de dados) são implementados via HARDWARE, ou seja, o próprio modem se encarrega disso.

Mas existem certas marcas de modems onde isso não é o caso. Quando você compra esse tipo de modem, pode pensar que ele possui o V.42 e o V.42bis, o que normalmente é informado na caixa, mas não repara no escrito “RPI MODEM”, “RPI COMPILANT” ou algo parecido.

RPI é a sigla de “Rockwell Protocol Interface” e basicamente significa que o modem depende de que a correção e compressão sejam feitos por algum software padrão RPI.

Esses modems normalmente vem com os seus softwares de comunicação, que são justamente padrão RPI e que são capazes de ativar esses protocolos e os usar. Mas assim que se deseje utilizar outro tipo de programa de comunicação, não mais se pode utilizar o V.42 e V.42bis do modem, pois a maioria deles não traz suporte para RPI.

A diferença básica de um modem RPI para um não-RPI reside no fato de ONDE o software dos protocolos V.42 e V.42bis são executados. No caso dos modems não-RPI, isto é feito pelo próprio modem, no caso dos modems RPI, isso deve ser feito pela CPU do computador.

Assim, além da desvantagem de falta de programas de comunicação, tem-se ainda o fato de estar se gastando ciclos do processamento do micro para a correção de erros, o que pode ser fatal em plataformas de multi-tasking.

### I.[3.5] O que são protocolos de transferência de arquivos?

Os modems que se comunicam entre si podem apenas passar BYTES de um lado para o outro. Eles não têm noção sobre arquivos, etc. Para que se possa transferir um arquivo, você deve utilizar um protocolo de transferência de arquivos.

Um protocolo desse tipo é o que define como os bytes serão enviados e interpretados pelo outro lado para formarem um arquivo exatamente do mesmo tamanho e conteúdo do existente no lado que o enviou.

Um protocolo de transferência define a quantidade de bytes que serão enviados em cada BLOCO, como serão tratados erros de transmissão, como um erro é detectado e como o mesmo é corrigido, como um lado fica sabendo o nome do arquivo que está sendo enviado, como é informada a conclusão da transferência, quantidade de arquivos a serem transmitidos, etc.

Um BLOCO é uma sucessão de bytes que são transmitidos sem pausas. Os blocos de transferência em protocolos de transmissão podem variar de 96 a 8192 bytes (8 kb). A cada final de bloco são enviados bytes de controle, que podem servir simplesmente para delimitar os blocos, como também para controle e correção de erro.

Existem muitos protocolos de transferência de arquivos à disposição, alguns são mais rápidos, outros mais confiáveis, alguns com mais recursos, outros com menos. O que importa é que para se utilizar um protocolo de transferência de arquivos, da mesma forma que para os protocolos de modulação, ambos os lados precisam ter este mesmo protocolo à disposição.

### I.[3.5.1] Quais protocolos de transferência existem?

Os protocolos mais comuns que são apresentados em programas de comunicação são os famosos: Xmodem, Ymodem e Zmodem. Existem ainda outros, que veremos mais tarde nessa mesma sessão.

#### - Xmodem

O protocolo Xmodem original foi desenvolvido em 1977 por um programador chamado Ward Christensen. Através dele, os arquivos são transmitidos em blocos de 128 bytes, aos quais sempre é adicionado um byte de controle para verificação de erros. Esse byte extra, chamado de SOMA DE VERIFICAÇÃO, é composto pelos oito bits de ordem inferior da soma dos 128 bytes. Assim o software receptor calcula igualmente essa soma dos bytes que chegaram e compara com a soma da verificação. Se a soma for outra, o receptor requisita uma retransmissão do bloco.

#### - Xmodem-CRC

O protocolo Xmodem-CRC substituiu a verificação de soma por um esquema chamado de verificação de redundância cíclica (CRC). Ela tem a mesma finalidade que a soma de verificação, porém é mais confiável.

#### - Xmodem-1k

Com os modems de alta velocidade, os blocos de 128 bytes se tornaram muito pequenos. Com isso surgiu o protocolo Xmodem usando blocos de 1024 bytes (1 kbyte).

Algumas vezes, o Xmodem-1k é também chamado de Ymodem. No caso disso ocorrer, o Ymodem real é chamado de Ymodem-Batch.

#### - Ymodem

Basicamente ele é a mesma coisa que o protocolo Xmodem, com algumas diferenças: Se a houver muita sujeira na linha, ele é capaz de comutar automaticamente de blocos de 1024 para blocos de 128 bytes. Em blocos menores, é menos provável que ocorra algum erro e a verificação é mais efetiva.

Além disso, os protocolos Ymodem utilizam um bloco de cabeçalho especial no início da transferência, contendo o nome do arquivo, simplificando a tarefa da transferência de arquivos em forma BATCH.

Transferência batch significa nada mais do que transferir diversos arquivos um atrás do outro, numa mesma sessão do protocolo de transferência.

Muitas vezes o Ymodem é também chamado de Ymodem-Batch.

#### - Ymodem-G

Variante do Ymodem que simplesmente não realiza correção de erros na transmissão dos arquivos. Ele confia na qualidade da linha, ou então ele confia na correção de erros do seu modem, transmitindo os dados sequencialmente sem parar. Quando ocorrer um erro ele imediatamente aborta a operação.

- Zmodem

É o protocolo mais usado atualmente. Apresenta basicamente os recursos do Ymodem-Batch, com uma performance maior ainda, blocos de 1024 bytes, transferência batch, e além disso o recurso de CRASH RECOVERY, que possibilita que transferências abortadas no meio possam ser prosseguidas mais tarde, a partir do local onde se parou da vez anterior.

Até os programas de comunicação mais simples devem suportar esse protocolo, pois é o mais normal de se usar.

- ZedZap

O mesmo protocolo Zmodem, com a diferença deste usar blocos de 8192 bytes ao invés dos habituais 1024.

Ele não está disponível em todos os programas de comunicação. Ele se encontra por exemplo no Terminate.

- Protocolos Bidirecionais

Os protocolos abordados até agora só servem para transmitir em UMA direção. Acontece que numa conexão normal de modems, existem DOIS canais de transferência, como visto na figura da sessão [1.0]. Numa transferência unilateral, um dos canais fica sem uso ou é usado somente para efeito de correção de erros. Por isso surgiram protocolos BIDIRECIONAIS, que são capazes de transmitir arquivos para ambos os lados ao mesmo tempo, sem perda de performance.

Exemplos são o HSLINK, Bimodem e o Hydracomm.

- ASCII

O protocolo ASCII é um protocolo de transferência em 7 bits que fazem os 128 primeiros caracteres da tabela ASCII. Esse protocolo não possui controle e correção de erros e usa o XON/XOFF (veja depois) para controle de transmissão. Assim sendo, o XON/XOFF precisa estar ligado neste caso. Quando o computador que receber os dados precisa parar a transmissão temporariamente, ele envia um CTRL-S (ASCII #17) para parar o envio. Quando ele estiver pronto para voltar a receber dados, envia um CTRL-Q (ASCII #19) para continuar a transmissão.

Esse protocolo não deve ser usado praticamente nunca, só para enviar ou receber rapidamente textos que aparecerem no terminal.

- Kermit

Kermit é um protocolo que é usado extensivamente na Internet. É um protocolo um tanto antigo, que possui uma vasta gama de opções mas que normalmente não tem boa performance.

Ele usa tamanhos de pacotes variáveis, com um máximo de 1024 bytes.

Como o Ymodem, traz suporte para transferências batch.

### I.[3.5.2] Quais são os melhores e os piores protocolos de transferência de arquivos?

Em casos normais você deve usar o protocolo Zmodem. Ele é adequado tanto para modems de alta velocidade quanto para modems sem correção de erros.

Mas é um fato que o Ymodem-G é mais rápido que o Zmodem normal. Só deve ser usado se seu modem faz correção de erros. Veja porque ele é mais rápido:

O Zmodem manda blocos de 1024 bytes por vez. Após mandar um bloco, ele ainda precisa fazer a comparação com o CRC do bloco que foi enviado com o que chegou. Com isso, além do bloco em si, estão sendo mandados bytes de frames (que indicam o começo e fim do bloco) e blocos CRCs para correção de erros.

O Ymodem-G não faz correção de erros. Por isso ele não precisa mandar os blocos CRC após cada bloco, o tornando ligeiramente mais rápido.

Se o seu modem possuir correção de erros, você pode utilizar o Ymodem-G para maior velocidade (a diferença é mínima). O que então ocorre é que falta o recurso utilíssimo de crash recovery a ele. Assim sendo, se uma transmissão com o Ymodem-G for abortada, para continuar da posição onde se parou, deve-se utilizar o Zmodem.

Uma outra opção é o ZedZap. Enviando blocos de 8192 bytes de cada vez, os frames e correção de erros são feitos para os blocos como um todo, assim são menos dados adicionais que precisam ser enviados na transferência de um arquivo.

Acontece que se ocorrer alguma falha na transmissão de um bloco, mesmo que já tenham chegado 8000 bytes corretos, o bloco inteiro precisa ser remandado, tornando a transferência ainda mais demorada do que com o Zmodem (onde no máximo 1024 bytes precisam ser reenviados). Se um modem possuir correção de erros e ambos os lados tiverem o ZedZap à disposição, este protocolo pode ser o indicado. Para modems SEM correção de erros (os de 2400 bps, por exemplo), nunca se deve tentar utilizar esse protocolo.

Resumindo, os protocolos que devem ter chances de serem usados atualmente são o Zmodem, ZedZap (Zmodem 8Kb) e o Ymodem-G.

Se você possuir um modem com correção de erros confiável (você NUNCA vê sujeira na tela), use o protocolo ZedZap se necessitar de velocidade e do recurso de crash recovery ou o protocolo Ymodem-G se necessitar de velocidade e não de crash recovery.

#### I.[4.0] O que são comandos? Como usá-los?

Para você se comunicar com o modem e dar-lhe instruções, você deve usar os chamados COMANDOS AT. Comandos são certas instruções que podem ser passados para o modem. Ele os interpretará retornando a informação requisitada, ou a confirmação de correto recebimento do comando ou de erro na sintaxe do mesmo (“OK” ou “ERROR”).

- O que é padrão Hayes?

Os comandos que podem ser usados nos modems variam de acordo com o modelo e fabricante, mas normalmente seguem o chamado PADRAO HAYES. Hayes é uma empresa fabricante de modems que se tornou conhecida por ter modelos de modems que chegaram como pioneiros no grande mercado, os modems Hayes Smartmodem 1200 e 2400. Nestes usavam-se certos comandos, e eles se tornaram quase que um padrão para os modems futuros que viriam.

- Como enviar comandos ao modem?

Os comandos podem ser enviados ao modem, através de um programa de comunicação, de um modo manual ou transparente ao usuário (sem que ele precise digitá-los).

No primeiro caso, para enviar comandos manualmente para o modem, deve-se estar em algum meio onde se esteja em direta comunicação com ele. É o caso dos TERMINAIS dos programas de comunicação. Normalmente são telas vazias, com o cursor piscando no canto superior esquerdo. Qualquer coisa digitada ali será enviada e interpretada diretamente pelo modem. Experimente digitar “AT” + <ENTER>. O modem deve responder “OK”, se ele estiver preparado e operante.

No segundo caso, os comandos são enviados através de opções existentes no programa de comunicação, como por exemplo através da init string, dial string, etc.

Normalmente programas de comunicação oferecem uma configuração chamada INIT STRING (= seqüência de inicialização). Ali devem ser colocados comandos que devem ser enviados ao modem assim que se entrar no programa de comunicação, é uma forma de configurar o modem e prepará-lo para as conexões subseqüentes.

Outras configurações de programas de comunicação que fazem uso dos comandos do modem normalmente são o DIAL STRING (= comando para discagem) e HANGUP STRING (= comando para desconectar o modem de uma ligação).

- Quais são os comandos mais comuns?

“AT” é o principal comando que deve ser enviado ao modem praticamente sempre antes de um outro comando. Serve para chamar a atenção do modem, informando-o que o que vem a seguir é uma seqüência de comandos que ele deve interpretar.

Os manuais dos modems normalmente trazem uma lista dos comandos possíveis para este determinado modelo. Alguns comandos se tornaram padrão entre todos os modems, como:

“ATD” - Para tirar o modem “do gancho” e mandá-lo discar o número

que vier a seguir. Ex: “ATD884-2446”. Veja também a sessão II.[6.0] para maiores detalhes neste comando.

“ATZ” - Carrega a configuração previamente salva na memória do modem.

“+++” - Quando você estiver conectado em algum lugar, não poderá enviar comandos ao modem. Caracteres digitados serão somente enviados ao outro lado da conexão. Para enviar comandos ao modem enquanto conectado, deve-se esperar um segundo depois do envio/recebimento do último byte, digitar “+++” e aguardar mais um tempo. O modem deve responder com “OK” e você estará no modo de comandos. Para depois voltar à conexão usa-se o “ATO”. O problema aqui é quando isso não funciona corretamente. Neste caso tenha certeza que de, quando você enviou o “+++”, não só o SEU modem foi para o modo de comandos, mas também o modem do outro lado, pois ele interpretou o “+++” da mesma forma. Para contornar o problema, a maioria dos modems possibilita a troca do caracter de “+” por um outro para simbolizar essa “seqüência de escape” (como é conhecida essa seqüência “+++”).

“ATA” - Tira o telefone do gancho e atende a chamada. Assim, quando alguém ligar e o modem reconhecer os toques no telefone imprimindo a string “RING” em seu terminal, digitando “ATA” seguido de <ENTER> o modem atenderá a chamada a inicializará o procedimento de conexão com o outro modem.

- Como fazer uma conexão de modems durante uma ligação VOZ?

Digamos que você ligou para alguém e está conversando com a pessoa pelo telefone normal. Resolvem então efetuar uma conexão entre os dois modems, mas sem perder esta conexão. Como fazer?

É bem simples, exige só um pouco de sincronismo:

- Uma ponta dá um “ATA<ENTER>” e desliga o telefone.
- A outra ponta dá um “ATX3D<ENTER>” e igualmente desliga o fone.

Assim que você digitar “ATA<ENTER>”, o modem já terá assumido a conexão, e você pode desligar o telefone imediatamente. O mesmo vale para o “ATX3D<ENTER>”. O ideal é ambos digitarem antes os comandos “ATA” e “ATX3D” e fazer em uma contagem regressiva para que ambos digitem o <ENTER> aproximadamente ao mesmo tempo.

## II. CONFIGURAÇÃO

Quando você precisa acertar as configurações de seu programa de comunicação, podem surgir algumas dúvidas para alguns itens, justamente pelo desconhecimento dos significados de cada opção.

- 1.0 Velocidade serial x Velocidade modem (DTE x DCE)
- 2.0 8N1, 7E1, etc
- 3.0 Flow control (RTS/CTS - Xon/Xoff)
- 4.0 Portas de comunicação
- 5.0 Terminais de comunicação
- 6.0 Dial String

### II.[1.0] Velocidade serial x Velocidade modem (diferenças entre DTE e DCE).

Uma opção na configuração do software de comunicação certamente diz algo a respeito de velocidade. Normalmente faz referência a Velocidade Serial, Baud Rate, Speed, etc. Um item relacionado a isso é a opção Lock Port Speed ou simplesmente Lock Port. Veremos aqui como deve ser setada cada opção.

Antes de mais nada, vejamos algumas definições. Devemos distinguir dois tipos de velocidades presentes em uma conexão entre dois modems.

1) Velocidade entre o SEU computador com o SEU modem, essa é a chamada velocidade DTE (Data Terminal Equipment), e na verdade é a velocidade de sua porta serial.

2) Velocidade entre o SEU modem e o modem onde você está conectando, sendo esta velocidade chamada de DCE (Data Communication Equipment). Esta é a velocidade efetiva que ocorre pela linha telefônica. O que aparece quando dois modems se conectam (CONNECT xxxx) é a taxa DCE, que os modems negociaram (nos piipiiii's) para ser o protocolo de modulação desta conexão.

Já o que você ajusta no seu programa de comunicação NÃO é essa taxa, já que ela é negociada pelos modems na hora da conexão, e não é possível evitar que eles o façam (a menos que você diga isso expressamente para seu modem). O que você ajusta é a DTE, ou seja, a velocidade máxima com que a sua porta serial irá receber os dados que chegam.

Ajustando a opção Lock Port Speed para NÃO, isso fará com que a velocidade serial (DTE) seja sempre a mesma do que a velocidade entre os modems pela linha telefônica (DCE). Ajustando essa opção para SIM, fará com que a velocidade serial permaneça a mesma independente do que o modem fizer. Na opção de velocidade serial, deve-se então colocar um valor mais elevado para modems de alta velocidade, como por exemplo 57600 bps em modems de 14400 bps.

- V.42 e MNP 4 ajudando na performance

Quando uma conexão é feita com V.42 ou MNP4, a transferência ocorre de modo um pouco diferente do que seria a de enviar os bytes em seqüência. Um byte tem 10 bits (1 start bit, 8 dados e 1 stop bit, isso na configuração 8N1, o que na verdade quer dizer: 8 bits de dados, sem bit de paridade e um start bit. Se um dos protocolos V.42 ou MNP4 estiver ativo numa conexão, o modems que envia transmite apenas os 8 bits de dados, os dois restantes são adicionados pelo modem que recebe, fazendo com que 20% dos dados (2 de 10 bits) a menos tenham que ser enviados.

Quando o modem que recebe adicionar esses 2 bits, eles tem que ser transmitidos junto com os 8 restantes através da porta serial, e o modem já se prepara para receber os novos bits que estiverem chegando. Pra isso, o modem tem que mandar os 10 bits na mesma velocidade que chegaram os 8 bits pela linha telefônica, para a porta serial. A 14400 bps (bits por segundo) 8 bits chegaram em 1/1800 segundos. Nesse mesmo tempo 10 bits tem que ser mandados pela porta serial, isso dá uma taxa de 18000 bps (bits por segundo) que a porta serial tem que suportar.

Por isso na maioria dos programas de comunicação nem aparece a opção de 14400 bps como velocidade serial, a próxima

depois dos 9600 bps normalmente é 19200 bps, que seria o ideal para o caso das conexões com o V.42 ou MNP 4.

- V.42bis e MNP 5 aumentando ainda mais a performance

Quando um protocolo de compressão de dados on-the-fly (V.42bis ou MNP5) está ativo e se recebe arquivos não compactados, o modem que envia vai conseguir compactar esses dados e assim ter que enviar bem menos bytes. Vejamos um exemplo do que ocorre neste caso:

Um arquivo TEST.TXT tem 3072 bytes de tamanho. O protocolo V.42bis consegue compactar ele digamos para somente 1024 bytes, ou seja, compactação no fator 3:1.

Esses 1024 bytes são então enviados pelo modem pela linha telefônica. Assim, se a conexão for de 14400 bps, transmitindo pela linha telefônica a 1600 bytes por segundo (cps), esses 1024 bytes chegam do outro lado em 0.64 segundos (o cálculo é uma simples regra de três).

Mas esses mesmos 1024 bytes serão descompactados por sua vez pelo modem que recebe, para os 3072 bytes originais que por sua vez são transmitidos do modem pela entrada serial para o computador (lembre-se que existe a ligação modem-modem e modem-serial, DTE e DCE).

Assim, em 0.64 segundos, 3072 bytes (24576 bits) tem que ser enviados do modem para o computador pela entrada serial. Isso quer dizer que a taxa real entre o modem e a serial é de 38400 bps (bits por segundo) e isso tem que ser suportado pela linha serial.

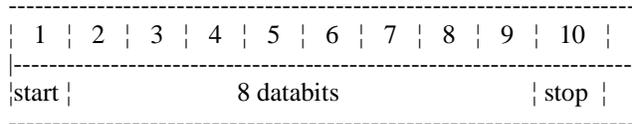
Por isso é ideal setar a velocidade serial para o maior possível.

Uma placa serial normalmente não agüenta mais de 57600 bps, por isso essa deve ser a opção preferencial.

## II.[2.0] O que é 8N1 e 7E1?

Outra configuração que se deve ajustar em praticamente todos os programas de comunicação são esses valores estranhos. Eles simbolizam como será a troca de bytes pela linha telefônica, como interpretar os bits que chegam e que vão.

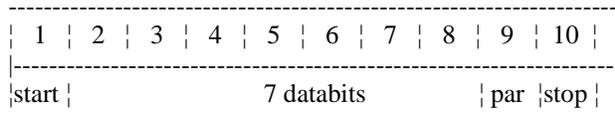
8N1 significa 8 databits, parity NONE e 1 stop bit. Traduzindo para uma linguagem mais clara, quer dizer que um byte transmitido terá o formato:



O primeiro bit é sempre “0” e é interpretado como um bit de início do byte. Os 8 seguintes são o byte propriamente dito, seguido pelo stop bit que é sempre “1”.

Com esses 8 bits de dados, forma-se um byte que conhecemos, o que aceita valores de 1 a 256.

O formato 7E1 simboliza 7 data bits, EVEN parity e 1 stop bit.



Um start bit (“0”), 7 bits de dados, podendo representar 128 valores, um bit de paridade e um stop bit (“1”).

O bit de paridade EVEN é um bit de controle. Ele é simplesmente o um valor (1 ou 0) para que a soma dos bits “1” dos dados mais o bit de paridade dê um número par (even). Por exemplo se os databits forem “0110011”, o bit de paridade é ajustado para “0”, pois já existem 4 “1” nos dados. Se os databits fossem “1110011”, o bit de paridade seria ajustado para “1”, para chegarem a 6 (=par) bits com valor “1”.

A paridade ODD é justamente o inverso, a soma dos bits “1” deve dar um número ímpar.

Na paridade tipo Mark, o bit de paridade é sempre “1” e na paridade tipo Space, este bit é sempre “0”.

- Qual usar?

Para conexões com BBSs comuns, use sempre a configuração 8N1. Ela é usada em 90% dos casos atualmente. Por exemplo para conexões com a CompuServe, deve-se ajustar a configuração para 7E1.

## II.[3.0] Como configurar o flow control (RTS/CTS - Xon/Xoff)?

Uma configuração importante para a transmissão de dados é o FLOW CONTROL.

Flow control é um método de controlar quando informação pode ser enviada. Se isso não fosse possível de se controlar, o modem poderia ficar sobrecarregado de informações que estivessem chegando enquanto ele está ocupado com outra coisa.

Um método é o SOFTWARE FLOW CONTROL, ou XON/XOFF, onde um BBS irá mandar dados até que o seu computador envie um sinal de XOFF (que é o caracter ASCII #17 - CTRL-S). Enviando um XON (caracter ASCII #19 - CTRL-Q), ele voltará a transmitir.

Outro método é o HARDWARE FLOW CONTROL ou CTS/RTS. Este caso funciona alterando-se voltagens em dois níveis nos pinos de RTS (Request To Send) e CTS (Clear To Send) da interface serial entre o modem e o computador (normalmente um cabo RS232).

CTS é usado pelo modem que está enviando dados. Quando o modem local está pronto para receber dados, ele envia um sinal CTS para o computador local, e este começa a mandar os dados para o modem. Quando o modem não está mais conseguindo receber os dados na velocidade com que o computador os está enviando, o modem irá desligar o CTS, informando ao computador que é para ele parar de enviar. Uma vez o modem está novamente em condições de receber dados, ele novamente liga o CTS.

RTS é usado pelo computador do lado que está recebendo dados. Quando o computador local não consegue receber dados na velocidade que eles estão sendo enviados para ele pelo modem local, ele irá desabilitar o RTS. O computador liga o RTS novamente quando está pronto para receber dados novamente.

- Qual usar?

Escolha sempre preferencialmente o HARDWARE FLOW CONTROL (CTS/RTS). O uso do XON e XOFF durante a transmissão de dados por causar problemas quando um arquivo binário conter os caracteres XON e XOFF. Neste caso a transmissão pode travar durante a transferência de arquivos.

## II.[4.0] O que é porta serial?

O computador precisa saber em qual PORTA SERIAL está conectado o modem para poder se comunicar com ele. As portas seriais em micros PC são chamadas de COM1, COM2, COM3 e COM4.

Se o seu modem é externo, a porta serial é onde está ligado o modem pelo cabo RS-232. Se o modem for interno, normalmente existe a opção de configurar em qual porta serial se deseja o modem por meio de JUMPERS, que são pequenas pontes de ligações que podem ser ligadas ou desligadas.

Embora se tenha quatro portas de comunicação à disposição, existem algumas considerações a se fazer quanto à correta configuração:

Cada porta COM possui uma chamada de interrupção que ela usará para informar ao micro que está precisando de atenção. Essas interrupções são chamados de IRQ, e são numerados de 0 a 15. Para as portas seriais, existem dois IRQs:

```
porta IRQ
-----
COM1  IRQ4
COM2  IRQ3
COM3  IRQ4
COM4  IRQ3
```

Assim sendo, não é recomendado ter dois periféricos em duas portas com IRQs iguais, ou seja, ao mesmo tempo na COM1 e COM3 ou na COM2 e COM4.

Se o modem for INTERNO e se puder configurar à gosto a porta serial a usar, devem ser seguidos algumas recomendações:

COM1: Use essa porta se seu micro não tiver uma conexão serial já como COM1 (o que não é provável). É comum de se ter um mouse instalado na COM1.

COM2: Se seu micro estiver equipado com somente uma porta serial na COM1, e se você não já está usando a COM2 para outro equipamento, use essa porta (é a configuração mais comum e os modems normalmente vêm com a COM2 por default).

COM3 ou COM4: Somente em casos especiais, normalmente não é o caso.

Além de poder configurar a porta serial, os modems internos também permitem configurar o IRQ a ser usado, sendo que assim você pode contornar o problema dos IRQs exposto acima.

Cuidado ao configurar um modem interno quando já houver uma placa I/O (entrada e saída) interna no seu micro. Normalmente é uma placa de multi-uso, oferecendo saídas seriais, paralelas (para impressoras) e para joystick. Essas placas normalmente podem ser configuradas por meio de jumpers para oferecer determinadas combinações de portas seriais, como por exemplo COM1 e COM2. Quando o modem interno for instalado, deve se tomar cuidado para não se instalar ele numa porta serial já ocupada pela placa de I/O. Use o programa MSD que vem com o MS-DOS e o MS-

Windows e veja o item <C>OM PORTS. Use no seu modem interno somente portas seriais marcadas como “N/A” nessa tela, tomando ainda as precauções do IRQ.

## II.[5.0] O que são terminais de comunicação?

Quando é efetuada uma conexão com um BBS, este normalmente começa enviando caracteres para seu modem, que são apresentados na tela. Ele poderia enviar simples caracteres texto que o seu programa de comunicação iria apresentando, mas convenhamos, uma apresentação só de texto não é muito atraente.

Para isso surgiram diversos tipos de terminais de comunicação que serviriam para interpretar o que chega pelo modem de uma outra forma, apresentando o que chega de maneira diferente da de só texto.

- Como funciona o terminal ANSI?

A forma mais popular que surgiu foi o padrão ANSI de cores e movimentação de cursor, também usado em outras áreas no computador. Neste padrão o que acontece é simplesmente que o que é enviado por um modem é interpretado como texto normal, com uma exceção: Quando chegar a combinação de caracteres “<ESC>[” (os sinais ASCII #27+#91) o terminal receptor aguarda os próximos caracteres, que serão alguma espécie de comando.

Estes comandos podem ser os mais diversos, e vão modificar a apresentação na tela. Por exemplo, pode-se mudar a cor de frente e de fundo dos próximos caracteres, limpar a tela, mover o cursor para outra posição, etc. Em alguns casos, pode-se até tocar música, mas são somente alguns os terminais que suportam esses comandos para música (um exemplo é o Terminate).

- Que outros terminais existem?

Existem mais alguns tipos de terminais que podem ser usados ao invés do ANSI. Os mais comuns além dele são o AVATAR e o VT-100.