

Sistemas operacionais

Aula 5 - Gerência de entrada e saída

INTRODUÇÃO

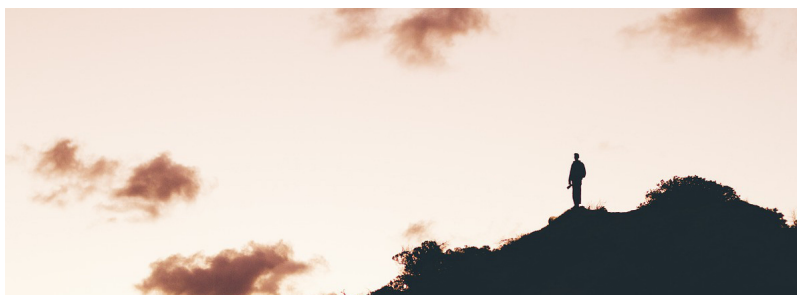


Uma das principais funções de um Sistema Operacional, sem sombra de dúvida, é gerenciar e controlar todos os dispositivos de entrada e saída disponíveis. Esta tarefa não é nada simples, posto que o S.O. deve, para realizar esta atividade, enviar comandos aos dispositivos, capturar e enviar interrupções, além de tratar possíveis erros.

O computador, de fato, se tornaria um equipamento inútil se não existisse alguma forma do meio externo se comunicar com ele. Os dispositivos de entrada e saída (E/S) são responsáveis por fazer essa comunicação entre o sistema, de modo geral, e o meio externo.

Nesta aula, você irá identificar os princípios de hardware e de software de E/S. Abordando ainda mecanismos de bufferização para tornar os programas independentes dos dispositivos, bem como dos spoolings que aceleram o trabalho de leitura e escrita de dados.

OBJETIVOS





Analisar o mecanismo de interrupção.

Identificar as técnicas de entrada e saída.

Diferenciar os princípios de software e de hardware.

GERÊNCIA DE ENTRADA E SAÍDA

Uma parte bastante importante dentro dos Sistemas Operacionais é a dos mecanismos de Entradas e Saídas, através do qual o sistema se comunica com o mundo externo. Este nível é composto por processos servidores de E/S (gerentes de dispositivos e drivers) que também são processos do S.O. e também competem pelos seus recursos.



Portanto, as instruções de E/S são sempre do tipo:

- Sai comando;
- Saem dados;
- Entram dados; e
- Entra estado do dispositivo.

Toda operação do tipo E/S pode ser subdividida no tempo em três etapas distintas:

Início da E/S



Transferência de dados



Finalização da operação

Técnicas empregadas em operações de Entrada e Saída

Buffering

Esta técnica armazena temporariamente, em uma região de memória física do computador, chamada buffer, os dados enquanto eles são movidos entre processos ou dispositivos.

São utilizados quando existe uma grande diferença de velocidade entre a origem e o destino.

Como exemplo, imagine que você esteja assistindo a um vídeo online (*video on demand* – VOD).

O VOD é muito sensível à variação do atraso, se um pacote demora muito para chegar o vídeo irá travar; para evitar isso, o player de vídeo “bufferiza” alguns segundos de vídeo para que possa compatibilizar a velocidade da rede com a da placa de vídeo.

Spooling

A técnica de *spooling* consiste em armazenar os dados em uma memória auxiliar. Ela permite que um dispositivo lento não provoque ociosidade na CPU.

Por exemplo, quando um programa deseja imprimir um documento antes de solicitar a operação de E/S ele salva o conteúdo em um arquivo no disco. Após isso, faz uma chamada de sistema que aciona o S.O. que lê o arquivo e transfere os dados para o buffer da impressora.

Arquivo
de spool

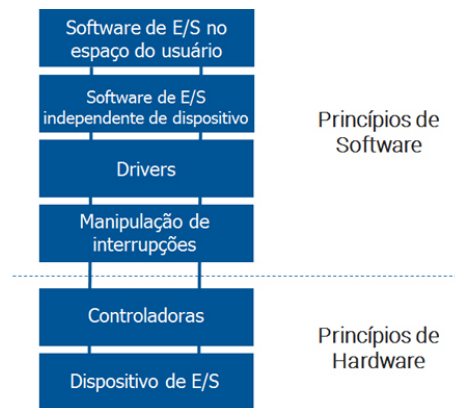
Impressão



A partir do que vimos sobre *buffering* e *spooling*, responda: Qual é a diferença entre essas duas técnicas?

Resposta Correta

O Gerenciamento de Entrada e Saída engloba o estudo dos princípios de hardware de software.



SOFTWARE DE E/S DE NÍVEL DE USUÁRIO



Fonte da Imagem: Vidoslava / Shutterstock

A visão dos dispositivos de E/S para o usuário consiste de bibliotecas vinculadas em programas de usuários. Essas bibliotecas são fornecidas pelas linguagens de programação e podem ser utilizadas pelos usuários e ligadas com seus programas para compor o arquivo executável.

As funções de entrada e saída são dependentes e específicas de [cada linguagem de programação \(glossário\)](#). As bibliotecas de entrada e saída não fazem parte do núcleo do Sistema Operacional. Elas são associadas às várias linguagens de programação.

SOFTWARE DE E/S INDEPENDENTE DE DISPOSITIVO

A principal função do software de E/S independente de dispositivo é executar funções que são comuns para vários dispositivos, e oferecer uma interface uniforme para o software de nível de usuário.

Alguns serviços oferecidos são:

Nomeação de Dispositivo

Cada dispositivo deve receber um nome lógico e ser identificado a partir dele;

Buferização

Buffer é uma zona de memória temporária utilizada para armazenar dados enquanto eles estão sendo transferidos entre as diferentes camadas do software de E/S;

Cache de Dados

Armazenar, na memória, um conjunto de dados que estão sendo frequentemente utilizados.

Alocação e Liberação

Devido a alguns dispositivos admitirem, no máximo, um usuário por vez, o software de E/S deve gerenciar a alocação, a liberação e o uso destes dispositivos;

Tratamento de Erros

O software de E/S deve fornecer mecanismos de manipulação de erros, informando à camada superior o sucesso ou o fracasso de uma operação.

DRIVERS DE DISPOSITIVOS



Fonte da Imagem:

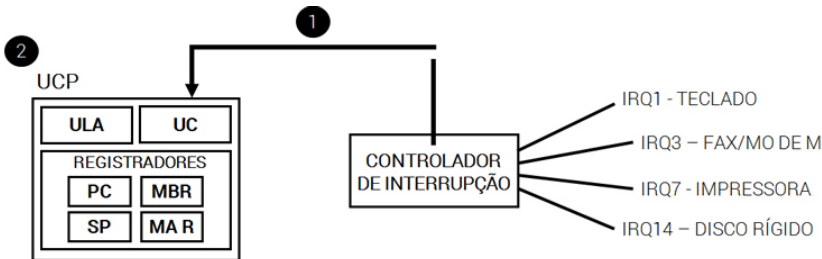
O driver de dispositivo (*device driver*) é composto de um conjunto de módulos de software cada um implementado para fornecer mecanismos de acesso a um dispositivo de entrada e saída específico. Assim, cada driver de dispositivo trata de um tipo de dispositivo ou de uma classe de dispositivos correlacionados.

De modo geral, um driver é responsável por aceitar uma solicitação abstrata (por exemplo, ler alguma informação) e cuidar para que esta solicitação seja atendida. Assim o driver de um dispositivo tem o conhecimento de como funciona a controladora.

INTERRUPÇÃO

É um sinal informando que um evento ocorreu. Vários eventos podem gerar uma interrupção como, por exemplo, o fim de uma operação de E/S, erros de programa como divisão por zero ou acesso inválido à memória entre outros.

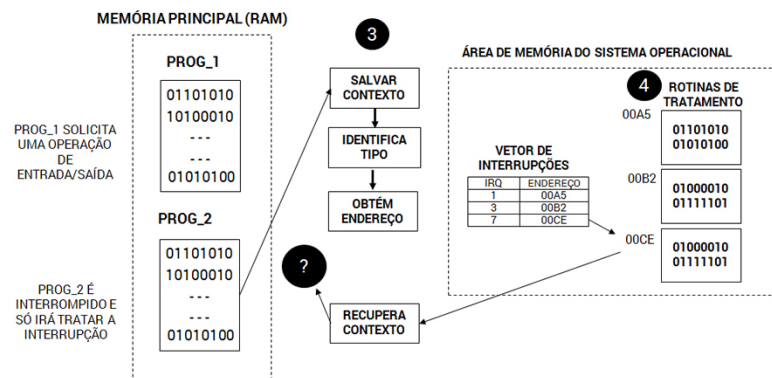
Cada interrupção é identificada com um número. Veja a ilustração a seguir:



Controle de Interrupção

Quando ocorre uma interrupção o programa em execução passa ao estado de disposição e é iniciada a rotina de tratamento de interrupção.

Para cada interrupção, uma rotina de serviço é designada para tratar a interrupção.



Com base nas imagens, podemos resumir o funcionamento das interrupções da seguinte forma:

No decorrer da execução das instruções de um programa é solicitada uma operação de E/S, por exemplo, um determinado trecho deve ser copiado, do disco para memória principal, para que o fluxo de execução deste programa continue;

Como a velocidade de acesso aos dispositivos de E/S é menor que a velocidade de processamento das instruções por parte do processador, o processador desvia o fluxo de execução de instruções para outro programa e o solicitado passa a bloqueado;

O pedido de interrupção, normalmente, é gerado através da ativação de um sinal conectado ao processador (1). Ao ocorrer o pedido de interrupção, o processador interromperá a execução do programa, salvará o valor atual dos seus registradores (contexto) (3), identificará a origem do pedido de interrupção e em seguida, executará a rotina associada ao tratamento da situação que levou à ocorrência do pedido de interrupção (4);

Essas rotinas de tratamento são parte do S.O. e encontram-se carregadas na memória principal. O endereço de cada rotina é obtido em uma tabela, chamada vetor de interrupções.

A animação, a seguir, resume todo o processo:

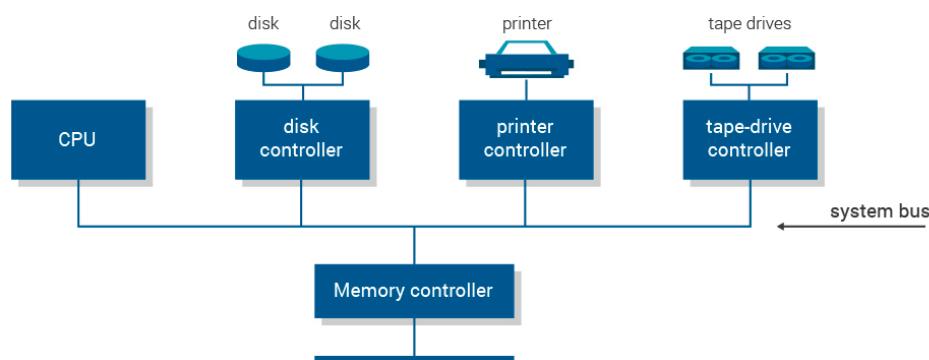
Princípios de hardware e de software de E/S

PRINCÍPIOS DE HARDWARE

Um sistema de computação de uso geral moderno consiste de uma CPU e em uma série de controladoras de dispositivos conectadas por um barramento comum que fornece acesso à memória compartilhada.

Cada controladora de dispositivo está encarregada de um tipo específico de dispositivo (por exemplo, unidades de disco, dispositivos de áudio e monitores de vídeo). A CPU e as controladoras de dispositivo podem executar de modo concorrente, competindo pelos ciclos de memória.

Observe, na imagem, um sistema moderno de computação.



DISPOSITIVOS DE E/S

É o mecanismo utilizado como interface entre o mundo exterior e o computador.

Existe uma enorme diversidade de dispositivos que são classificados quanto a fluxo de dados da seguinte forma:

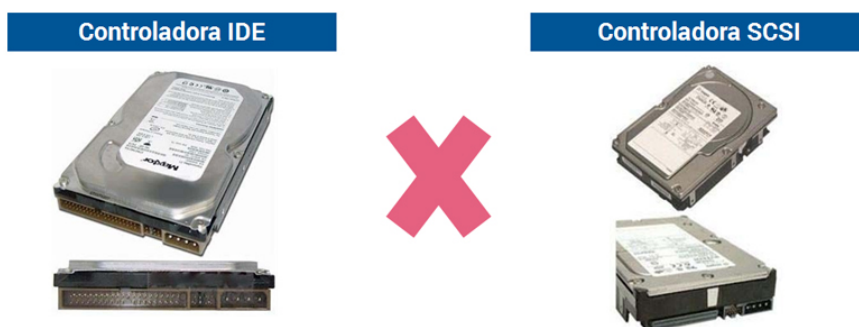


CONTROLADORAS DE DISPOSITIVOS

Os dispositivos são ligados ao computador através de um componente de hardware chamado de interface. Devido à diversidade de tipos de dispositivos, que abstrai diferentes formas de operações e complexidade, as interfaces empregam, no seu projeto, outro componente de hardware conhecido como controlador de dispositivo.

A controladora de dispositivo (chamada também de adaptador de dispositivo) trata-se de um componente eletrônico, comumente na forma de uma placa de circuito impresso, que pode ser inserida na placa mãe do computador.

O dispositivo em si trata-se de um componente mecânico. Uma controladora pode manipular mais de um dispositivo e, quando padronizadas, podem ser fabricadas por diversas empresas. Como exemplo, temos as controladoras de disco IDE ou SCSI.



A distinção entre dispositivo e controladora deve ser ressaltada, já que o sistema operacional vê a controladora, não o dispositivo.

Cada controladora possui registradores que são utilizados para comunicação com a CPU. A operação de E/S é realizada quando o S.O. escreve os comandos nestes registradores. Quando o comando é aceito a CPU pode atender a outra tarefa, pois, a partir deste ponto, a controladora executa o comando dado e efetiva a operação solicitada.

Ao final da operação a controladora provoca uma interrupção visando permitir que o S.O. assuma a CPU e teste o resultado da operação. A CPU obtém o resultado e o *status* do dispositivo pela leitura de um ou mais bytes de informação nos registradores da controladora.

ACESSO DIRETO À MEMÓRIA (DMA)

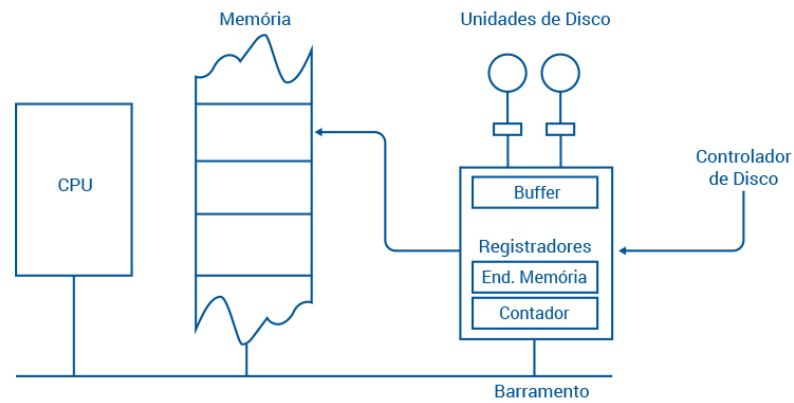
Muitas controladoras, como, por exemplo, as dos HDs modernos suportam DMA.

Uma controladora comum (sem acesso direto à memória) armazena os dados em seu buffer e verifica possíveis erros. A seguir, gera uma interrupção. O Sistema Operacional passa a executar e realiza a leitura do buffer da controladora, cada byte por vez, e armazena-os na memória principal. Esta operação, naturalmente, exige desperdício de CPU.

O DMA foi projetado para liberar a CPU do trabalho de cópia dos dados da controladora para a memória. O controlador DMA é conectado ao barramento de dados e de endereços do computador, para ter a capacidade de acessar diretamente endereços de memória. Desta forma, após os dados terem sido lidos do dispositivo, a controladora se encarrega de copiá-los para a memória, sem interferência da CPU. Após terminar a transferência dos dados para a memória principal, a controladora DMA gera uma interrupção para o Sistema Operacional, que ao ser iniciado já

encontra o dado em memória principal.

A transferência via DMA é processada sem intervenção da CPU, conforme nos mostra o exemplo abaixo:



Glossário

CADA LINGUAGEM DE PROGRAMAÇÃO

Por exemplo, na linguagem C as funções `printf` e `scanf` são utilizadas para impressão formatada e leitura, respectivamente.