

SISTEMAS OPERACIONAIS II

Prof. Renato Jensen

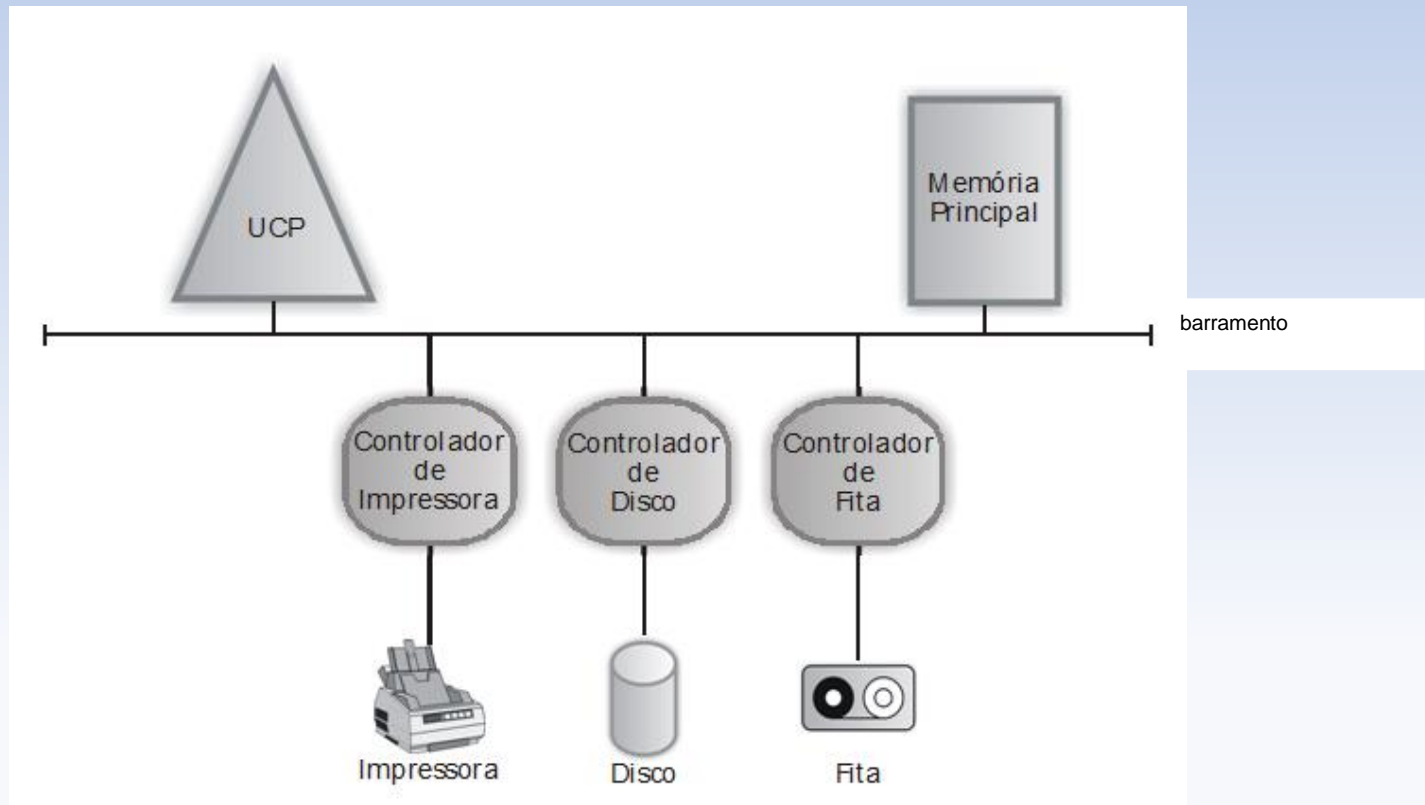
- **Sistemas de E/S**

- Uma das principais funções de um sistema operacional é controlar os dispositivos de E/S. Ele deve:
 - enviar comandos para os dispositivos.
 - capturar interrupções.
 - tratar erros.
 - oferecer uma interface entre os dispositivos e o restante do sistema.

- **Hardware de E/S**
- **Dispositivo de E/S**
 - Basicamente dividido em dois tipos: dispositivo de bloco e dispositivo de caractere.
 - **Dispositivo de bloco**
 - Armazena informações em blocos de tamanho fixo
 - Cada bloco tem seu endereço
 - Tamanho do bloco varia de 512 a 32.768 bytes
 - Lê e grava cada bloco independente dos outros
 - Exemplo: discos
 - **Dispositivo de caractere**
 - Entrega ou aceita um fluxo de caracteres sem estrutura de bloco
 - Não é endereçável
 - Não é possível realizar operação de busca
 - Exemplo: impressoras, interfaces de rede, mouse, etc.

- **Hardware de E/S**
- **Controladoras de dispositivos**
 - As **unidades de E/S** geralmente consistem de dois componentes:
 - Componente mecânico, que é o dispositivo em si
 - Componente eletrônico, que é a controladora
 - Cada controladora utiliza registradores para se comunicar com a CPU; por meio de informações nesses registradores o sistema operacional pode comandar o dispositivo
 - **Mapeamento de E/S por espaço em memória:**
 - Memória para dispositivos e memória para instruções e dados de programas estão no mesmo espaço de endereçamento
 - **Mapeamento de E/S por espaço de E/S:**
 - Memória para dispositivos e memória para instruções e dados de programas estão em espaços de endereçamento separados

- Hardware de E/S
- Controladoras de dispositivos

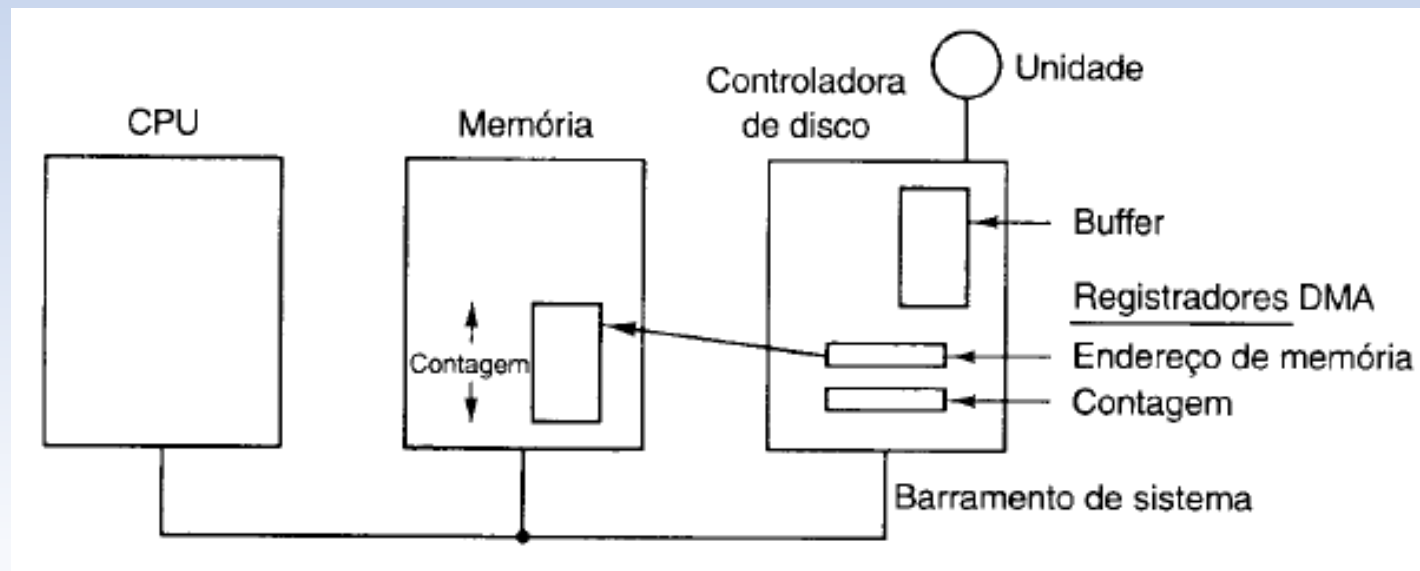


Modelo para conectar CPU, memória, controladoras e dispositivos de E/S

- **Hardware de E/S**
- **Execução de E/S**
 - **Programada:**
 - Computadores antigos onde todo o processo é executado pela CPU
 - O sistema operacional gerencia o acesso ao dispositivo
 - **Orientada a Interrupção:**
 - A CPU emite uma requisição de E/S e imediatamente pode continuar a executar instruções de outro processo
 - O dispositivo de E/S avisa a CPU quando a operação estiver concluída através de uma interrupção
 - O sistema operacional começa a executar, lê os dados do buffer da controladora e armazena-os na memória

- **Hardware de E/S**
- **Execução de E/S**
 - **Acesso Direto a Memória (DMA – *Direct Memory Access*):**
 - O controlador DMA deve existir no hardware para poder ser utilizado
 - Em uma leitura em disco, por exemplo, a CPU programa o controlador DMA passando informações como o endereço de memória para onde o bloco deve ir e o número de bytes a transferir
 - O controlador DMA aciona a controladora do disco que lê do disco para seu buffer interno
 - Ao concluir a leitura o DMA controla a escrita do buffer para a memória e gera uma interrupção
 - Quando o sistema operacional inicia, os dados lidos já estão na memória

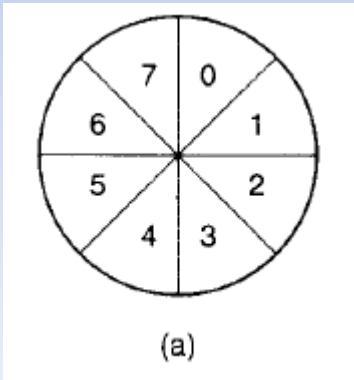
- Hardware de E/S
- Execução de E/S



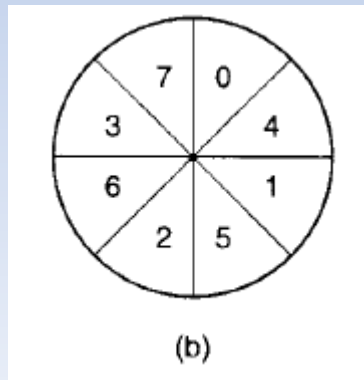
Acesso Direto a Memória – DMA

- **Hardware de E/S**
- **Execução de E/S**
 - **Leitura de Blocos por Intercalação (*Interleaving*):**
 - Enquanto os dados estão sendo transferidos da controladora para a memória, o próximo setor do disco estará passando sob o cabeçote de leitura, e este não poderá ser lido neste momento
 - A intercalação (interleaving) é a técnica de se formatar os blocos no disco com uma numeração que leva em conta um fator de intercalação
 - A ideia de numerar os blocos de modo intercalado é para permitir que blocos consecutivamente numerados sejam lidos com a taxa máxima do hardware
 - O fator de intercalação é definido pela controladora e dependerá do tempo de transferência de um bloco da controladora para a memória

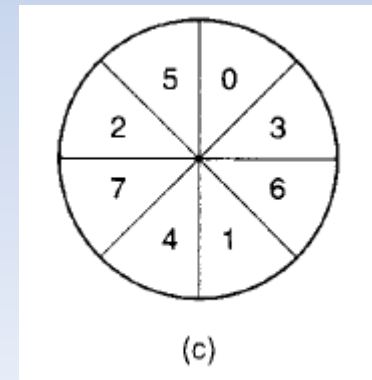
- Hardware de E/S
- Execução de E/S



Nenhuma intercalação



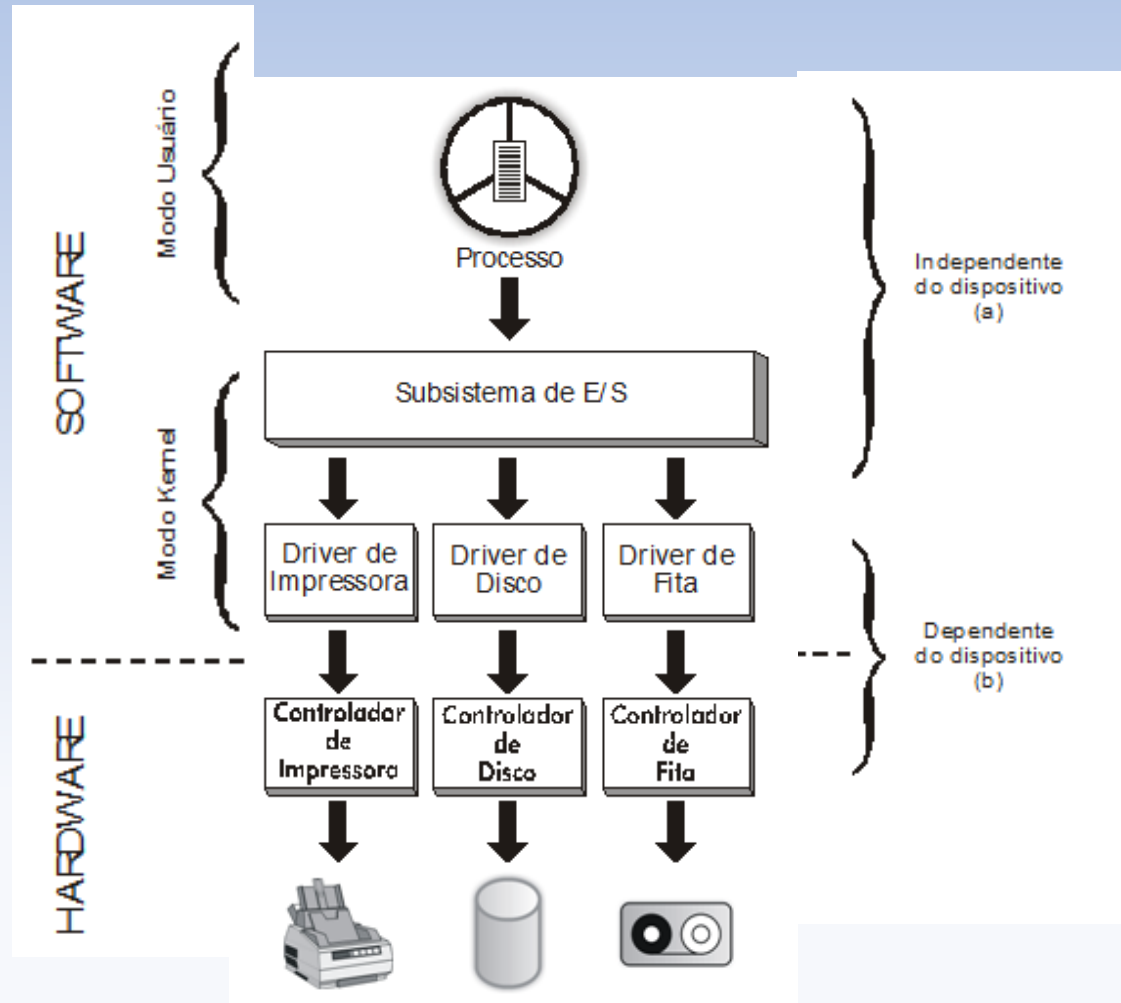
Intercalação única



Intercalação dupla

Leitura de Blocos por Intercalação

- Software de E/S



Estrutura de E/S

- **Software de E/S**

- ***Drivers* de Dispositivo**

- De modo geral seu trabalho é aceitar solicitações abstratas do software independente de dispositivo acima dele e cuidar para que a solicitação seja executada
- Possui as características:
 - Deve conter todo código dependente de dispositivo
 - Cada *driver* de dispositivo trata de um tipo de dispositivo ou no máximo de uma classe de dispositivos intimamente relacionados.
 - O *driver* de dispositivo é a única parte do sistema operacional que sabe quantos registradores uma controladora tem e para que eles são utilizados
 - Exemplo: *driver* de uma série de impressoras de um determinado fabricante

- **Software de E/S**
- ***Drivers* de Dispositivo**
 - Para a execução de uma solicitação de E/S o *driver* de dispositivo:
 - Determina que comandos serão enviados para a controladora e em que sequência
 - Emite os comandos gravando nos registradores da controladora
 - Bloqueia a si próprio até que as interrupções ocorram para desbloqueá-lo
 - Quando for acordado pela interrupção faz verificação de erros
 - Passa, se existir, os dados para o software independente de dispositivo

- **Software de E/S**
- **Software de E/S Independente de Dispositivo**
 - Sua função básica é executar as funções de E/S que são comuns a todos dispositivos e oferecer uma interface uniforme para o software de nível de usuário
 - Impede que o sistema operacional tenha que ser modificado para cada novo dispositivo
 - É responsável por mapear nomes simbólicos de dispositivo para o driver adequado. Por exemplo, no *Unix* um nome de dispositivo como */dev/disk0* aponta para um arquivo especial que contem a informação para localizar o *driver* apropriado
 - Normalmente também faz a proteção de acesso aos dispositivos aplicando as regras definidas no sistema operacional