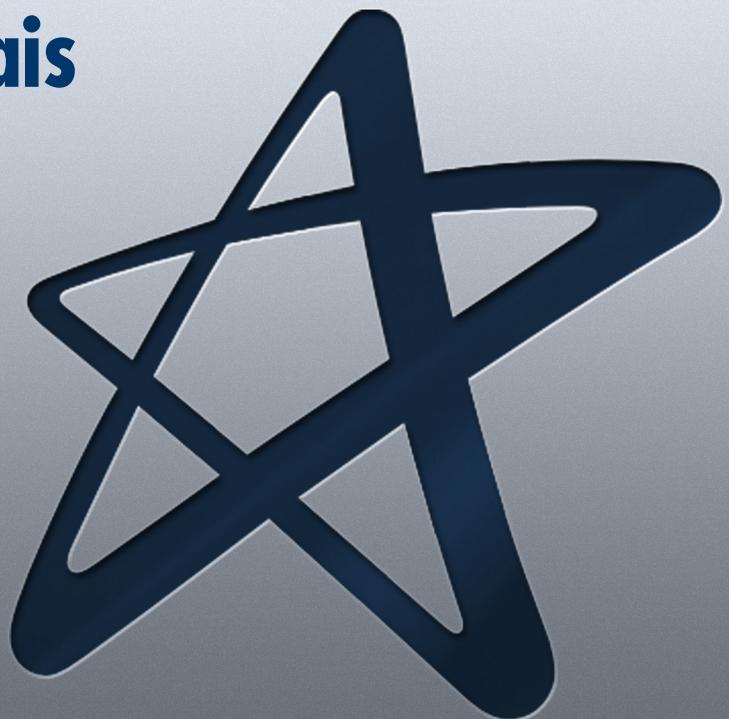


# Sistemas operacionais



Educação a Distância  
Cruzeiro do Sul Educacional  
Campus Virtual



# Material Teórico



Gerenciador de Entrada e Saída

**Responsável pelo Conteúdo:**

Prof. Ms. Claudney Sanches Junior

**Revisão Textual:**

Profa. Ms. Magnólia Gonçalves Mangolini



# UNIDADE

## Gerenciador de Entrada e Saída



- Introdução
- Gerenciamento de Entrada e Saída
- Rotina de E/S
- Controladores
- DMA - Direct Memory Access (Acesso Direto à Memória)
- Driver de Dispositivo ou Device Driver



- Nesta unidade você deverá estudar os conceitos do gerenciador de entrada e saída dos SO.
- A unidade apresenta a definição do gerenciador de entrada e saída e mostra o funcionamento do SO e os controladores. Detalha os device drivers e apresenta o conceito de DMA. Neste contexto, espera-se que ao final da unidade você seja capaz de entender o gerenciamento de E/S dos SOs modernos.

Para que você consiga atingir o objetivo desta unidade, sugiro o seguinte plano de estudos:

1. Leia o material teórico relativo a esta unidade.
2. Assista a apresentação narrada, onde serão expostos os principais conceitos de Gerenciamento de Entrada e Saída.
3. Realize as atividades de sistematização (AS\_V), que serão compostas de 6 questões de múltipla-escolha sobre os conceitos-chave tratados na unidade. Quando a solução das questões for liberada, verifique quais foram as suas respostas incorretas e procure novamente pelo conceito nos nossos materiais.
4. Participe ativamente da atividade de aprofundamento (AP\_V), que será realizada através da participação no fórum de discussão.
5. Consulte as indicações de referências complementares desta unidade e pesquise exclusão mútua relacionada com E/S.

## Contextualização



A empresa comprou uma nova impressora e gostaria de compartilhar com outros usuários. Na máquina virtual com o servidor Linux, monte um servidor de impressão com o Samba para garantir a segurança e autenticação.

## 1. Introdução



O gerenciamento de dispositivos de Entrada e Saída (E/S) é uma das principais funções de um SO e consiste em controlar o acesso a todos os dispositivos de E/S, tais como: teclado, vídeo, impressoras, disco, fita magnética, etc.

Nesta unidade você vai estudar como o SO organiza, controla e acessa os dispositivos de E/S. Também serão apresentadas as rotinas de E/S e você conhecerá como funciona os controladores e os device drivers presentes em um SO.

Para que possa entender os conceitos de gerenciamento de E/S no SO, esta unidade está organizada da seguinte forma:

- a seção 2 apresenta o conceito de gerenciador de E/S;
- a seção 3 detalha as rotinas de E/S;
- a seção 4 mostra Controladores;
- a seção 5 apresenta DMA - Direct Memory Access;
- a seção 6 mostra o conceito de device drivers.

Ao final do estudo e das atividades desta unidade, você deve ser capaz de:

- entender sobre o relacionamento entre hardware e software;
- conhecer os fundamentos básicos de equipamento (hardware) e sistemas operacionais.

Não deixe de utilizar os fóruns associados à unidade para apresentar e discutir qualquer dificuldade encontrada.

## 2. Gerenciamento de Entrada e Saída



Uma das principais funções do SO é controlar todos os dispositivos de E/S do computador. Ele deve enviar comandos para os dispositivos, tratar erros, atender as interrupções e fornecer uma interface simples e fácil de usar entre os dispositivos e o resto do sistema. De maneira geral, gerenciar E/S é uma parte significativa do código do SO.

O principal objetivo do Gerenciamento de E/S é facilitar ao usuário o acesso aos dispositivos, sem que ele tenha que se preocupar com detalhes de funcionamento do hardware dos diversos dispositivos. Essa tarefa é bastante complexa, devido ao grande número de dispositivos e as diferenças eletromecânicas entre eles. Além disso, muitos dispositivos como os discos podem ser utilizados por vários usuários simultaneamente e o SO é responsável pela integridade dos dados compartilhados.

O software de E/S pode ser estruturado em uma das camadas do SO, sendo que cada camada apresenta uma tarefa bem definida para executar. Veja, na figura 1, a estrutura de um sistema computacional em camadas.

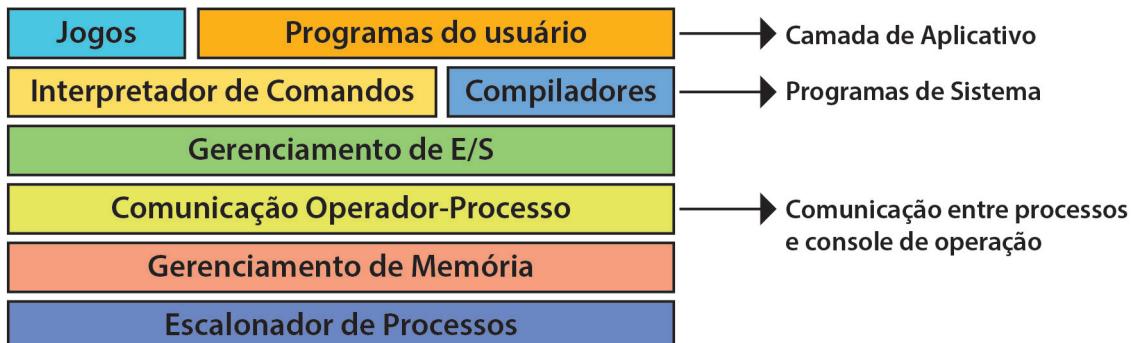


Figura 1. Apresentação das camadas de SO

Cada profissional da área computacional encara os dispositivos de E/S de uma forma diferente. Os engenheiros elétricos veem como chips, ligações elétricas, motores e componentes enquanto que os programadores veem os comandos, as interfaces, as funções e os erros que podem ser repassados ao software. No entanto a programação dos dispositivos de E/S muitas vezes está interligada com o hardware e como ele se relaciona com o programa.

De modo genérico, podem-se classificar os dispositivos de E/S como sendo: dispositivos de bloco e dispositivo de caractere. O dispositivo de bloco é aquele que armazena a informação em blocos de tamanho fixo, normalmente de 512 bytes a 32 768 bytes. A principal característica dos dispositivos de bloco é que cada bloco pode ser lido e escrito independentemente. Os discos são os dispositivos de bloco mais comum. O dispositivo de caractere envia e recebe um fluxo sem considerar quaisquer estruturas de bloco, ele não é endereçável e não dispõe de qualquer operação de posicionamento ou operação de acesso aleatório - seek operation. As impressoras, interfaces de redes, terminais e mouse são dispositivos de caractere.

Contudo este sistema de classificação não é perfeito. Os relógios ou timer que causam as interrupções não são classificados como dispositivos de bloco ou caractere. Os vídeos mapeados na memória também não se enquadram na classificação. Os sistemas de arquivos tratam os blocos como abstratos, mas, ainda assim, o modelo calcado em blocos ou caractere é amplamente utilizado para o desenvolvimento de softwares de E/S.

Os dispositivos de E/S apresentam uma ampla variação de velocidades, o que impõem a necessidade de aceitar diferentes ordens de magnitudes. Veja algumas variações na figura 2.

| Dispositivos     | Taxa de Dados |
|------------------|---------------|
| Teclado          | 10 bytes/s    |
| Mouse            | 100 bytes/s   |
| Modem            | 7 Kb/s        |
| Impressora Laser | 100 Kb/s      |
| Scanner          | 400 Kb/s      |
| Ethernet         | 1,25 Mb/s     |

|                         |           |
|-------------------------|-----------|
| USB                     | 1,5 Mb/s  |
| Câmera de Vídeo Digital | 4 Mb/s    |
| Disco IDE               | 5 Mb/s    |
| CD-ROM 40x              | 6 Mb/s    |
| Disco ATA-2             | 16,7 Mb/s |
| FireWire                | 50 Mb/s   |
| Disco SCSI Ultra-2      | 80 Mb/s   |
| Ethernet Gigabit        | 125 Mb/s  |
| Barramento PCI          | 528 Mb/s  |

Figura 2. Taxa de dados típicos

O SO fornece ao usuário uma interface de acesso aos dispositivos independente das características do hardware. Por exemplo, quando um usuário, ao desenvolver um programa em C, quer apresentar uma mensagem no monitor ele simplesmente utiliza o seguinte comando:

```
println("Oi");
```

Esse simples comando não é capaz de executar todos os procedimentos necessários para instruir o hardware do monitor a mostrar a mensagem. Na realidade, o compilador C traduz esse comando para uma chamada a uma rotina de E/S apropriada do SO, a qual envia os comandos para o driver do dispositivo que, por sua vez, instrui o controlador do dispositivo ou placa controladora. O controlador é quem envia os sinais ao monitor de vídeo para que este mostre a mensagem. Veja a figura 3 que apresenta o caminho que um comando escrito em um processo percorre até sua execução.

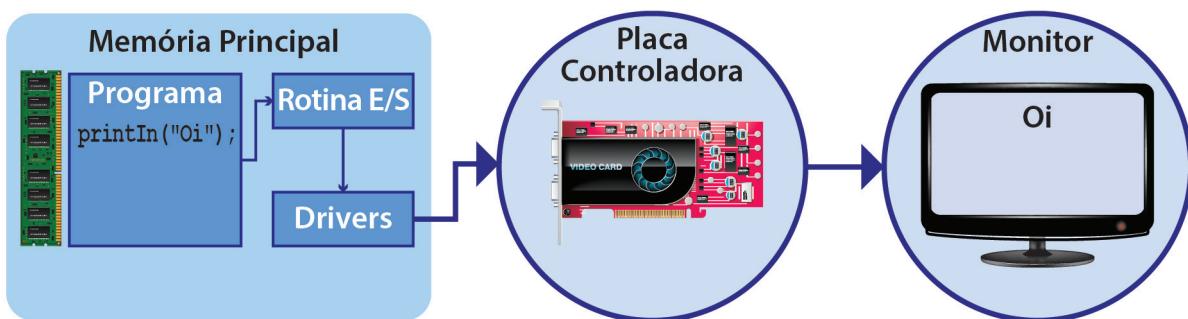


Figura 3. Relacionamento entre os componentes de software e hardware em uma operação de escrita no monitor de vídeo.

Outro exemplo seria ler um registro de um arquivo gravado em um disco. Para isso, o programador somente necessita fornecer o nome do arquivo e o registro que deseja ler. Entretanto, para que os dados sejam lidos do disco, são necessários vários comandos, por exemplo:

- dizer a localização física do registro no disco, ou seja, em qual disco, qual trilha, qual setor;
- enviar o comando para rotacionar o disco e posicionar a cabeça de leitura e gravação sobre a trilha;
- enviar o comando para ler os dados quando o setor passar pela cabeça;
- enviar os dados lidos para a localização de memória especificada;

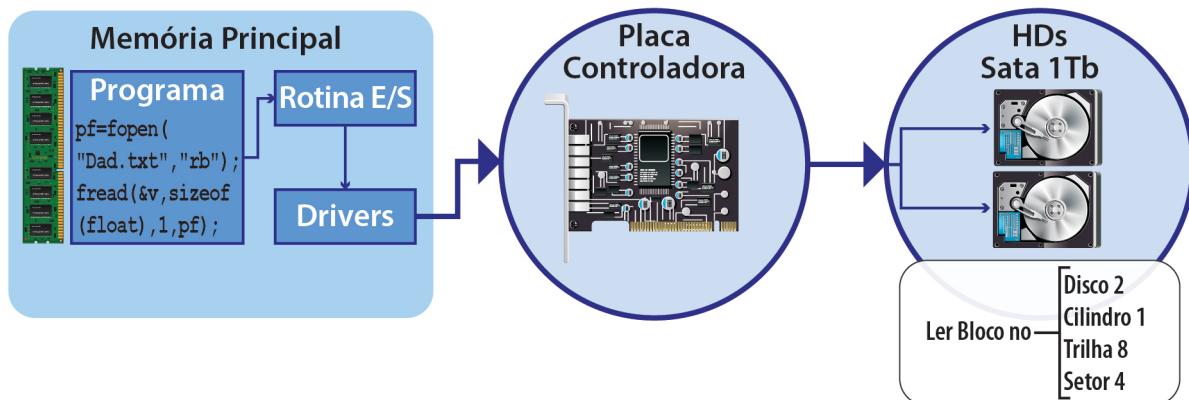


Figura 4. Relacionamento entre os componentes de software e hardware em uma operação de leitura do disco.

O sistema de gerenciamento de dispositivos é feito dividindo-se a tarefa de acesso aos dispositivos em várias camadas, cada qual, desempenhando uma função, conforme ilustrado na figura abaixo:

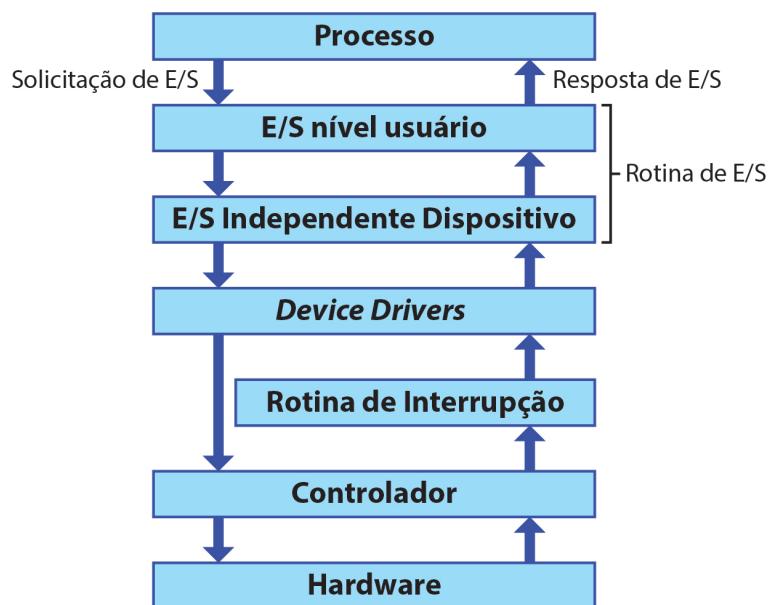


Figura 5. Componentes do sistema de gerenciamento de dispositivos de E/S.

### 3. Rotina de E/S



Apesar de a maior parte do software de E/S estar dentro do SO, uma pequena parte dele pode ser acessada pelo usuário desenvolvedor. Esta parte está na camada de E/S em nível de usuário, onde se encontram as bibliotecas que podem ser ligadas a programas de usuário. Por exemplo, os comandos printf, read, write, wireteln e o spooling.

Os objetivos da camada E/S independente do dispositivo são abrangentes, e pode ser citado como principais objetivos: criar uma interfaceamento uniforme para os device drivers, fornecer um mecanismo de nomeação do dispositivo, criar um tamanho de bloco independente do dispositivo, criar um espaço de bufferização para dispositivos de bloco e de caractere, cuidar da alocação de blocos livres em dispositivos de bloco, alocar e liberar dispositivos e manipular erros.

A rotina de interrupção ou manipuladores de interrupção devem ser escondidas do usuário ou transparente ao usuário. A forma de implementação mais comum dos SOs é a de quea rotina de solicitação de um processo, quando solicita uma E/S, o coloca no estado bloqueado e bloqueia o devicedirvers. Quando a interrupção de E/S ocorrer, a rotina de interrupção deve desbloquear o devicedrivers que, por sua vez, irá desbloquear o processo colocando-o na fila dos prontos.

## 4. Controladores



O dispositivo de E/S se conecta ao sistema através de um controlador, o qual corresponde à parte eletrônica do dispositivo responsável por enviar os comandos para o dispositivo externo (a parte eletromecânica).

As unidades de E/S são geralmente compostas de dois componentes principais: componentes eletrônicos e os componentes mecânicos. Os componentes eletrônicos são conhecidos como o controlador de dispositivos ou adaptador. Nos computadores pessoais é comum encontrá-los na forma de uma placa controladora ou placa de circuito impressa que pode ser inserida em algum conector de expansão ou slots do barramento. Por exemplo, existe a placa controladora de vídeo, a placa controladora de disco, a placa de rede, etc.

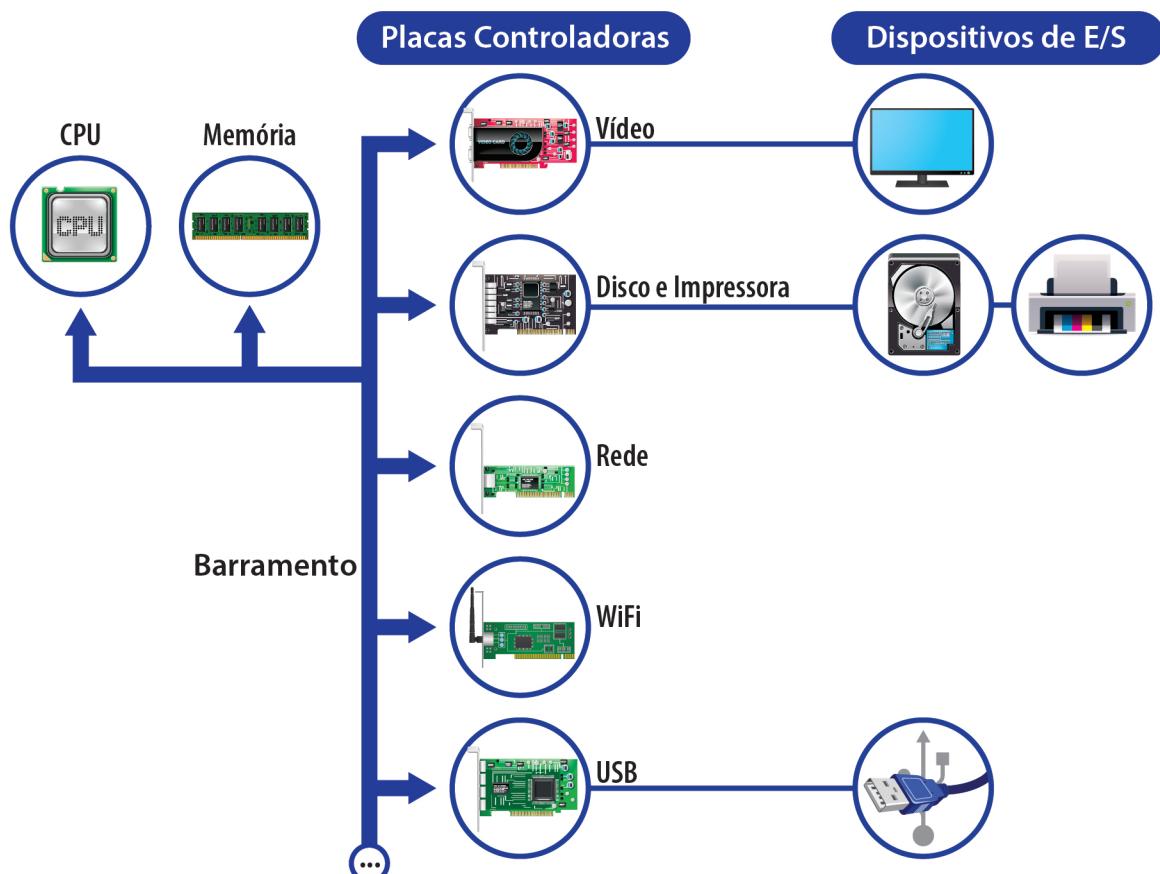


Figura 6. CPU, Memória, Barramento e Placas Controladoras

Muitos controladores foram criados para controlar mais de um dispositivo mecânico, e normalmente são padronizados (IEEE, ISO, ANSI etc.). Com o controlador, o SO não necessita baixar o nível ou enviar instruções ao dispositivo explicitando cada ação. Por exemplo, o controlador de vídeo controla o disposto monitor explicitando como o sinal será gerado, a intensidade, o feixe e o rendering. Enquanto o SO pode enviar ao controlador instruções como: desenhe algo, escreva, utilize a resolução x etc. Se não existisse o controlador, o SO deveria conhecer o dispositivo e enviar instruções detalhadas para a execução dos mesmos comandos.

Cada controlador apresenta registradores para a comunicação com a CPU. Por meio da escrita nestes registradores, o SO comanda o controlador e com a leitura de alguns registradores ele sabe o estado do dispositivo. Além disso, alguns controladores apresentam um buffer de dados, no qual o SO pode ler e escrever.

A CPU acessa os registradores e buffers do controlador, basicamente de duas maneiras. A primeira é associar o registrador de controle ao número da porta de E/S. Assim, a instrução em Assembly ficaria IN REG, PORT, ou seja, a CPU deve ler o registrador do controlador PORT e armazenar o resultado no registrador REG. Os primeiros computadores utilizavam esta solução. A segunda maneira é mapear todos os registradores no espaço de endereçamento da memória. Assim, cada registrador tem associado um endereço de memória único. Este sistema é chamado de E/S mapeada na memória. Em geral, os endereços associados estão no topo da memória principal. Um esquema híbrido, onde você encontrará as duas soluções juntas, foi elaborado, sendo o buffer associado a E/S mapeado na memória e portas de E/S associada aos registradores.

O SO não envia os comandos diretamente para o dispositivo de E/S e sim para o controlador do dispositivo. Este é quem comanda diretamente o dispositivo. Por exemplo, quando o SO identifica um comando para mostrar alguma mensagem no monitor de vídeo (writeln ou println), ele envia a mensagem para a placa controladora de vídeo, a qual enviará os sinais para comandar o monitor de vídeo para formar a mensagem.

Os controladores, em geral, possuem registradores internos que são utilizados para armazenar dados e comandos. O SO envia comandos para os controladores carregando-os nos registradores do controlador. Os parâmetros do comando também são armazenados em outros registradores.

Após o comando ter sido aceito pelo controlador, a CPU fica livre para realizar outra tarefa, enquanto o controlador executa o comando de E/S.

Por exemplo, quando um programa quer ler um bloco de dados do disco:

- o SO, através da CPU, envia os comandos de quais dados ler, onde colocar para o controlador, cabendo ao controlador comandar a leitura dos dados;
- enquanto a leitura está sendo feita, a CPU pode ser direcionada para executar outra tarefa;
- completada a leitura, o controlador gera uma interrupção para permitir que o SO ganhe o controle da CPU, verifique os resultados da operação e passe o controle para o programa que solicitou a leitura dos dados.

## 5. DMA - Direct Memory Access (Acesso Direto à Memória)



Os dados que entram no sistema através de um dispositivo de entrada devem ser armazenados em uma área de memória, quer seja uma variável, ou um buffer de dados, para que possam ser utilizados por um programa.

Nos primeiros sistemas de computadores, essa entrada de dados era feita em três etapas:

1. o controlador lê os dados do dispositivo e armazena-os em um buffer dentro do próprio controlador;
2. o controlador gera uma interrupção para avisar o SO de que os dados já estão disponíveis;
3. o SO é acionado, lê os dados do buffer do controlador e coloca-os em um buffer na memória principal.

Desse modo, na etapa 3 o SO utiliza a CPU para fazer a transferência dos dados do controlador para a memória, sendo que a CPU poderia estar executando outro processo se o próprio controlador se encarregasse de colocar os dados na memória.

A maioria dos controladores de hoje já conseguem fazer DMA. Nesta técnica, o próprio controlador já transfere os dados para a memória principal, liberando o SO desse trabalho e, consequentemente, liberando a CPU para realizar outro trabalho enquanto a transferência é efetuada. Por exemplo, imagine que um programa quer ler dados do disco:

- o SO envia um comando para o controlador de disco informando-o quais dados devem ser lidos e onde devem ser armazenados na memória principal;
- o controlador lê os dados do disco, colocando-os em seu buffer interno;
- após ter lido os dados do disco, o controlador transfere os dados para a localização da memória principal indicada pelo SO;
- após a transferência, o controlador gera uma interrupção para avisar ao SO que os dados já se encontram na memória.

## 6. Driver de Dispositivo ou Device Driver



Um devicedriver é a parte do SO que é dependente do hardware. Ou seja, seu código é específico para manipular um dispositivo de E/S. A função de um devicedriver é receber os comandos da Rotina de E/S, reconhecê-los e enviar os comandos correspondentes para o controlador para que este possa comandar o dispositivo de E/S. O devicedrivers, que também podem ser descritos como direcionadores de dispositivos ou ser conhecido popularmente apenas por driver, tem como objetivo geral aceitar os pedidos abstratos de serviços independentemente do dispositivo e ver como este pode ser executado. Normalmente, um sistema possui diversos drivers, tais como drivers para discos - disk drivers, fitas magnéticas - tape drivers, terminais - terminal drivers, impressoras, etc.

Para entender melhor o funcionamento de um driver, considere o que acontece quando um processador de textos, por exemplo o Word, manda um documento ser impresso em negrito:

- cada impressora tem seu conjunto de comandos específicos, que são diferentes entre impressoras. Um desses comandos será o de impressão em negrito;
- mesmo que o processador de texto conheça o conjunto de comandos de várias impressoras, quando uma nova impressora é lançada, o processador não saberá manipulá-la;
- para resolver esse problema, o processador de texto não envia os comandos diretamente para a impressora. Ele envia um comando abstrato (por exemplo, o comando negrito) para o driver da impressora;
- o driver da impressora converte este comando abstrato para o comando correspondente da impressora. Esse comando é passado para o controlador da impressora e posteriormente para a impressora.

Como visto, existe um conjunto de comandos abstratos que o driver utiliza, independentes de dispositivo que o programa utiliza e um conjunto de comandos correspondentes, comandos dependentes de dispositivo.

No caso de driver de impressoras, quando um fabricante lança uma nova impressora no mercado, ele também disponibiliza o driver para utilização da mesma. Portanto, a interface entre os comandos abstratos e o driver deve ser bem definida. Ou seja, o driver deve conhecer todos os comandos abstratos para saber identificá-los e enviar os comandos correspondentes para o controlador do dispositivo.

O drivers atende as requisições da seguinte forma: se estiver ocioso no momento de uma requisição ou pedido, ele o atende de imediato, mas se estiver ocupado, ele cria uma fila de pedidos pendentes. Para atender ao pedido de E/S, o drivers deve traduzir os termos abstratos para uma forma mais concreta, ou seja, deve definir quais operações devem ser programadas no controlador. Deve ainda escrever os comandos dos registradores do controlador, se bloqueando até ser acordado por uma interrupção. Após a operação ser concluída, o device drivers deve verificar a presença de erros e despertar o processo solicitador passando os dados solicitados bem como o código do status da operação. Por fim, irá verificar na fila de pedidos se existe algum pendente, e caso não encontre nenhum irá se bloquear e ficar assim até um novo pedido.

## Material Complementar

Com as Máquinas Virtuais (VM) que você instalou na Unidade I, inicie no SO Linux. O SO Linux apresenta um dos melhores gerenciador de E/S, assim, monte um gerenciador de impressão e de arquivos. Instale e configure o Samba para autenticar máquinas Linux e Windows, mantendo o Linux como servidor central. Monte outras máquinas virtuais - a clientes que deverão autenticar no servidor e, dependendo do privilégio do usuário, libere acesso à Impressão e a algumas pastas.

- **Gerenciamento de Arquivos no Linux**

Artur de Paula Coutinho, Curso de Linux Básico, [http://www.oocities.org/br/arturpc/curso\\_linux.pdf](http://www.oocities.org/br/arturpc/curso_linux.pdf)

- **Como instalar um servidor de impressão**

<http://www.tp-link.com.br/article/?id=352>

- **Como instalar o Samba**

[http://www.oficinadanet.com.br/artigo/450/configurando\\_o\\_samba\\_no\\_ubuntu](http://www.oficinadanet.com.br/artigo/450/configurando_o_samba_no_ubuntu)

<http://www.hardware.com.br/artigos/servidor-rede-local/>

## Referências

DEITEL, H.M. **Sistemas Operacionais**. 3. ed. São Paulo: Pearson Prentice Hall, 2005.

TANENBAUM, A.S. **Sistemas Operacionais Modernos**. 3. ed. São Paulo: Pearson Prentice Hall, 2009.

## Anotações



---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---



**Educação a Distância**  
Cruzeiro do Sul Educacional  
*Campus Virtual*

www.cruzeirodosulvirtual.com.br  
Campus Liberdade  
Rua Galvão Bueno, 868  
CEP 01506-000  
São Paulo SP Brasil  
Tel: (55 11) 3385-3000





Universidade  
**Cruzeiro do Sul**



**UNICID**  
Universidade  
Cidade de S. Paulo



**UNIFRAN**  
Universidade  
de Franca



**UDF**  
Centro  
Universitário



**Módulo**  
Centro  
Universitário