

## Comunicação Dados

A **Comunicação Dados** é uma disciplina da área de “Ciências da Computação” que trata da **transmissão** de informação entre sistemas computacionais e dispositivos diferentes através de um meio de transmissão.

A transmissão de informação pressupõe a passagem de sinais através dos meios físicos de comunicação que compõem as redes.

### Quanto a Eficiência

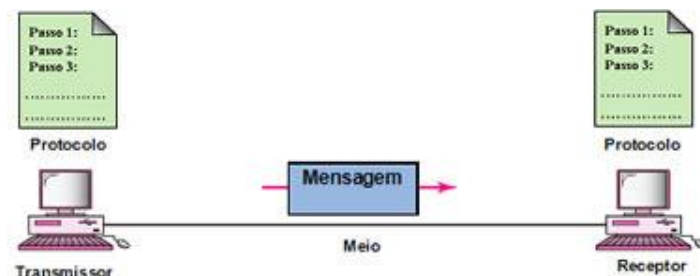
A eficiência de um sistema de comunicação de dados, é necessário conhecer 3 fundamentais características: Serviço de Entrega, Confiabilidade e Tempo de Atraso.

- **Entrega (ou *delivery*):** Os dados devem ser recebidos somente pelo dispositivo ou usuário de destino, ou seja, o sistema deve entregar os dados ao destino correto;
- **Confiabilidade:** Dados modificados ou corrompidos em uma transmissão são pouco úteis, portanto, o sistema deve garantir a entrega dos dados;
- **Tempo de atraso:** Dados entregues tardiamente são pouco úteis. Por exemplo, no caso de transmissões multimídia, como vídeo, os atrasos não são desejáveis, de modo que eles devem ser entregues praticamente no mesmo instante em que foram produzidos, isto é, sem atrasos significativos. Neste caso, o sistema deve entregar dados em um tempo predeterminado e evitar ao máximo os atrasos.

### Quanto aos Componentes

Para que a comunicação seja realizada, o processo é composto de elementos básicos:

- **Mensagem:** é a informação a ser transmitida. Pode ser constituída de texto, números, figuras, áudio e vídeo – ou qualquer combinação desses;
- **Transmissor (TX):** é o dispositivo que envia a mensagem de dados. Pode ser um computador, uma estação de trabalho, um telefone, uma câmera de vídeo e assim por diante;
- **Receptor (RX):** é o dispositivo que recebe a mensagem. Pode ser um computador, uma estação de trabalho, um telefone, uma câmera de vídeo e assim por diante;
- **Meio de Transmissão:** é o caminho físico por onde viaja uma mensagem originada e dirigida ao receptor;
- **Protocolo:** é um conjunto de regras que governa a comunicação de dados. Ele representa um acordo entre os dispositivos que se comunicam.



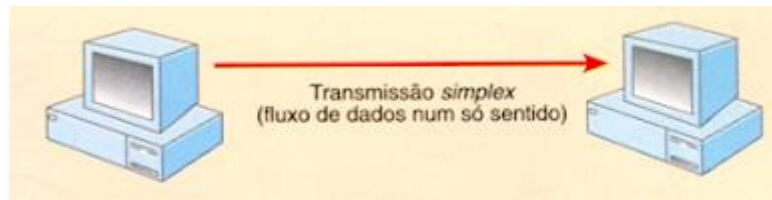
## Transmissão de Dados

A **Transmissão de Dados** é uma matéria cada vez mais importante para qualquer pessoa que opere com equipamentos que estejam inseridos num sistema de comunicação, que explora maneiras ou técnicas, através das quais as informações são transmitidas. Para uma transmissão dada numa via de comunicação entre duas máquinas, a comunicação pode ser realizada de formas diferentes, definindo-se por: Sentido da Transmissão; Modo de Transmissão e Sincronização.

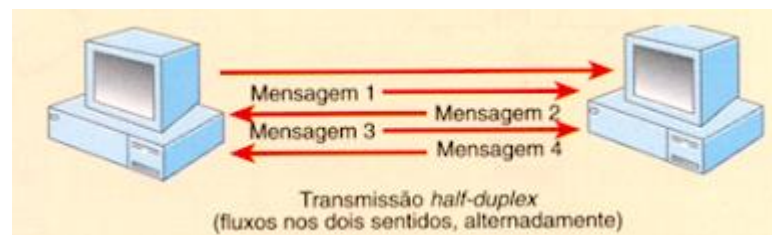
## Sentido da Transmissão

O sentido de transmissão (ou sentido das trocas) entre dois dispositivos em redes pode acontecer de três maneiras diferentes: **Simplex**, **Half-duplex** ou **Full-duplex**.

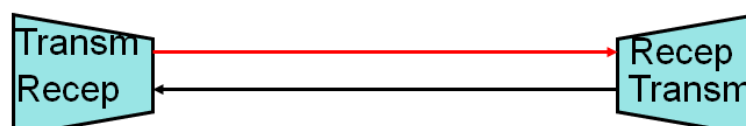
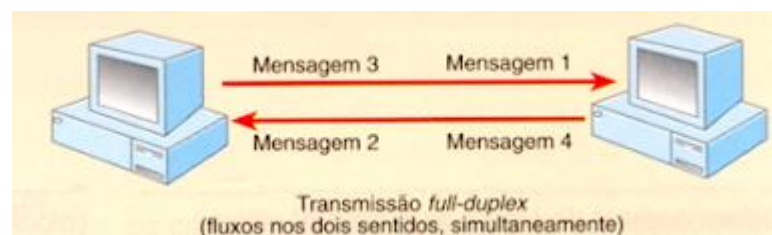
- **Simplex:** A comunicação simplex é aquela em que há somente um transmissor e um receptor. A comunicação é unidirecional, como em uma rua de mão única. Somente um dos dois dispositivos no link é capaz de transmitir, logo o outro só será capaz de receber. Como exemplo temos a transmissão de TV e rádio AM e FM onde apenas podemos receber os dados enviados pelo receptor e não interagimos com o sistema.



- **Half-duplex:** Neste modo, cada estação pode transmitir e receber, mas nunca ao mesmo tempo. Quando um dispositivo está transmitindo o outro está recebendo e vice-versa. Em uma transmissão half-duplex, toda a capacidade do canal é dada ao dispositivo que estiver transmitindo no momento. Como exemplo temos a comunicação usada por exemplo por um *Walk Talking* assim como o rádio Nextel.



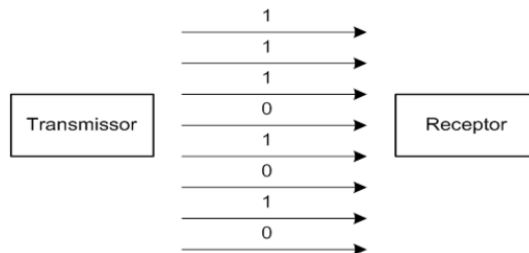
- **Full-duplex:** Neste modo, ambas estações podem transmitir e receber simultaneamente, ou seja, ao mesmo tempo. Sinais em direções opostas compartilham a capacidade do link ou canal. Como exemplo temos a comunicação através de um telefone celular, onde conseguimos falar e ouvir ao mesmo tempo.



## Modos de Transmissão

O **modo de transmissão** designa o número de unidades elementares de informações (bits) que podem ser transmitidas simultaneamente pelo canal de comunicação, ou seja, trata diretamente, a quantidade de bits a ser transmitida ao mesmo tempo.

- **Transmissão em modo Paralelo:** Na transmissão em modo paralelo, os bits que compõem o carácter são enviados simultaneamente através de várias vias de dados. Uma via é, por exemplo, um fio, um cabo ou qualquer outro suporte físico. A ligação paralela dos computadores de tipo PC necessita geralmente de 10 fios. A imagem abaixo, exemplifica o modo de transmissão.



Exemplos: LPT1, LPT2, IDE, SCSI, PCI.

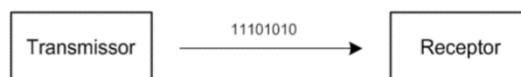


Estas vias podem ser:

1.  $N$  linhas físicas: neste caso, cada bit é enviado para uma linha física (é a razão pela qual os cabos paralelos são compostos de vários fios em cobertura);
2. Uma linha física dividida em vários subcanais compartilhando a mesma banda. Assim, cada bit é transmitido numa frequência diferente.

Dado que os fios condutores estão próximos numa cobertura, existem perturbações / interferências que degradam a qualidade do sinal.

- **Transmissão em modo Série:** Na transmissão em modo série, os bits que compõem a informação são enviados um a um através de uma única via de dados.



Exemplos: USB, Sata (Serial Ata), SaS, PCI-Express.



## Quanto a Sincronização

Dados os problemas com a transmissão paralela, é a em modo série que é mais utilizada. Entretanto, como é apenas um só fio que transporta a informação, existe um problema de sincronização entre o emissor e o receptor, ou seja, o receptor não pode a priori distinguir os caracteres (ou mesmo, de maneira mais geral, as sequências de bits) porque os bits são enviados sucessivamente.

Existem então dois tipos de transmissão que permitem remediar este problema: **Síncrona e Assíncrona**.

- **Transmissão Assíncrona:** No modo de transmissão Assíncrono os dados são enviados um a um sem controle de tempo entre um e outro. Agora, imagine que só um bit é transmitido durante um longo período de silêncio, onde o receptor não poderia saber que se trata de 00010000, ou 10000000 ou ainda 00000010. Para remediar este problema, cada dado é precedido de uma informação que indica o início da transmissão deste (a informação de início de emissão chama-se bit START) e termina com o envio de uma informação de fim de transmissão (chamada bit STOP, pode eventualmente haver vários bits STOPS). Normalmente utilizada quando não é estabelecido, no receptor, nenhum mecanismo de sincronização relativamente ao emissor.

Características:

- Baixo Rendimento (alto *overhead*).
- Fácil Implementação;
- Baixa Velocidade;
- **Transmissão Síncrono:** Na transmissão em modo Síncrono os dados são enviados em blocos e em intervalos de tempo definidos, dados de sincronismo são enviados durante a transmissão para manter o sincronismo entre as máquinas. O receptor recebe continuamente (mesmo quando nenhum bit é transmitido) as informações ao ritmo em que o emissor as envia. É por isso é necessário que emissor e receptor estejam sincronizados à mesma velocidade. Além disso, informações suplementares são inseridas para garantir a ausência de erros na transmissão.

Características:

- Boa qualidade de transmissão;
- Custo de transmissão mais elevado;
- Equipamento mais sofisticado;
- Ideais para transmissão de sinais sensíveis a atraso (voz, música, vídeo);
- Transmissão com maior confiabilidade;
- Adequado para aplicações multimídia.

## Métodos para Operações de Entrada e Saída

### 1. E / S programada:

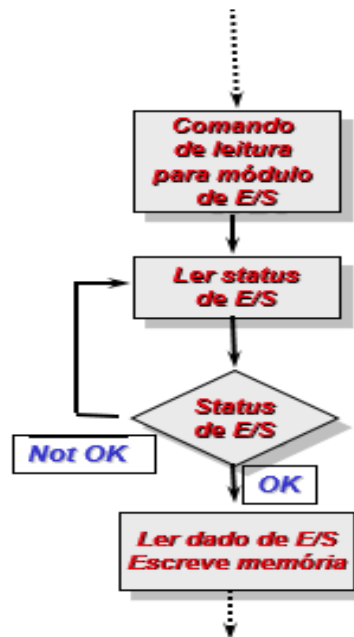
A CPU controla diretamente todas as etapas da comunicação. O programa deve verificar so dispositivos de entrada e saída e para o processamento durante a transmissão. O tempo de transmissão dos dispositivos de entrada e saída são longos se comparados ao  $\mu$ computador. Processo muito pouco eficiente.

É devido ao resultado das instruções de E / S que estão escritas no programa de computador. Cada transferência de item de dados é iniciada por uma instrução no programa. Normalmente, a transferência é de um registro e memória da CPU.

Neste caso, requer monitoramento constante pela CPU dos dispositivos periféricos.

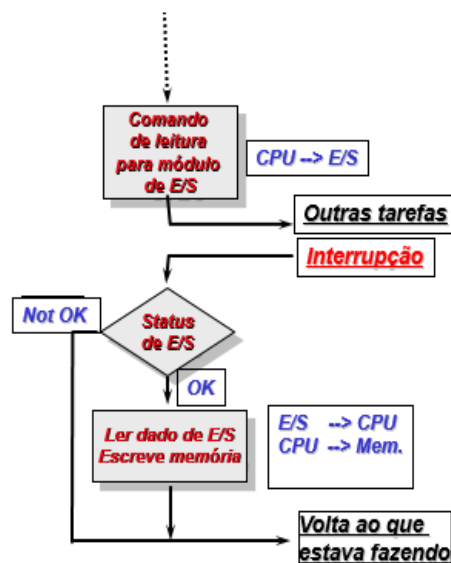
### Exemplo de E/S programada:

O dispositivo de E/S não tem acesso direto à unidade de memória. Uma transferência do dispositivo de E/S para a memória requer a execução de várias instruções pela CPU, incluindo uma instrução de entrada para transferir os dados do dispositivo para a CPU e armazenar instruções para transferir os dados da CPU para a memória. Na E/S programada, a CPU permanece no loop do programa até que a unidade de E/S indique que está pronta para a transferência de dados. O processo demora, pois mantém desnecessariamente a CPU ocupada.



## 2. E / S iniciada por interrupção:

Essa situação pode muito bem ser evitada usando um método controlado por interrupção para transferência de dados. Usando o recurso de interrupção e comandos especiais para informar a interface para emitir um sinal de solicitação de interrupção sempre que houver dados disponíveis de qualquer dispositivo.



A CPU aguarda a interface de entrada e saída requisitar uma transmissão. Enquanto isso o  $\mu$ Processador pode realizar outras tarefas. Quando a interface está pronta para transmissão ela

avisa o  $\mu$ Processador. O  $\mu$ Processador interrompe a atividade corrente e inicia a comunicação com o dispositivo de entrada e saída.

A CPU pode prosseguir para a execução de qualquer outro programa. A interface, entretanto, continua monitorando o dispositivo. Sempre que é determinado que o dispositivo está pronto para transferência de dados, ele inicia um sinal de solicitação de interrupção para o computador.

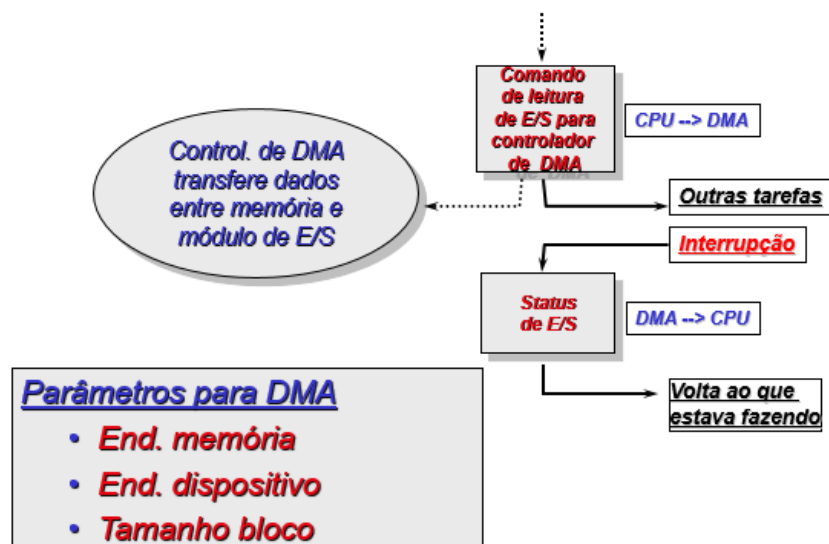
Após a detecção de um sinal de interrupção externa, a CPU para momentaneamente a tarefa que já estava executando, desvia para o programa de serviço para processar a transferência de E / S e, em seguida, retorna à tarefa que estava executando originalmente.

O processo é mais eficiente que a operação por varredura, mas ainda sobrecarrega o  $\mu$ Processador durante a comunicação com o periférico.

1. Atende a acontecimentos Assíncronos (imprevisível);
2. Não precisa esperar que ele ocorra, o  $\mu$ Processador não deixa de ser usado por outras funções;
3. Pode ser interna ou externa;
4. Interna: Divisão por zero, Overflow, e outras;
5. Externa: Interface de Entrada e Saída;
6. Um evento qualquer envia um sinal pedido de interrupção (INTERRUPT REQUEST - IRQ) ao  $\mu$ Processador por meio de uma linha de controle externo do sistema;
7. O  $\mu$ Processador pode aceitar ou rejeitar, gerando um sinal de reconhecimento de interrupção numa linha de controle do barramento externo: INTERRUPT ACKNOWLEDGE - IACK;
8. O  $\mu$ Processador para a execução do programa (via hardware) grava o endereço de retorno (PC+1) na pilha e atende à rotina de interrupção;
9. Uma rotina é um evento síncrono (previsível) - varredura;
10. Após a execução da rotina de interrupção, o  $\mu$ Processador volta ao ponto onde parou no programa principal;
11. Impossível prever o local exato de retorno da interrupção.

### 3. Acesso Direto à Memória:

A transferência de dados entre uma mídia de armazenamento rápida, como disco magnético e unidade de memória, é limitada pela velocidade da CPU. Assim, pode-se permitir que os periféricos se comuniquem diretamente uns com os outros usando os barramentos de memória, removendo a intervenção da CPU.



Esse tipo de técnica de transferência de dados é conhecido como DMA ou acesso direto à memória. Durante o DMA, a CPU fica ociosa e não tem controle sobre os barramentos de memória. É o processo mais eficiente que todos os outros, pois não utiliza o  $\mu$ Processador.

O controlador DMA assume os barramentos para gerenciar a transferência diretamente entre os dispositivos de E / S e a unidade de memória.

- Dispositivo de hardware dedicado à operação de transferência de dados entre um dispositivo de entrada e saída e memória;
- Coloca o Processador em estado de alta impedância (desligado) para permitir a um dispositivo externo o Acesso Direto à Memória - Bus Resquest;
- Acesso Direto à Memória (DMA) permite uma forma mais rápida de mover dados entre portas de entrada e saída e a memória;

### Diferentes configurações de DMA

