

Sistema de Entrada e Saída

Prof. Edson Pedro Ferlin

Agradecimento ao Prof. Osmar Betazzi Dordal

- **Objetivos**
 - Analisar os sistemas de E/S sob a ótica do SO
- **Conteúdos**
 - Dispositivos de E/S
 - Modos de Operação
 - Drivers
 - Clock

Dispositivos de Entrada e Saída – E/S e *Clocks*

Dispositivos de Entrada e Saída

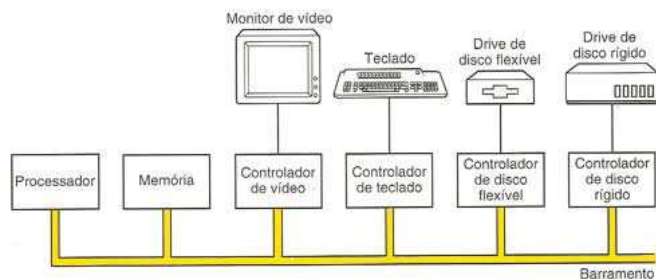
- Os Sistemas Operacionais também controlam os dispositivos de Entrada e Saída.
- Os Sistemas Operacionais devem emitir comandos para instruções como
 - *read*, *write* e etc.
- Ele também deve interceptar as interrupções e tratar erros destes dispositivos.
- O sistema de E/S deve fornecer uma interface única (API)
 - Para todos os dispositivos.
- Maioria dos Sistemas Operacionais corresponde aos *drivers*.

Tipos de Entrada e Saída – Classificação

- Tipo de conexão
 - Serial;
 - Paralelo.
- Tipo de transferência de dados
 - Dispositivos de blocos;
 - Dispositivos de caracteres.
- Tipo de compartilhamento de conexões
 - Ponto a Ponto;
 - Multiponto.

Hardware

- O Sistema Operacional trata os dispositivos de E/S por meio de controladoras.
- Cada controladora é particular para cada dispositivo de E/S.
- Comunicação entre CPU e Controladores
 - Usa barramentos comuns (interface de **alto nível**).
- Interface entre controladora e dispositivo (**baixo nível**).



O sistema Operacional não deve se preocupar com o dispositivo, os *drivers* devem tratar o erro o mais próximo possível do dispositivo. Uma requisição de **leitura** ou **escrita** deve ser transparente ao Sistema Operacional (uma leitura em disco ou pendrive ou em cd-rom deve ser idêntica).

Modos de Operações de Entrada e Saída

- Programada
- Interrupções
- Acesso Direto à Memória
 - *Direct Memory Access – DMA.*
- Qual a diferença destes 3 modos?
 - Participação da CPU;
 - Utilização das interrupções.

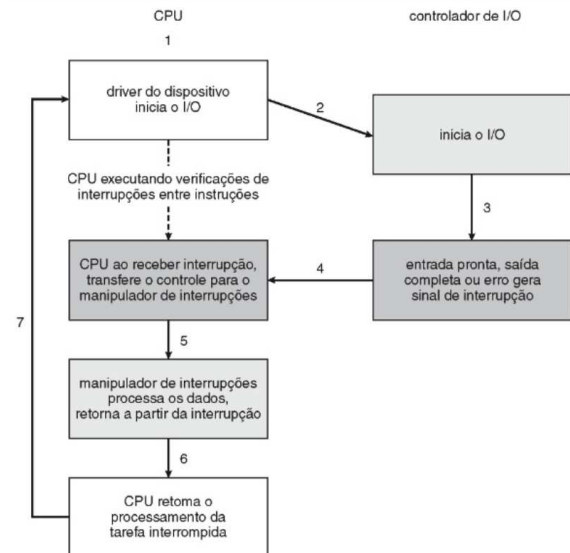
Entrada e Saída Programada

- Esta forma é a mais simples, no qual **tudo é realizado pela CPU.**
- Os dados são trocados entre a CPU e o módulo de E/S.
- A CPU executa um programa para:
 - Verificar o estado do módulo de E/S;
 - Enviar comandos de operação;
 - Aguardar o resultado;
 - Efetuar a transferência para os registradores da CPU.
- Porém, a CPU ficará ocupada todo o tempo
 - Isso é uma grande desvantagem.



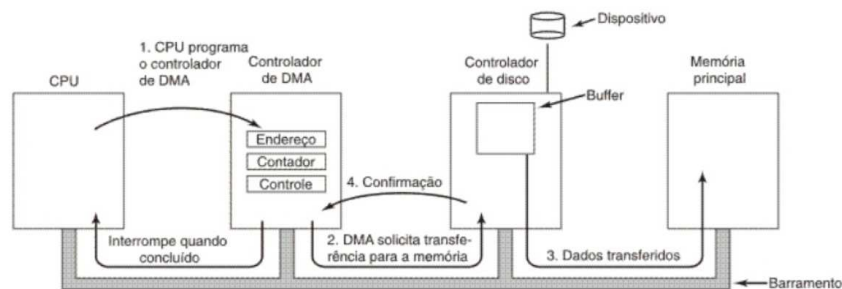
Entrada e Saída via Interrupção

- **Exemplo** do funcionamento da interrupção (Aulas de processos)
 - A CPU requisita uma leitura no disco;
 - A controladora lê os dados enquanto a CPU faz outras tarefas;
 - A controladora envia uma **interrupção** para a CPU;
 - A CPU solicita os dados;
 - A controladora envia os dados.
- A CPU fica mais livre, mas ainda participa do processo.



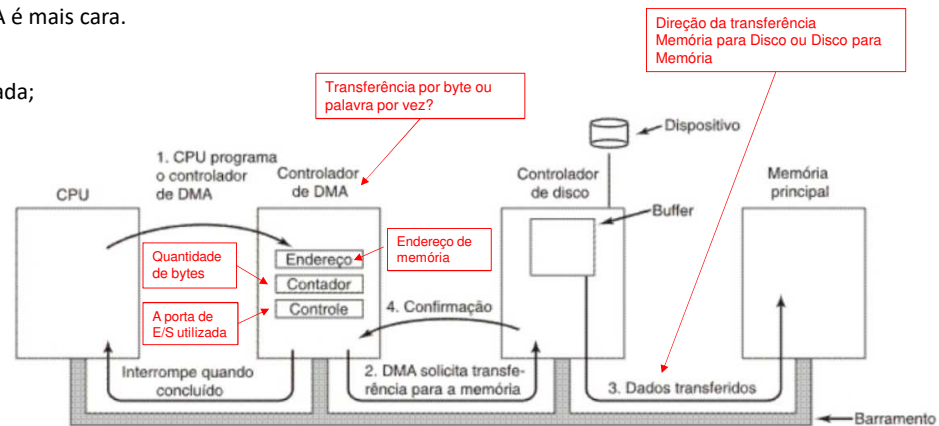
Entrada e Saída via Acesso Direto à Memória

- *Direct Memory Access - DMA*
- Este controlador retira a CPU da função de gerenciamento
- Necessita de uma controladora de DMA
- A DMA faz a E/S programada e vez da CPU



Entrada e Saída via Acesso Direto à Memória

- Desvantagem:
 - CPU é mais rápida que a controladora de DMA – ociosidade da CPU;
 - Arquitetura com DMA é mais cara.
- Vantagens
 - Executa E/S programada;
 - Faz todo o trabalho;
 - Libera a CPU.



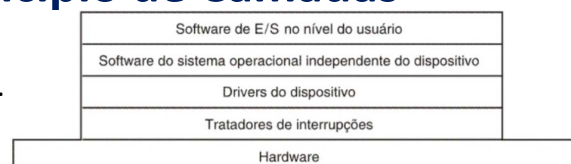
11

Sistemas de Entrada e Saída

Prof. Edson Pedro Ferlin

Entrada e Saída – Princípio de Camadas

- Facilita a **independência** dos dispositivos.
- Modularidade e coesão.
- Camadas mais baixas
 - Detalhes de *hardware*;
 - Drivers e manipuladores de interrupção.
- Camadas mais altas
 - Interface para usuário;
 - Aplicações de usuário;
 - Chamadas de sistemas; parte independente de E/S.



Independência o SO não necessita ser modificado para fazer uma leitura em disco ou *pendrive*

Windows comando "*dir*"
Linux comando "*ls*"

Windows "*explorer*"
MacOSX comando "*finder*"

Fazer *buffering*, ajustes de velocidade, cache de dados, escalonamento de E/S, reportar erros de programação, erro de leitura de E/S, erros de memória (escrita em endereços inválidos), entre outros

12

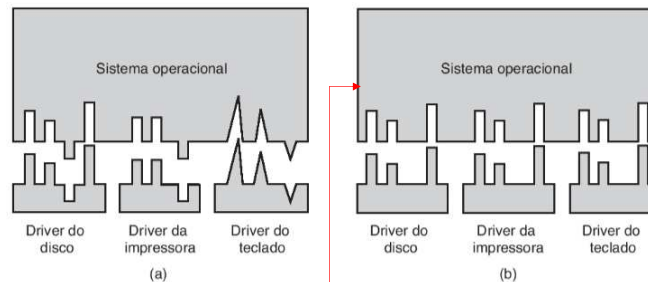
Sistemas de Entrada e Saída

Prof. Edson Pedro Ferlin

Entrada e Saída – Princípios de Software

- Fornece interface uniforme.
- Cada dispositivo fornece drivers com funções.
- Padronizar as funções fornecidas.
- O Sistema Operacional define que funções o driver deve fornecer
 - (a) sem padrão;
 - (b) com padrão.

A API de cada drive deve seguir um determinado conjunto de funções: `read()`; `write()`; `send()`; `receive()`...



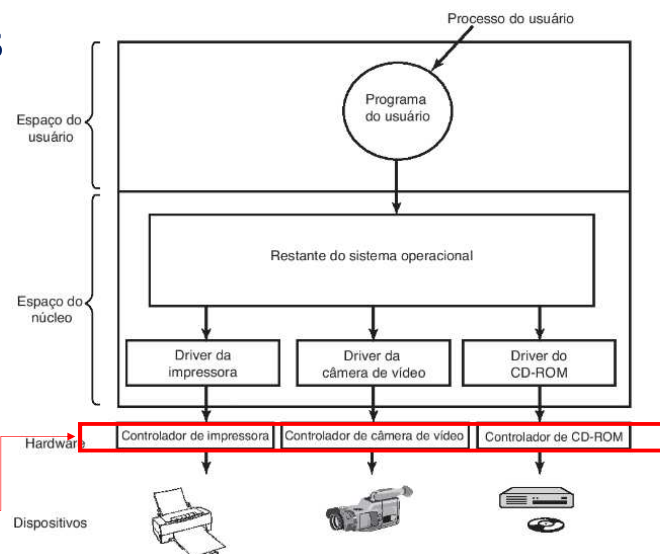
13

Sistemas de Entrada e Saída

Prof. Edson Pedro Ferlin

Entrada e Saída – Drivers

- Escrito pelo fabricante do dispositivo
- SOs diferentes, os *drivers* são diferentes
- Fazem parte do *kernel* do SO
- Acesso total a todo dispositivo
- Podem causar problemas no SO



Comunicação realizada pelo barramento

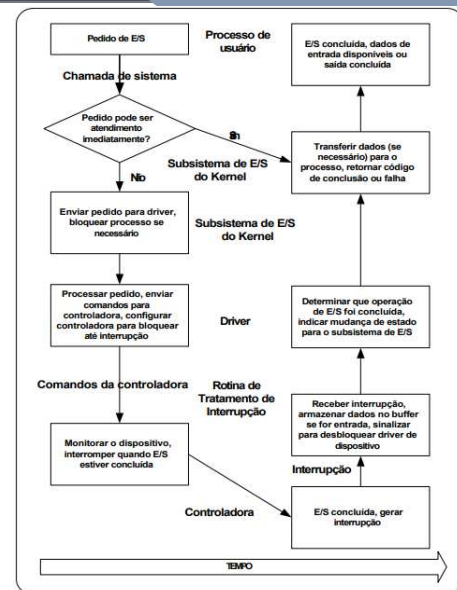
14

Sistemas de Entrada e Saída

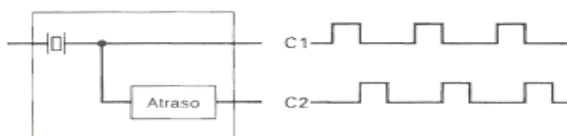
Prof. Edson Pedro Ferlin

E/S – Funções dos Drivers

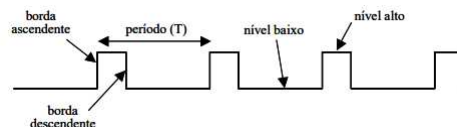
- Aceitar pedidos de leitura/escrita so software
 - Independente de dispositivos;
 - Pedido vale para qualquer dispositivo;
 - Cuidar que seja executadas.
- Fazer um *check-up* no início
 - Inicializar o dispositivo, caso seja necessário;
 - Gerenciar as necessidades de energia do dispositivo;
 - Criar um *log* de eventos.



Clock



$$f = \frac{1}{T}$$



Clocks (Timers)

- *Clock* de hardware
 - Dispositivo que gera pulsos síncronos (coração);
 - Localizado na CPU ou na placa-mãe;
 - Sinal utilizado para a execução de instruções;
 - Presente em qualquer sistema multiprogramado;
 - Frequência: número de vezes que o pulso se repete por segundo (Hz).
- *Clock* de Software
 - O hardware só gera pulsos em tempos;
 - Todo o resto depende do *driver* de *clock*;
 - Funções do *clock drive*:
 - Manter a hora do dia;
 - Contabilizar o tempo do processo;
 - Verificar o uso da CPU (*timeslice*);
 - Cuidar da chamada de sistema *alarm/settimer*;
 - Fazer monitoramento e estatísticas.

Contato



eferlin@live.com



(BLOG) professorferlin.blogspot.com

(SITE) professorferlin.webnode.com.br

(YOUTUBE) ProfEdsonPedroFerlin