

# Organização de Computadores

## Aula 10: Dispositivo de E/S

### INTRODUÇÃO

---



Em nossa última aula da disciplina, vamos concluir de forma integral o estudo da organização de arquitetura dos computadores modernos.

Neste momento, já conhecemos os componentes do sistema de computação, a relação entre os mesmos, bem como a dependência de diversos subcomponentes envolvidos na comunicação da Memória Principal com a CPU. Porém, resta o

entendimento quanto à relação de todo este processo de comunicação junto a dispositivos externos, que conhecemos como dispositivos de entrada e saída.

Perceberemos que de nada adianta ocorrer todo este processamento interno na máquina sem que haja a possibilidade de efetuar entrada ou saída de dados.

Um artigo científico, por exemplo, só pode ser transcrito para um arquivo de editor de textos se houver alguma forma do pesquisador inserir suas anotações e ideias. Uma planta baixa de um projeto arquitetônico só pode ser impresso se houver um dispositivo de impressão que se comunique com o computador.

Enfim, veremos que são necessários dispositivos de entrada e saída de dados para que toda a capacidade de processamento da máquina seja possível, de acordo com a demanda do usuário.

## OBJETIVOS

---



Reconhecer o funcionamento dos dispositivos de entrada e saída de dados.

Diferenciar mecanismos de transmissão paralelo e serial.

Identificar métodos de uso de interrupções na comunicação de dispositivos.

# O que são dispositivos de entrada e saída?

São considerados dispositivos de entrada e saída todos aqueles equipamentos de computação que, sendo interligados ao sistema de computação, possam inserir e extrair informação do mesmo.

Esses dispositivos também fazem parte do modelo de Von Newman e até mesmo do projeto inicial do Calculador Analítico de Charles Babbage e, da mesma forma que os processadores e memórias, têm evoluído constantemente, existindo diversas formas de inserir e extrair informações de um computador.

Vejamos alguns exemplos:

## Dispositivos de Entrada

Como exemplo mais simbólico, pode-se citar o **teclado** como um grande e necessário dispositivo de entrada, afinal, sem o mesmo não seria possível inserir os códigos de programação ou instruções a serem usadas na programação de computadores.

O mesmo para o **mouse**, dispositivo único e ainda muito utilizado, mas que em alguns casos tem sido substituído por telas Touch-Screen, também dispositivos de entrada.



Mouse



Teclado



Webcam



Scanner

### Dispositivos de Saída

---

Como dispositivo de saída, nada melhor do que citar uma impressora como representante ou um monitor de vídeo, que permite a visualização dos resultados obtidos em um processamento ou relatório.



Monitor



Impressora



Alto-falante



Projektor

Independente da finalidade, a função de dispositivos de E/S sempre será receber ou enviar informações ao meio exterior de um sistema de computação, convertendo as informações obtidas (de entrada ou de saída) em uma forma inteligível para a máquina (se estiver recebendo dados) ou para o usuário (se estiver enviando).

Para tanto, cada dispositivo de E/S, assim como as células de memória acessadas pela CPU, precisam ter um endereço computacional para que sejam contatados.

## Exemplo

, Um exemplo a ser citado e talvez conhecido por alguns é que uma Porta Paralela, ainda existente em boa parte dos computadores pessoais, e antigamente muito utilizada para impressoras, possui um endereço específico. Ex: Porta paralela – 378h.

Assim como estudado no modelo de comunicação entre CPU e MP, o envio de sinais de controle, bem como demais informações, era feito por meio de barramentos. Para E/S, também são necessários barramentos de comunicação, pois é através deles que se pode interligar todos os componentes do sistema de computação.

## Tipos de transmissão

Existem basicamente duas formas básicas de se realizar transmissão e recepção de dados entre os periféricos, **CPU e MP**, que pode ser serial ou paralela.

Dependendo da tecnologia ou finalidade de comunicação, cada periférico pode possuir uma dessas formas de comunicação e sua tecnologia respectiva, respeitando a capacidade de transferência entre dispositivos, uma vez que cada periférico possui uma capacidade diferente para transmissão.

### Transmissão serial

Na transmissão serial, o periférico é conectado ao dispositivo controlador (veremos posteriormente) ou interface de E/S por uma única linha de transmissão de dados.

Nesse tipo de transmissão, a informação é transmitida/recebida bit a bit e, por não haver necessidade de controle e gerenciamento como o paralelo, é de fácil controle e implementação.

Por essa facilidade, o uso deste tipo de transmissão, bem como tecnologias disponíveis, têm crescido consideravelmente.

Por exemplo, a porta USB (Universal SERIAL Bus, ou barramento SATA - SERIAL ATA), são tecnologias padronizadas no uso de equipamentos.

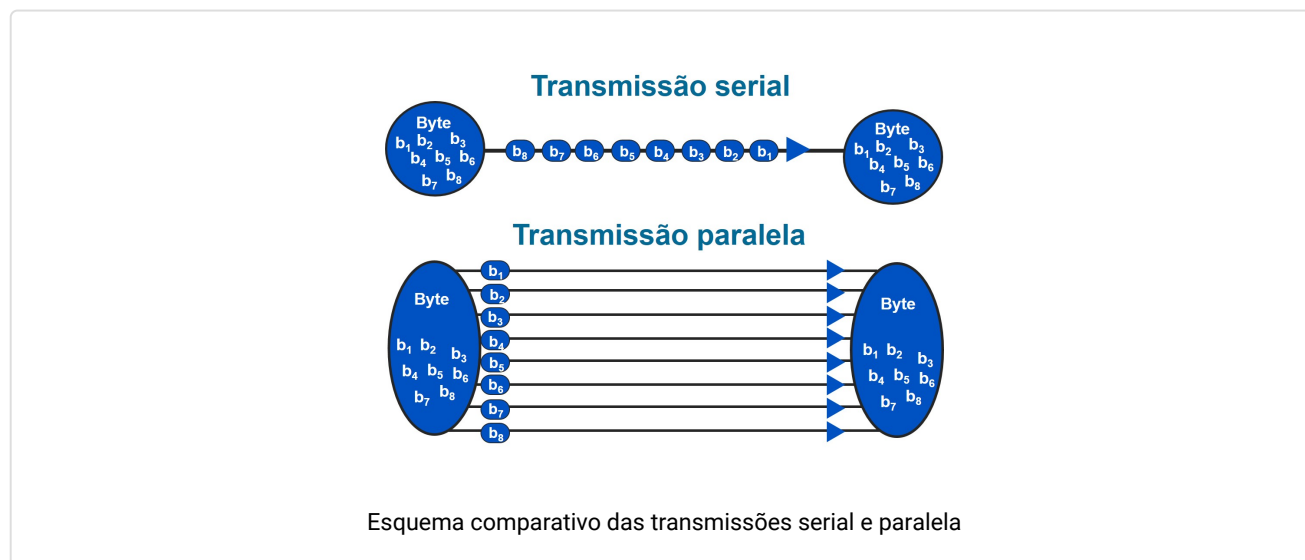
### Transmissão paralela

Na transmissão em paralelo, um grupo de bits é transmitido de cada vez, cada um sendo enviado por uma linha separada de transmissão.

A informação é transmitida/recebida em grupos de bits de cada vez, simultaneamente.

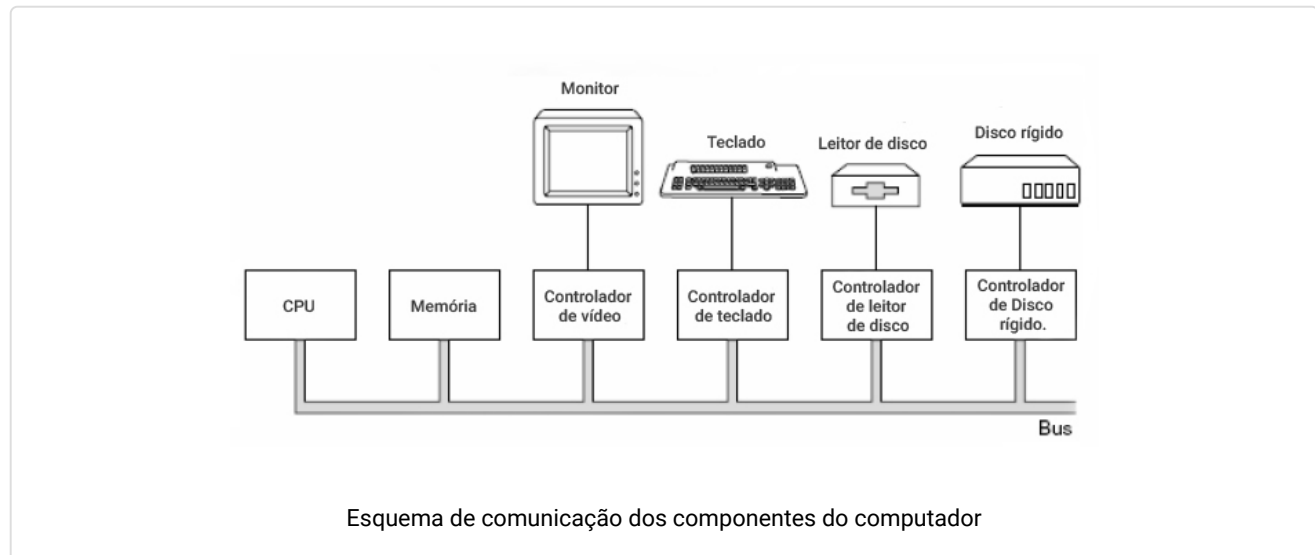
Por ser feita dessa forma, os sinais de cada linha de barramento precisam chegar simultaneamente. Sendo assim, é difícil de controlar, pois os mecanismos de verificação desse "paralelismo" têm de custo elevado.

Entenda melhor por meio da imagem a seguir:



# Controladora de E/S

A CPU e Memória Principal não se comunicam diretamente com o periférico, necessitando de uma interface de comunicação entre CPU/MP e o periférico, conhecidos normalmente como controladoras de E/S, conforme mostra a figura:



Esses dispositivos ficam interligados através de barramento ou “bus” e normalmente possuem unidades de memória (registradores) para comunicação com a CPU, que se comunica com sua controladora, responsável por converter o fluxo de bits em um bloco de bytes, executando também as correções de erros necessárias e tornando os dados disponíveis para serem copiados para a MP.

Sendo assim, as controladoras de E/S são um hardware que controla uma porta, barramento ou dispositivo e sua comunicação com a CPU, ou seja, quando um programa precisa de dados do disco, por exemplo, ele envia um comando ao controlador de disco, que emite um sinal de comando como “seek” à unidade de disco.

## Exemplos

- Controlador da porta serial;
- Controlador de Vídeo;
- Controlador de disco.

# Controladora de E/S - Processo de comunicação

1

Inicialmente, a CPU interroga o dispositivo, enviando o endereço do dispositivo e um sinal dizendo se quer mandar ou receber dados através da controladora;

- 2 A controladora, reconhecendo seu endereço, responde quando está pronta para receber (ou enviar) os dados;
- 3 A CPU então irá transferir (ou receber) os dados através da controladora;
- A controladora responderá à CPU confirmando que:
  - 4
    - Recebeu (ou transferiu) os dados ("Acknowledge" ou ACK); ou
    - Não recebeu os dados, neste caso solicitando retransmissão ("Not-acknowledge" ou NAK).

## Device Drivers

Para que o Sistema Operacional possa se comunicar com novos periféricos e dispositivos interligados ao sistema de computação, é necessário que ele reconheça esses dispositivos.

Dessa forma, podemos entender como Device Driver o software que permite que o sistema operacional e este dispositivo específico possam se comunicar dentro do processamento da CPU.

Normalmente, por se tratar de um novo periférico a ser instalado no sistema de computação, o driver é desenvolvido pela mesma empresa que projetou e fabricou o dispositivo, de forma que o driver serve como complemento a fim de traduzir as requisições de alto nível (feitas pelo usuário) para o dispositivo específico (junto à controladora), podendo também trabalhar junto à controladora de E/S na conversão de dados e detecção e correção de erros, garantindo o correto funcionamento do dispositivo.

## Formas de comunicação entre CPU/MP e interface de E/S

Existem formas de comunicação da CPU e Memória Principal com as interfaces de entrada e saída.

Para qualquer tipo de comunicação, sempre há o armazenamento de conteúdo de dados em memória, a fim de que este conteúdo possa ser trafegado durante o processamento.

Porém, esta comunicação pode se diferir pelo tipo de memória utilizada durante esse processamento, sendo eles:

### Memória compartilhada

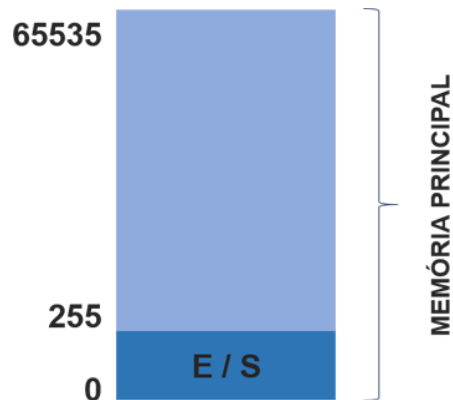
---

Nesse tipo, a memória principal é compartilhada tanto por instruções e dados comuns de um programa quanto por instruções/dados de operações de E/S, sendo necessário somente lançar no barramento de controle o comando de leitura/escrita que deve ser executado, pois a memória principal é justamente a memória utilizada neste tipo de operação.



**Vantagem** Não necessita de instruções especiais para processamento.

**Desvantagem** Ocupa parte do espaço da memória principal para seu processamento.



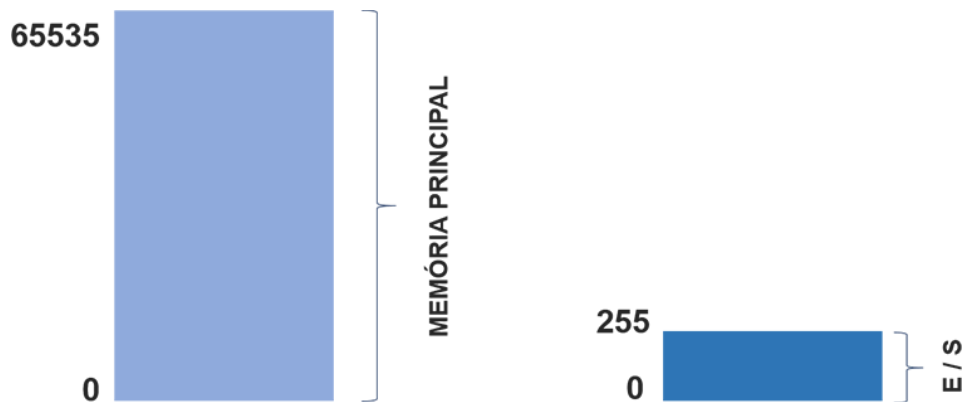
### Memória isolada

Nesse exemplo, existe uma memória isolada. Consiste em criar um espaço de memória próprio de E/S para aquela operação, não utilizando assim uma parcela da memória principal.

Para isso, é necessário um sinal de identificação a fim de saber se a instrução a ser processada é de E/S ou não, pois o local de armazenamento do conteúdo a ser processado estará em local diferente.

**Vantagem** Não utiliza espaço da memória principal.

**Desvantagem** Necessita de instruções especiais.



## Formas de realização de entrada/saída

Agora que estudamos como funciona a comunicação dos dispositivos de entrada e saída em um Sistema de Computação, é importante entender como é realizada a comunicação de entrada e saída nos mesmos.

Existem três tipos principais:

### E/S programada

Nesse tipo, a CPU lê constantemente o status do controlador de E/S e verifica se já acabou o processamento (Polling ou Busy-waiting), espera até o fim da operação para dar continuidade aos demais.

Há um intenso uso da CPU, bem como uma ociosidade de tempo, uma vez que o polling faz com que existam momentos de espera no processamento.

### E/S com emprego de interrupção

Muito utilizada na arquitetura atual, como o próprio nome diz, nesse tipo, a CPU solicita a transferência de um processamento à controladora e desvia-se para outra atividade, sem aguardar que o processamento seja finalizado.

Dessa forma, ela pode se preocupar com outros processamentos até que o mesmo seja finalizado pelo dispositivo de E/S, que, ao finalizar, avisa à CPU, por meio de um sinal de interrupção, que faz com que a CPU “desvie” sua atenção para o resultado do processamento requisitado.

Tipos de interrupção:

**Internas:** Geradas pela execução de uma instrução. Exemplo: Divisão por Zero, etc.

**Externas:** Geradas por um sinal externo à CPU. Utilizadas na comunicação com periféricos.

### Acesso direto à memória (DMA – Direct Memory Access)

Controlador de DMA: Permite a transferência de dados entre uma interface e a MP praticamente sem intervenção da CPU.

CPU solicita transferência. Controlador sinaliza final através de interrupção.

## Tratamento de interrupção (Interrupt Handler)

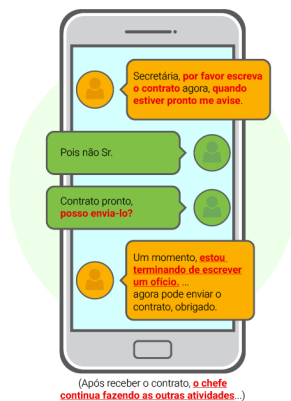
O tratamento de interrupções pode ser resumido pelo exemplo de diálogo abaixo, onde é demonstrada a comunicação SEM interrupção e COM interrupção.

Este exemplo pode diferenciar claramente como o emprego de interrupção interfere e auxilia na otimização do processamento em um Sistema de Computação.

Programa **sem** interrupção



Programa **com** interrupção



É importante reforçar que cada tipo de interrupção tem um tratamento específico a ser feito.

Se estivermos falando de um click do mouse, o "mecanismo de tratamento" vai verificar que o botão foi pressionado e, dependendo do botão (vamos considerar um mouse de 2 botões), uma operação diferente será executada.

Geralmente, a CPU utiliza o código identificador do botão pressionado para um endereço específico de memória e cada programa irá tomar sua decisão do que será feito quando este "click" for efetuado, seja ele um navegador de internet, um jogo, entre outros aplicativos.

Quando esse processo de interrupção pelo mouse é encerrado, o tratamento também é encerrado, e a CPU volta à execução do programa que havia sido interrompido para tratar o "click" do mouse, restaurando todos os valores, salvos antes da interrupção, e retomando exatamente do ponto em que parou.

## Tipos de interrupções

### Maskable interrupt (IRQ)

É uma interrupção de hardware que pode ser ignorada por configurar um bit em um registro da máscara de interrupção (IMR) bit-mask.

### Non-maskable interrupt (NMI)

É uma interrupção de hardware que carece um bit-mask associado. Então, isso nunca pode ser ignorado.

NMIs são frequentemente usados por timers, especialmente por watchdog timers.

### Inter-processor Interrupt (IPI)

Caso especial gerado por um processador para interromper outro processador em um sistema de multiprocessadores.

### Software Interrupt

É uma interrupção gerada dentro de um processador pela execução de uma instrução.

Interrupções de software são frequentemente usadas para implementar chamadas de sistema porque elas implementam uma chamada de sub-rotina com a mudança de nível da CPU.

## ATIVIDADE

---

1 - Pensando em dispositivos de entrada e saída, você conseguiria perceber dispositivos que hora podem servir como entrada, hora como saída ou que podem ter somente uma dessas funções? O que você pensa sobre isso?

## Resposta Correta

---

2 - Em função da interação entre o sistema operacional e os dispositivos de E/S, pode-se afirmar que:

- ☐ a) o driver de dispositivo é o programa que faz a comunicação direta entre a CPU e o dispositivo, por meio de envio e recebimento de comandos.
- ☐ b) tudo o que os sistemas operacionais veem é a interface da controladora, que pode ser muito diferente da interface para o dispositivo de E/S.
- ☐ c) um driver pode ser inserido no núcleo, por meio da execução do programa no espaço de usuário e, na sequência, do reinício do sistema operacional.
- ☐ d) o driver, para ser usado, deve ser colocado dentro do sistema operacional para que seja executado em modo usuário.
- ☐ e) a entrada e a saída podem ser realizadas de várias maneiras; na mais simples, um programa de usuário emite uma chamada diretamente ao driver apropriado para iniciar o processo de E/S.

## Justificativa

---

3 - (FCC/TJ-PE/2012) Em relação aos modelos de entrada e saída, considere:

I. Na entrada e saída mapeada, o programa vê os dispositivos periféricos como endereços de memória, mandando dados para eles como se estivesse escrevendo na memória.

II. No modo de transferência simples, o controlador de DMA devolve o controle de barramento à CPU a cada transferência de um byte (ou palavra); no modo de transferência por bloco, o controlador de DMA não devolve o controle do barramento à CPU até que toda a transferência tenha sido efetuada.

III. Na entrada e saída mapeada, se cada dispositivo tem seus registradores de controle em uma página diferente do espaço de endereçamento, o sistema operacional pode dar a um usuário o controle sobre dispositivos específicos, simplesmente incluindo as páginas desejadas em sua tabela de páginas.

IV. Os controladores de DMA que usam endereçamento de memória física para suas transferências requerem que o sistema operacional converta o endereço virtual do buffer de memória pretendido em um endereço físico e escreva esse endereço físico no registrador de endereço do DMA.

Está CORRETO o que se afirma em

- ☐ a) I, II e III, apenas.
- ☐ b) I, II e IV, apenas.
- ☐ c) I, III e IV, apenas.
- ☐ d) II, III e IV, apenas.
- ☐ e) I, II, III e IV.

## Justificativa

---

4 - (UFG/2017) A adoção de um mecanismo de E/S orientada à interrupção tem como desvantagem a ocorrência de uma interrupção para cada caractere, o que desperdiça uma certa quantidade de tempo de CPU. Uma solução, em geral, mais eficiente para realizar E/S é usar:

- ☐ a) o sistema de spooling
- ☐ b) a E/S programada
- ☐ c) o acesso direto à memória
- ☐ d) a técnica de polling
- ☐ e) o pipeline

## Justificativa

# Glossário