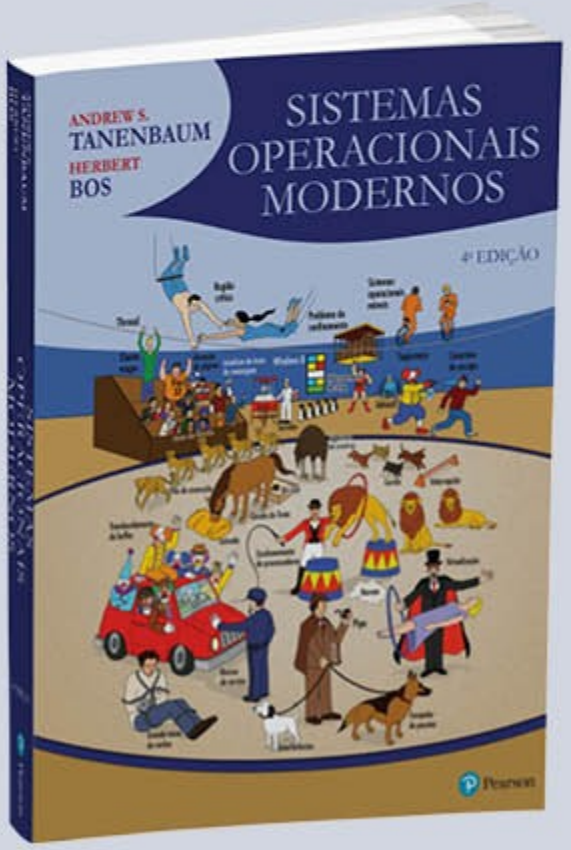
Capítulo 5:
Entrada/Saída



Princípios do hardware de E/S

- A entrada/saída é um tópico importante, mas muitas vezes negligenciado. Uma fração substancial de qualquer sistema operacional diz respeito à E/S. A E/S pode ser conseguida de três maneiras.
  - **1º:** há a E/S programada, na qual a CPU principal envia ou recebe cada byte ou palavra e aguarda em um laço estreito esperando até que possa receber ou enviar o próximo byte ou palavra.
  - **2º:** há a E/S orientada à interrupção, na qual a CPU inicia uma transferência de E/S para um caractere ou palavra e vai fazer outra coisa até a interrupção chegar sinalizando a conclusão da E/S.
  - **3º:** há o DMA, no qual um chip separado gerencia a transferência completa de um bloco de dados, gerando uma interrupção somente quando o bloco inteiro foi transferido.

Princípios do software de E/S

- Um conceito fundamental no projeto de software de E/S é conhecido como **independência de dispositivo.**
- O que isso significa é que devemos ser capazes de escrever programas que podem acessar qualquer dispositivo de E/S sem ter de especificá-lo antecipadamente.
- Por exemplo, um programa que lê um arquivo como entrada deve ser capaz de ler um arquivo em um disco rígido, um DVD ou em um pen-drive sem ter de ser modificado para cada dispositivo diferente.

Princípios do software de E/S

- Um objetivo muito relacionado com a independência do dispositivo é a **nomeação uniforme.** O nome de um arquivo ou um dispositivo deve simplesmente ser uma cadeia de caracteres ou um número inteiro e não depender do dispositivo de maneira alguma.
- **EXEMPLO**: Um pen-drive pode ser montado em cima do diretório */usr/ast/backup* de maneira que, ao copiar um arquivo para */usr/ast/backup/Monday*, você copia o arquivo para o pen-drive. Assim, todos os arquivos e dispositivos são endereçados da mesma maneira: por um nome de caminho.

Princípios do software de E/S

- Outra questão importante para o software de E/S é o **tratamento de erros**. Em geral, erros devem ser tratados o mais próximo possível do hardware.
- Se o controlador descobre um erro de leitura, ele deve tentar corrigi-lo se puder. Se ele não puder, então o driver do dispositivo deverá lidar com ele, talvez simplesmente tentando ler o bloco novamente.
- Muitos erros são transitórios, como erros de leitura causados por grãos de poeira no cabeçote de leitura, e muitas vezes desaparecerão se a operação for repetida. Apenas se as camadas mais baixas não forem capazes de lidar com o problema as camadas superiores devem ser informadas a respeito.
- Em muitos casos, a recuperação de erros pode ser feita de modo transparente em um nível baixo sem que os níveis superiores sequer tomem conhecimento do erro.

Princípios do software de E/S

- Ainda outra questão importante é a das transferências **síncronas** (bloqueantes) *versus* **assíncronas** (orientadas à interrupção). A maioria das E/S físicas são assíncronas — a CPU inicializa a transferência e vai fazer outra coisa até a chegada da interrupção. Programas do usuário são muito mais fáceis de escrever se as operações de E/S forem bloqueantes — após uma chamada de sistema read, o programa é automaticamente suspenso até que os dados estejam disponíveis no buffer.
- Fica a cargo do sistema operacional fazer operações que são realmente orientadas à interrupção parecerem bloqueantes para os programas do usuário. No entanto, algumas aplicações de muito alto desempenho precisam controlar todos os detalhes da E/S, então alguns sistemas operacionais disponibilizam a E/S assíncrona para si.

Princípios do software de E/S

- Outra questão para o software de E/S é a **utilização de buffer**. Muitas vezes, dados provenientes de um dispositivo não podem ser armazenados diretamente em seu destino final. Por exemplo, quando um pacote chega da rede, o sistema operacional não sabe onde armazená-lo definitivamente até que o tenha colocado em algum lugar para examiná-lo.
- Também, alguns dispositivos têm severas restrições de tempo real (por exemplo, dispositivos de áudio digitais), portanto os dados devem ser colocados antecipadamente em um buffer de saída para separar a taxa na qual o buffer é preenchido da taxa na qual ele é esvaziado, a fim de evitar seu completo esvaziamento. A utilização do buffer envolve consideráveis operações de cópia e muitas vezes tem um impacto importante sobre o desempenho de E/S.

Princípios do software de E/S

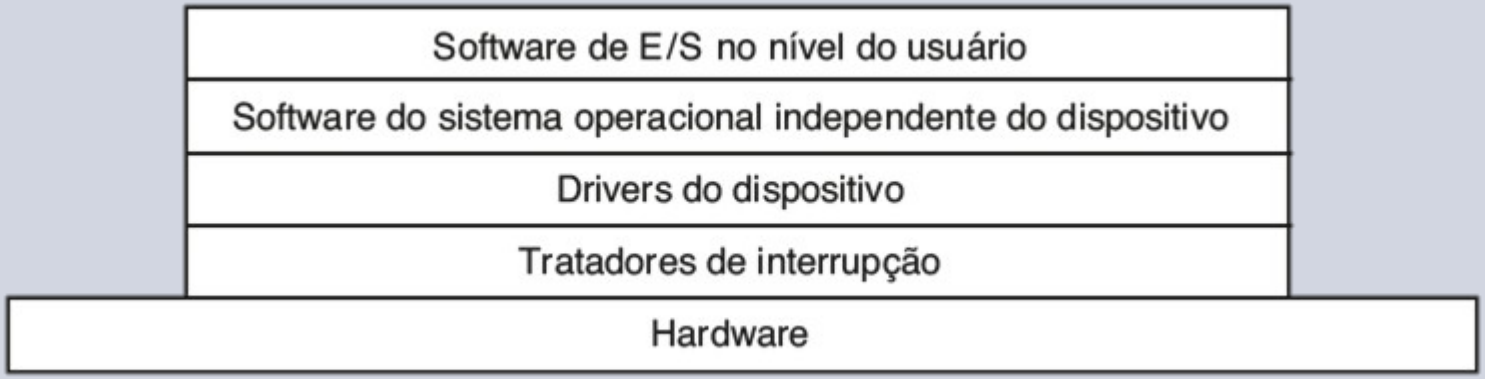
- O conceito final que mencionaremos aqui é o de dispositivos **compartilhados** *versus* **dedicados**. Alguns dispositivos de E/S, como discos, podem ser usados por muitos usuários ao mesmo tempo. Nenhum problema é causado por múltiplos usuários terem arquivos abertos no mesmo disco ao mesmo tempo.
- Outros dispositivos, como impressoras, têm de ser dedicados a um único usuário até ele ter concluído sua operação. Então outro usuário pode ter a impressora. Ter dois ou mais usuários escrevendo caracteres de maneira aleatória e intercalada na mesma página definitivamente não funcionará. Introduzir dispositivos dedicados (não compartilhados) também introduz uma série de problemas, como os impasses. Novamente, o sistema operacional deve ser capaz de lidar com ambos os dispositivos — compartilhados e dedicados — de uma maneira que evite problemas.

Camadas do software de E/S

- A E/S pode ser estruturada em quatro níveis: as rotinas de tratamento de interrupção, os drivers de dispositivos, o software de E/S independente do dispositivo, e as bibliotecas de E/S e spoolers que executam no espaço do usuário.
- Os drivers do dispositivo lidam com os detalhes da execução dos dispositivos e com o fornecimento de interfaces uniformes para o resto do sistema operacional.
- O software de E/S independente do dispositivo realiza atividades como o armazenamento em buffers e relatórios de erros.

Camadas do software de E/S

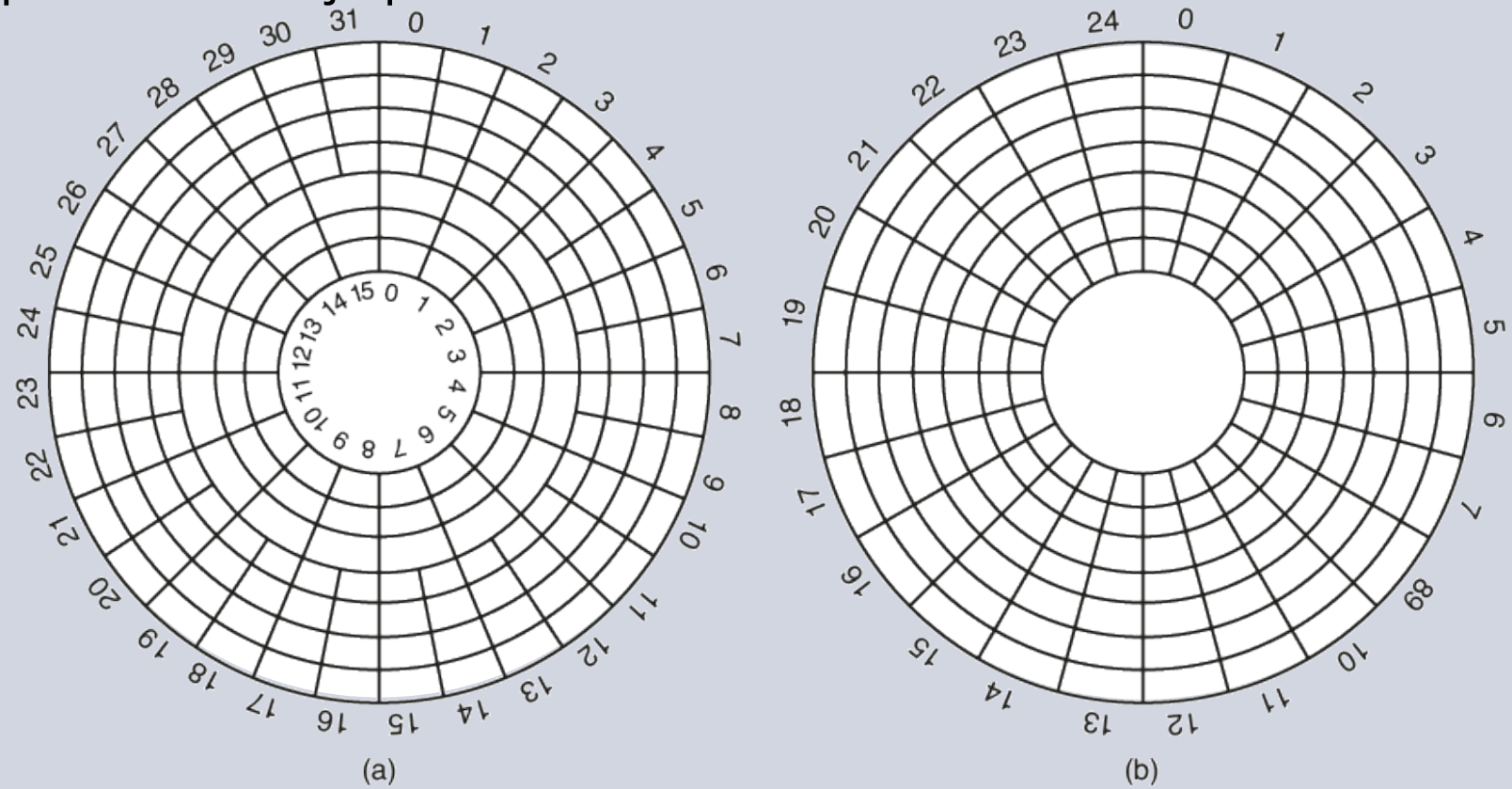
- Cada camada tem uma função bem definida a desempenhar e uma interface bem definida para as camadas adjacentes. A funcionalidade e as interfaces diferem de sistema para sistema; portanto, a discussão que se segue, que examina todas as camadas começando de baixo, não é específica para uma máquina.



Discos

- Os discos vêm em uma série de tipos, incluindo discos magnéticos, RAIDS, pen-drives e discos ópticos.
- Nos discos rotacionais, os algoritmos de escalonamento do braço do disco podem ser usados muitas vezes para melhorar o desempenho do disco, mas a presença de geometrias virtuais complica as coisas.
- Pareando dois discos, pode ser construído um meio de armazenamento estável com determinadas propriedades úteis.

- Uma geometria virtual possível para o disco físico da **figura (a)** é mostrada na **figura (b).** Em ambos os casos o disco tem 192 setores, apenas o arranjo publicado é diferente do real.



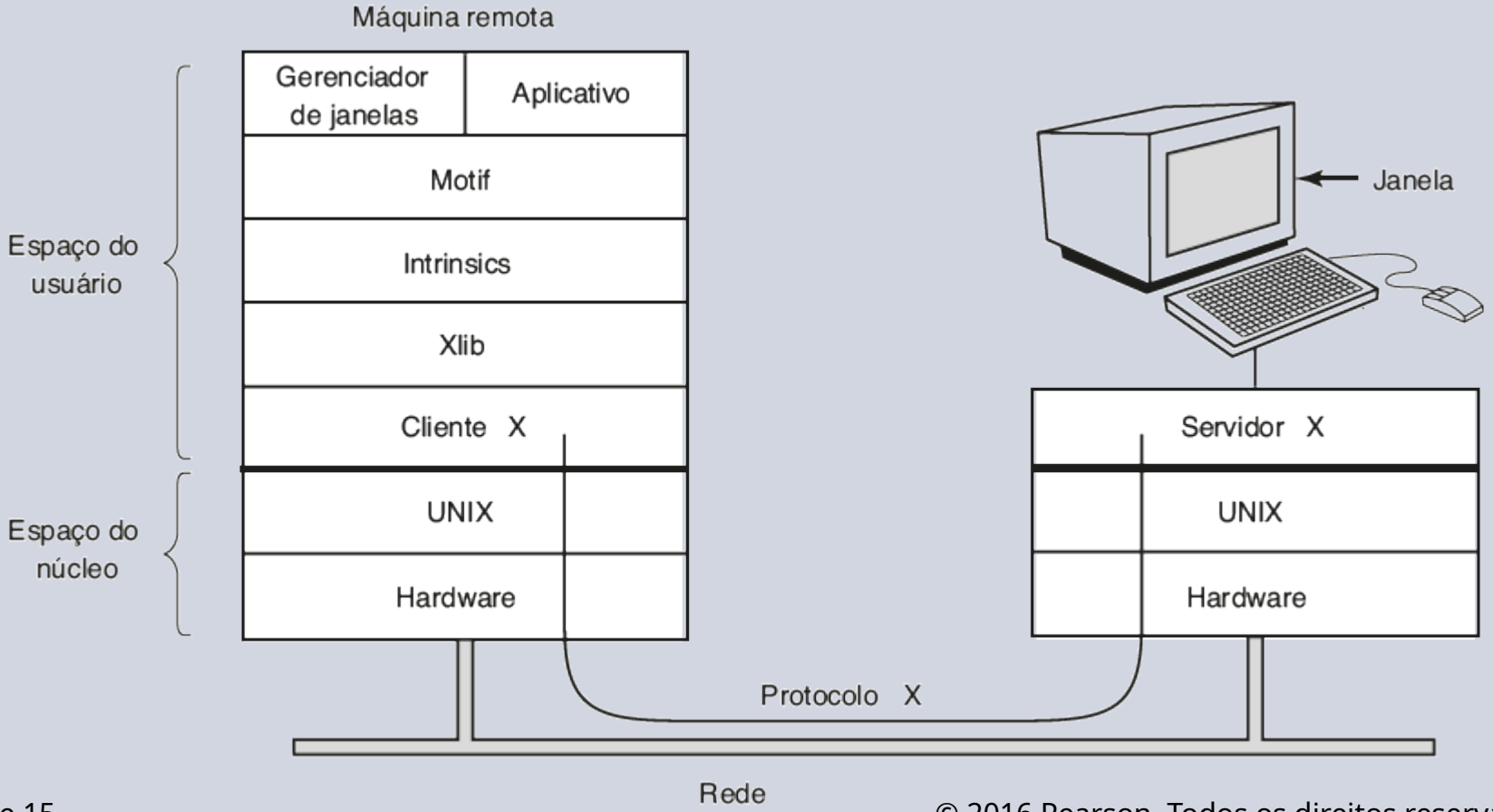
Relógios

- Relógios são usados para manter um controle do tempo real — limitando o tempo que os processos podem ser executados —, lidar com temporizadores *watchdog* e contabilizar o uso da CPU.

Interfaces com o usuário: teclado, mouse, monitor

- Terminais orientados por caracteres têm uma série de questões relativas a caracteres especiais que podem ser entrada e sequências de escape especiais que podem ser saída.
- A entrada pode ser em **modo cru** ou **modo cozido**, dependendo de quanto controle o programa quer sobre ela. Sequências de escape na saída controlam o movimento do cursor e permitem a inserção e remoção de texto na tela.

Clientes e servidores no sistema X Window do MIT



Interfaces com o usuário: teclado, mouse, monitor

- A maioria dos sistemas UNIX usa o Sistema X Window como base de sua interface do usuário. Ele consiste em programas que são ligados a bibliotecas especiais que emitem comandos de desenho e um servidor X que escreve na tela.
- Muitos computadores usam GUIs para sua saída. Esses são baseados no paradigma WIMP: janelas, ícones, menus e um dispositivo apontador (Windows, Icons, Menus, Pointing device).
- Programas baseados em GUIs são geralmente orientados a eventos, com eventos do teclado, mouse e outros sendo enviados para o programa para serem processados tão logo eles acontecem. Em sistemas UNIX, os GUIs quase sempre executam sobre o X.

Clientes magros (thin clients)

- Dizer que a maioria dos usuários quer uma computação interativa de alto desempenho, mas não quer realmente administrar um computador, provavelmente seja uma conclusão justa. Isso levou os pesquisadores a reexaminar os sistemas de tempo compartilhado usando terminais burros (agora educadamente chamados de **clientes magros**) que atendem às expectativas de terminais modernos.
- Clientes magros têm algumas vantagens sobre os PCs padrão, notavelmente por sua simplicidade e menos manutenção para os usuários.

Gerenciamento de energia

- É uma questão fundamental para telefones, tablets e notebooks, pois os tempos de vida das baterias são limitados, e para os computadores de mesa e de servidores devido às contas de luz da organização.
- Várias técnicas podem ser empregadas pelo sistema operacional para reduzir o consumo de energia.
- Programas também podem ajudar ao sacrificar alguma qualidade por mais tempo de vida das baterias.

Consumo de energia de um notebook

Dispositivo	Li et al. (1994)	Lorch e Smith (1998)
Tela	68%	39%
CPU	12%	18%
Disco rígido	20%	12%
Modem		6%
Som		2%
Memória	0,5%	1%
Outros		22%

Pesquisas em entrada/saída

- Há uma produção considerável de pesquisas sobre entrada/saída. Parte delas concentra-se em dispositivos específicos, em vez da E/S em geral. Outros trabalhos concentram-se na infraestrutura de E/S inteira.
- Por exemplo, a arquitetura Streamline busca fornecer E/S sob medida para cada aplicação, que minimize a sobrecarga devido a cópias, chaveamento de contextos, sinalização e uso equivocado da cache e TLB (DEBRUIJN et al., 2011)
- Ela é baseada na noção de Beltway Buffers, buffers circulares avançados que são mais eficientes do que os sistemas de buffers existentes (DEBRUIJN e BOS, 2008).

Pesquisas em entrada/saída

- Megapipe (HAN et al., 2012) é outra arquitetura de E/S em rede para cargas de trabalho orientadas a mensagens.
- Ela cria canais bidirecionais por núcleo de CPU entre o núcleo do SO e o espaço do usuário, sobre os quais os sistemas depositam camadas de abstrações como soquetes leves.
- Esses soquetes não são totalmente compatíveis com o POSIX, de maneira que aplicações precisam ser adaptadas para beneficiar-se da E/S mais eficiente.

Pesquisas em entrada/saída

- Muitas vezes, a meta da pesquisa é melhorar o desempenho de um dispositivo de uma maneira ou de outra. Os sistemas de discos são um desses casos. Os algoritmos de escalonamento de braço de disco são uma área de pesquisa sempre popular.
- Às vezes o foco é o uso mais baixo de energia.
- Com a popularidade da consolidação de servidores usando máquinas virtuais, o escalonamento de disco para sistemas virtualizados tornou-se um tópico em alta também.

Pesquisas em entrada/saída

- Drivers de dispositivos também são uma área de pesquisa muito ativa. Muitas falhas de sistemas operacionais são causadas por drivers de dispositivos defeituosos.
- Clientes magros também são um tópico que gera interesse, especialmente dispositivos móveis conectados à nuvem.
- Por fim, existem alguns estudos sobre tópicos incomuns como prédios como grandes dispositivos de E/S.