

# Aula 06

# Software de E/S

---

DISCIPLINA: SISTEMAS OPERACIONAIS II

PROFESSOR: Jheymesson Apolinário Cavalcanti

# Software de E/S

---

O sistema operacional é responsável por oferecer acesso aos dispositivos de entrada/saída às aplicações e, em consequência, aos usuários do sistema;

Prover acesso **eficiente, rápido e confiável** a um conjunto de periféricos com características diversas de comportamento, velocidade de transferência, volume de dados produzidos/consumidos e diferentes interfaces de hardware é um enorme desafio;

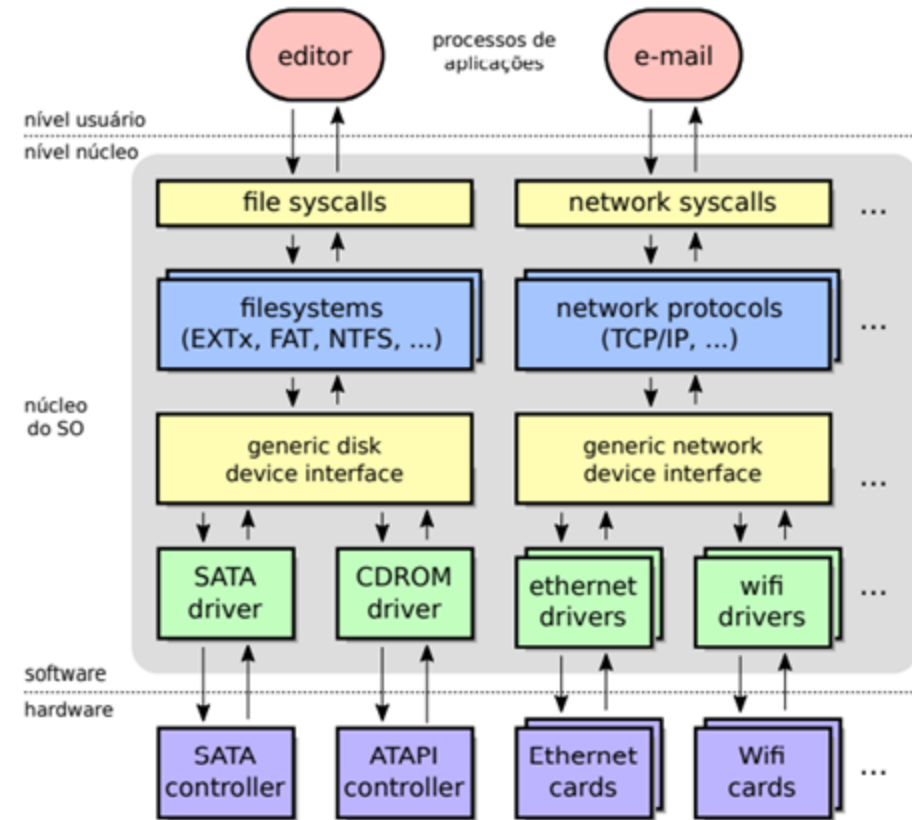
Além disso, como **cada dispositivo define sua própria interface** e modo de operação, o núcleo do sistema operacional deve implementar o código necessário para interagir com **milhares** de tipos de dispositivos distintos;

Como exemplo, cerca de **60%** das **20 milhões de linhas de código do núcleo Linux** na versão 4.3 pertencem ao **código de drivers** de dispositivos de entrada/saída.

# Software de E/S

## Arquitetura de software de entrada/saída;

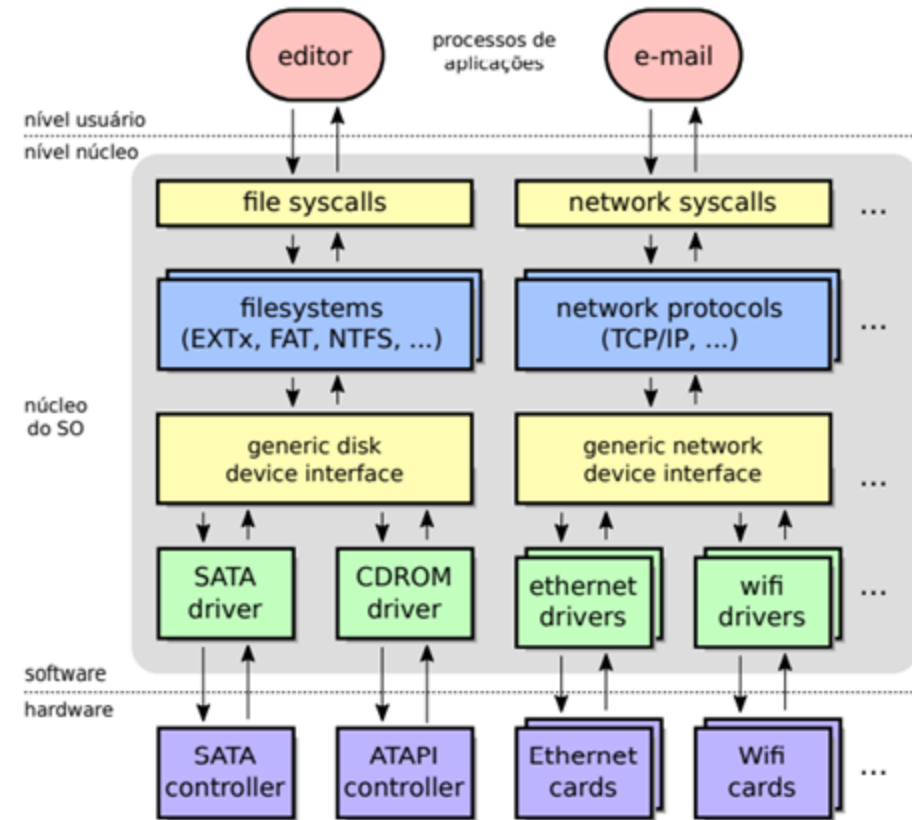
Para simplificar o uso e a gerência dos dispositivos de entrada/saída, **o código do sistema operacional é estruturado em camadas**, que levam da interação direta com o hardware às interfaces de acesso abstratas e genéricas oferecidas às aplicações, como arquivos e *sockets* de rede.



Estrutura em camadas do software de entrada/saída.

# Software de E/S

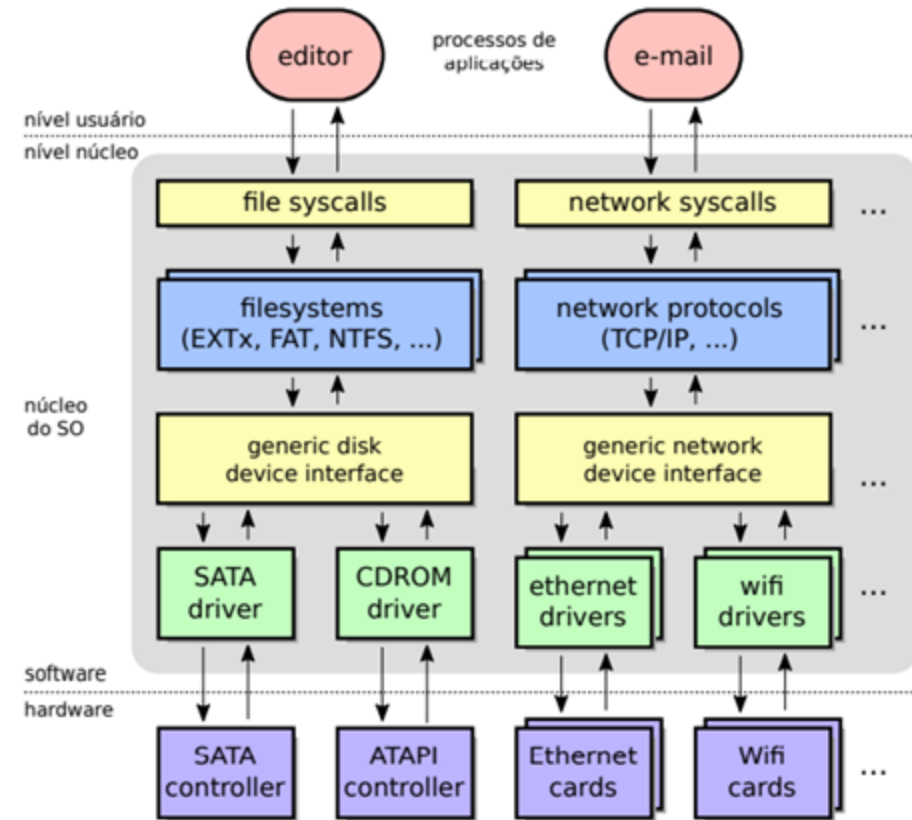
A primeira camada de software no núcleo do sistema operacional corresponde aos **drivers** de dispositivos, ou simplesmente drivers, que são os componentes de código que interagem diretamente com cada controlador, para realizar as operações de **entrada/saída**, receber as requisições de **interrupção** e fazer o **gerenciamento** do dispositivo correspondente.



Estrutura em camadas do software de entrada/saída.

# Software de E/S

Acima dos drivers existe uma camada de código, denominada **generic device interface**, cuja finalidade é **construir uma visão genérica de dispositivos similares**, para que o restante do sistema operacional não precise ter consciência das peculiaridades de cada dispositivo, mas possa tratá-los por **famílias** ou **classes**, como dispositivos de armazenamento, interfaces de rede, de vídeo, etc.

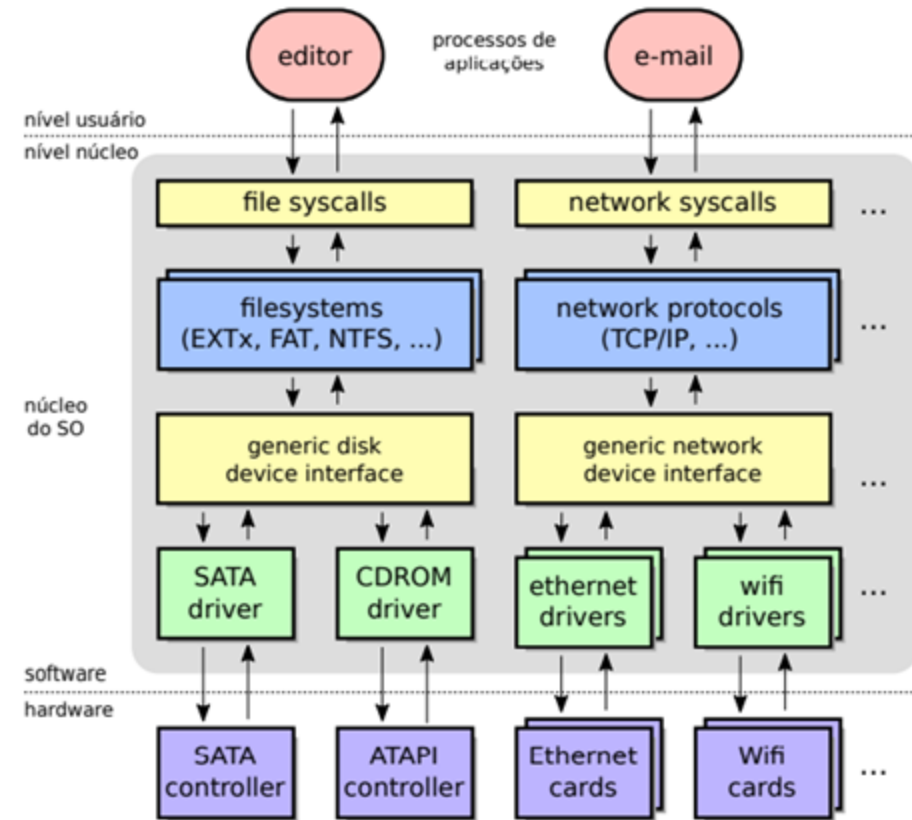


Estrutura em camadas do software de entrada/saída.

# Software de E/S

Acima da camada de interface genérica de dispositivos, uma ou mais camadas de código estão presentes, para implementar abstrações mais complexas, como **sistemas de arquivos** e **protocolos de rede**;

Finalmente, no topo da arquitetura de software, são implementadas as **chamadas de sistema** fornecidas às aplicações para acessar as abstrações construídas pelas camadas inferiores, como arquivos, diretórios e sockets de rede, etc.



Estrutura em camadas do software de entrada/saída.

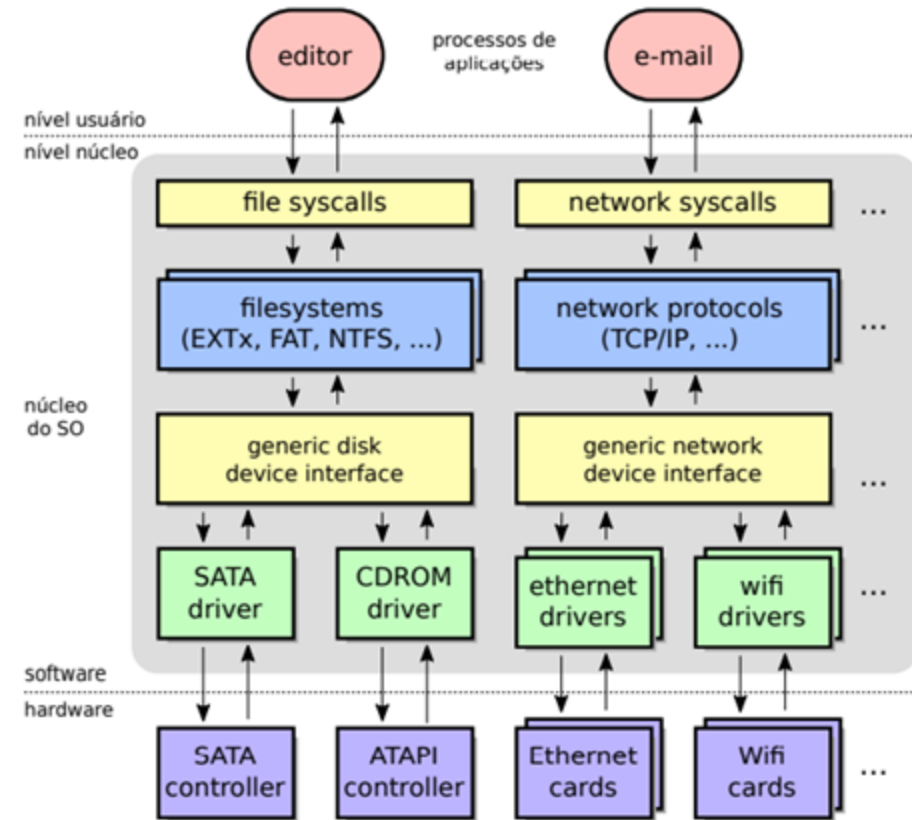
# Software de E/S

## Classes de dispositivos;

Os dispositivos de entrada/saída são geralmente agrupados em **classes** ou **famílias** com **características similares**, para os quais uma interface genérica pode ser definida;

Por exemplo, discos rígidos SATA, discos SSD e DVD-ROMs têm características mecânicas e elétricas distintas, mas servem basicamente para o mesmo propósito: **armazenar arquivos**.

O mesmo pode ser afirmado sobre interfaces de rede **Ethernet** e **Wifi**: embora usem tecnologias distintas, ambas permitem a comunicação entre computadores.

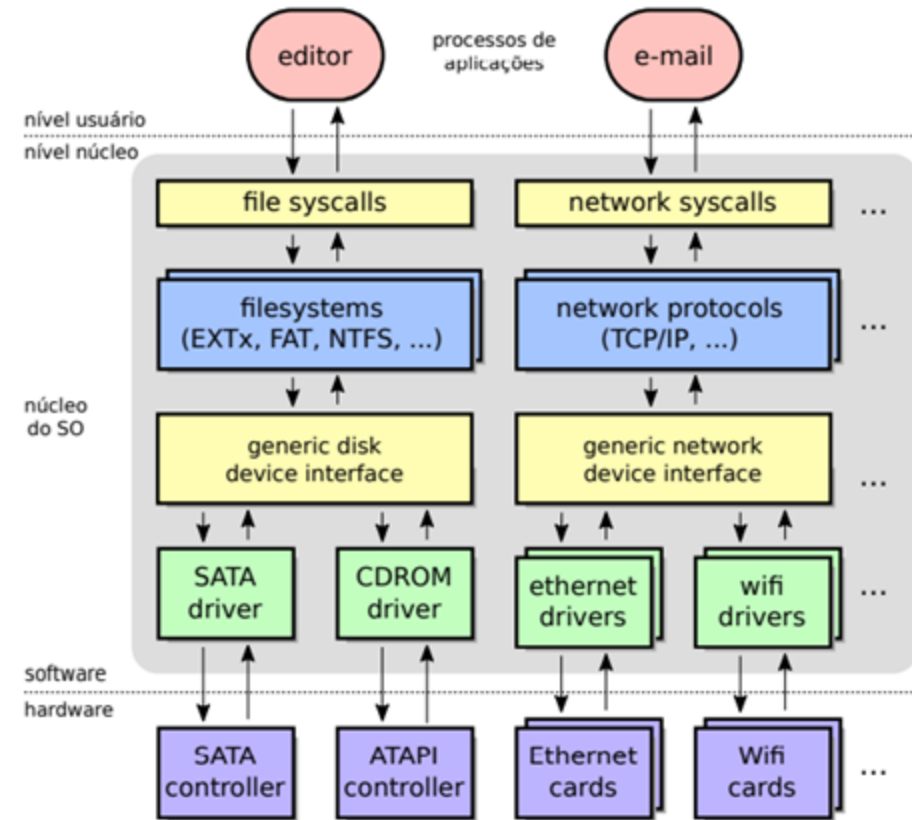


Estrutura em camadas do software de entrada/saída.

# Software de E/S

Nos sistemas de padrão UNIX os dispositivos são geralmente agrupados em **quatro grandes famílias**:

1. Dispositivos orientados a caracteres;
2. Dispositivos orientados a blocos;
3. Dispositivos de rede;
4. Dispositivos gráficos.



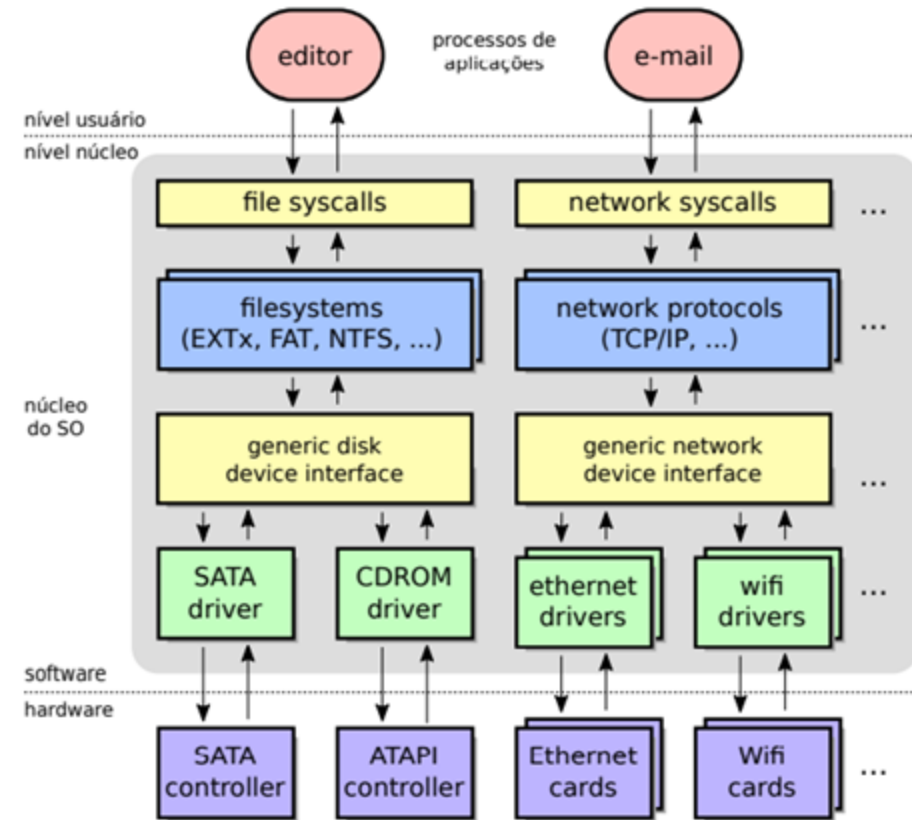
Estrutura em camadas do software de entrada/saída.



# Software de E/S

## 1. Dispositivos orientados a caracteres:

- a. São aqueles cujas transferências de dados são sempre feitas byte por byte, em sequência;
- b. Pode ser visto como um fluxo contínuo de entrada ou de saída de bytes;
- c. A característica sequencial faz com que não seja possível alterar o valor de um byte que já foi enviado;
- d. O **mouse** e **teclado**, são os exemplos mais clássicos desta família.

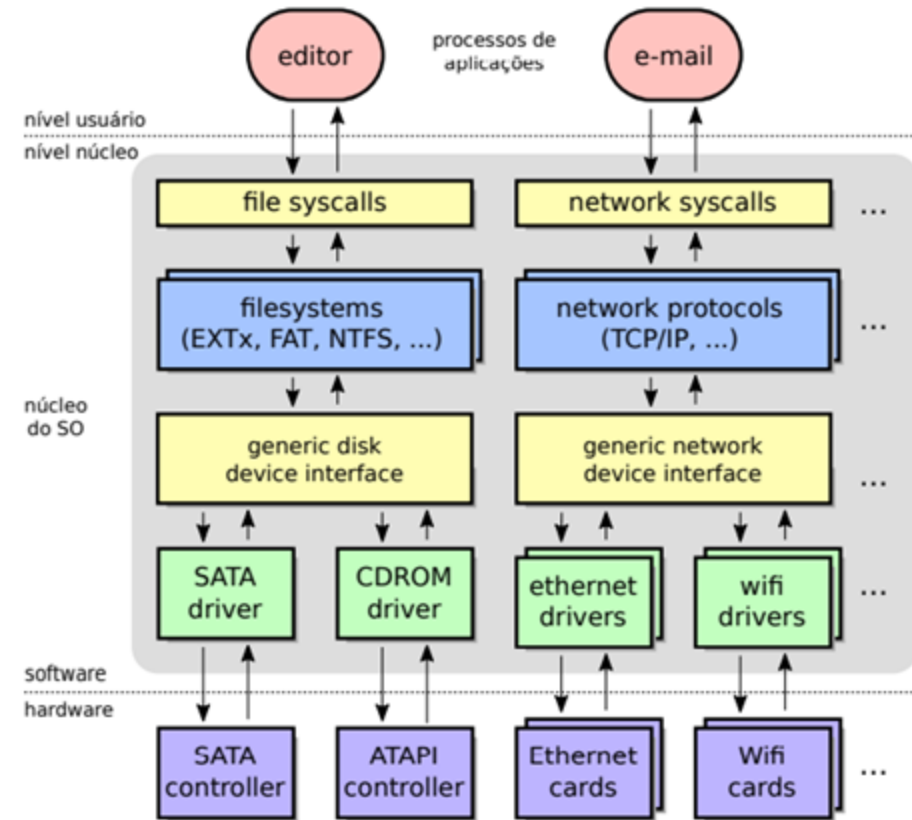


Estrutura em camadas do software de entrada/saída.

# Software de E/S

## 2. Dispositivos orientados a blocos:

- a. São aqueles dispositivos em que as operações de entrada ou saída de dados são feitas usando blocos de bytes de tamanho fixo;
- b. Esses blocos são lidos ou escritos em posições específicas do dispositivo, ou seja, são endereçáveis;
- c. Discos rígidos e outros dispositivos de **armazenamento** são exemplos típicos desta família.

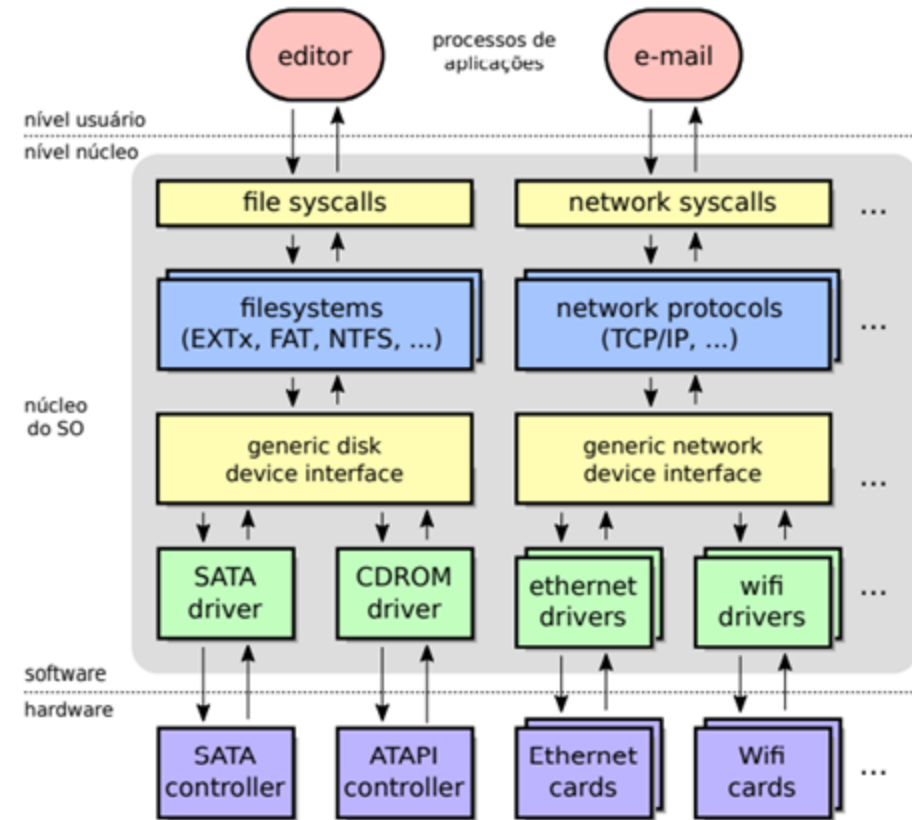


Estrutura em camadas do software de entrada/saída.

# Software de E/S

## 3. Dispositivos de rede:

- a. Estes dispositivos permitem **enviar e receber** mensagens entre processos e computadores distintos;
- b. As interfaces **Ethernet, Wifi, Bluetooth e GPRS** são bons exemplos desta classe de dispositivos.

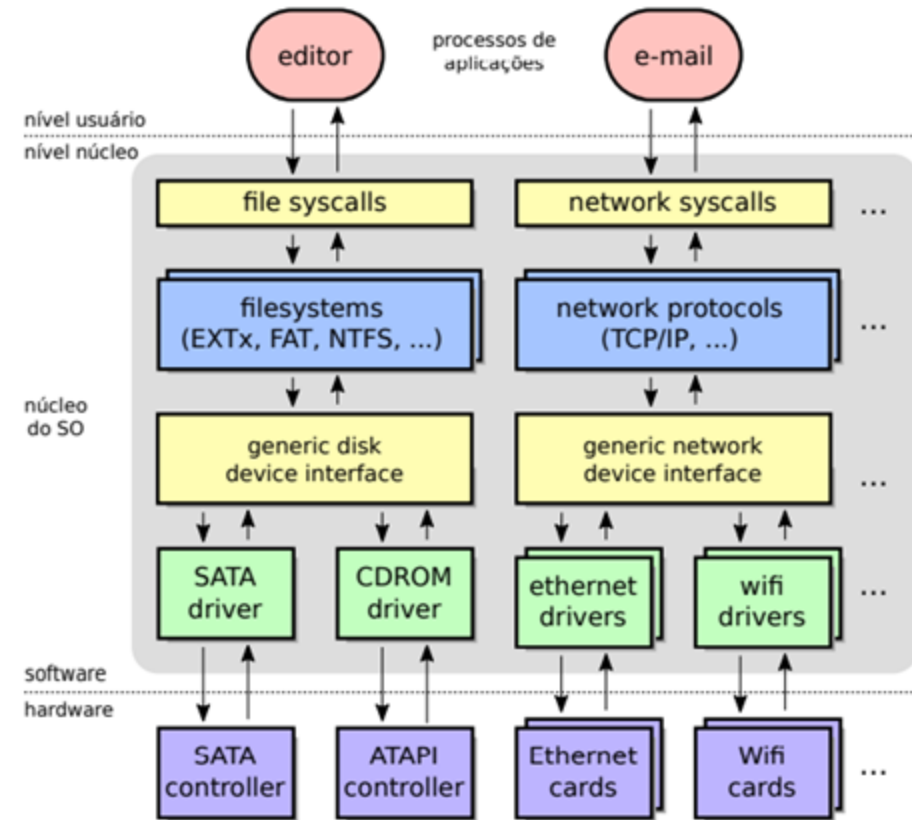


Estrutura em camadas do software de entrada/saída.

# Software de E/S

## 4. Dispositivos gráficos:

- a. Permitem a **renderização** de **texto** e **gráficos** em terminais de vídeo;
- b. Exigem um alto desempenho na transferência de dados;
- c. Sua interface genérica é constituída por funções para consultar e configurar o dispositivo gráfico e uma área de memória compartilhada entre o processador e o dispositivo, usualmente denominada **frame buffer**, que permite acesso direto à memória de vídeo.
- d. *DirectX* em ambientes Windows ou *DRI – Direct Rendering Engine* no Linux.



Estrutura em camadas do software de entrada/saída.

# Software de E/S (Drivers)

---

Um driver é um **componente do sistema operacional** responsável por **interagir** com um **controlador** de dispositivo;

Cada tipo de dispositivo possui seu **próprio driver**, muitas vezes fornecido pelo fabricante do mesmo;

Cada driver é geralmente capaz de tratar um **único tipo de dispositivo**, ou uma **família de dispositivos correlatos** do mesmo **fabricante**.

# Software de E/S (Drivers)

---

Um **driver** consiste de um **conjunto de funções** que são ativadas pelo núcleo do sistema operacional conforme necessário.

Existem basicamente três grupos de funções implementadas por um driver:

1. Funções de entrada/saída;
2. Funções de gerência;
3. Funções de tratamento de eventos.

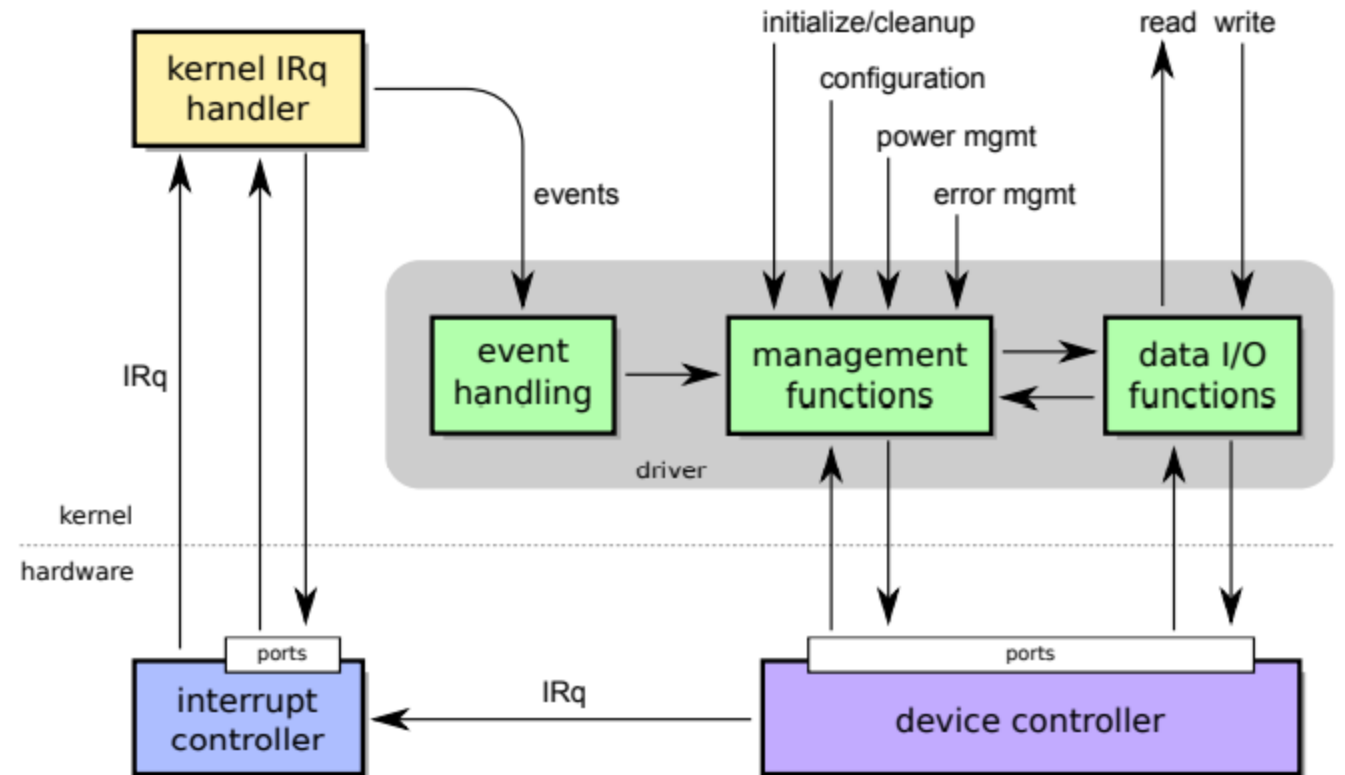
Executam dentro do **núcleo do sistema operacional**, em **modo privilegiado**;

Por ser código de terceiros executando com acesso total ao hardware, eles constituem um dos maiores riscos à estabilidade e segurança do sistema operacional.

# Software de E/S (Drivers)

Funções implementadas por um driver:

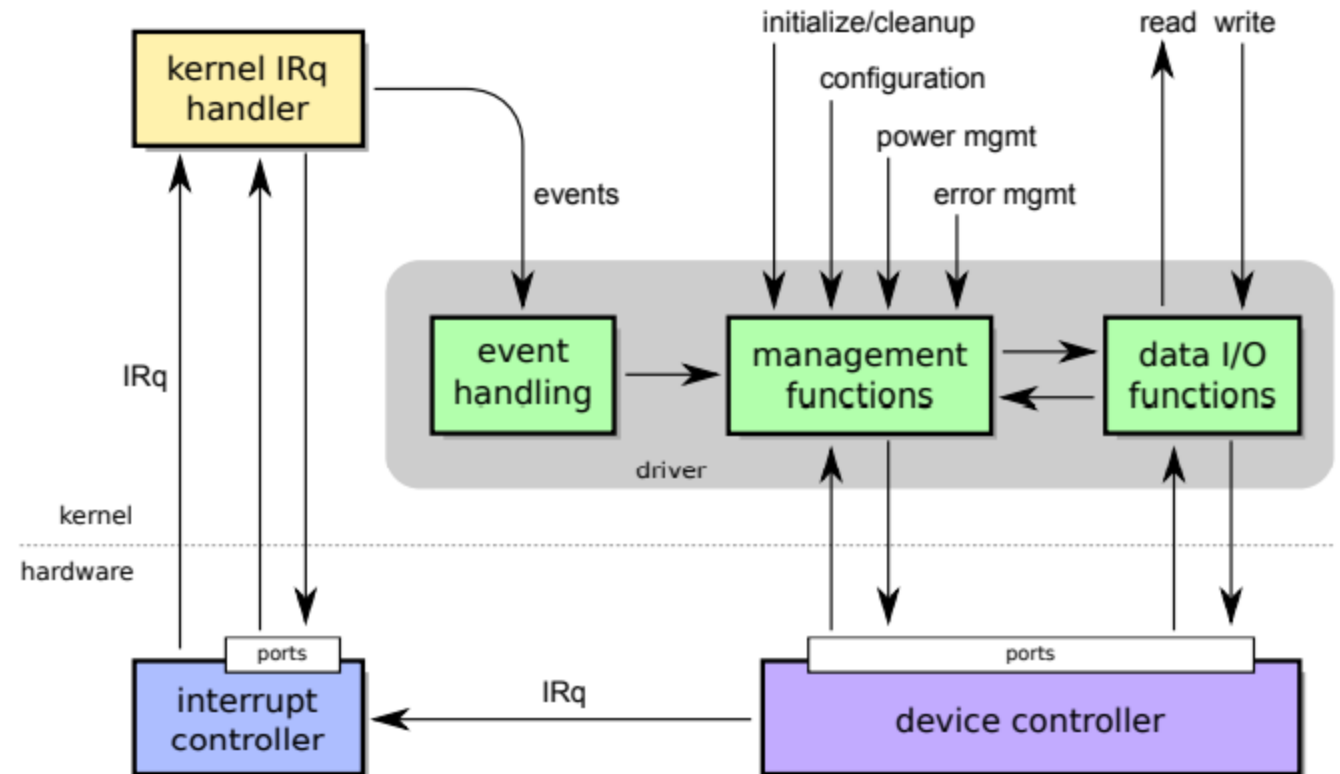
1. Funções de entrada/saída;
2. Funções de gerência;
3. Funções de tratamento de eventos.



# Software de E/S (Drivers)

## 1. Funções de entrada/saída:

- a. Responsáveis pela transferência de dados entre o dispositivo e o sistema operacional;
- b. **Recebem e enviam** dados de acordo com a classe dos dispositivos: caracteres (bytes), blocos de tamanho fixo (discos), blocos de tamanho variável (pacotes de rede), etc.

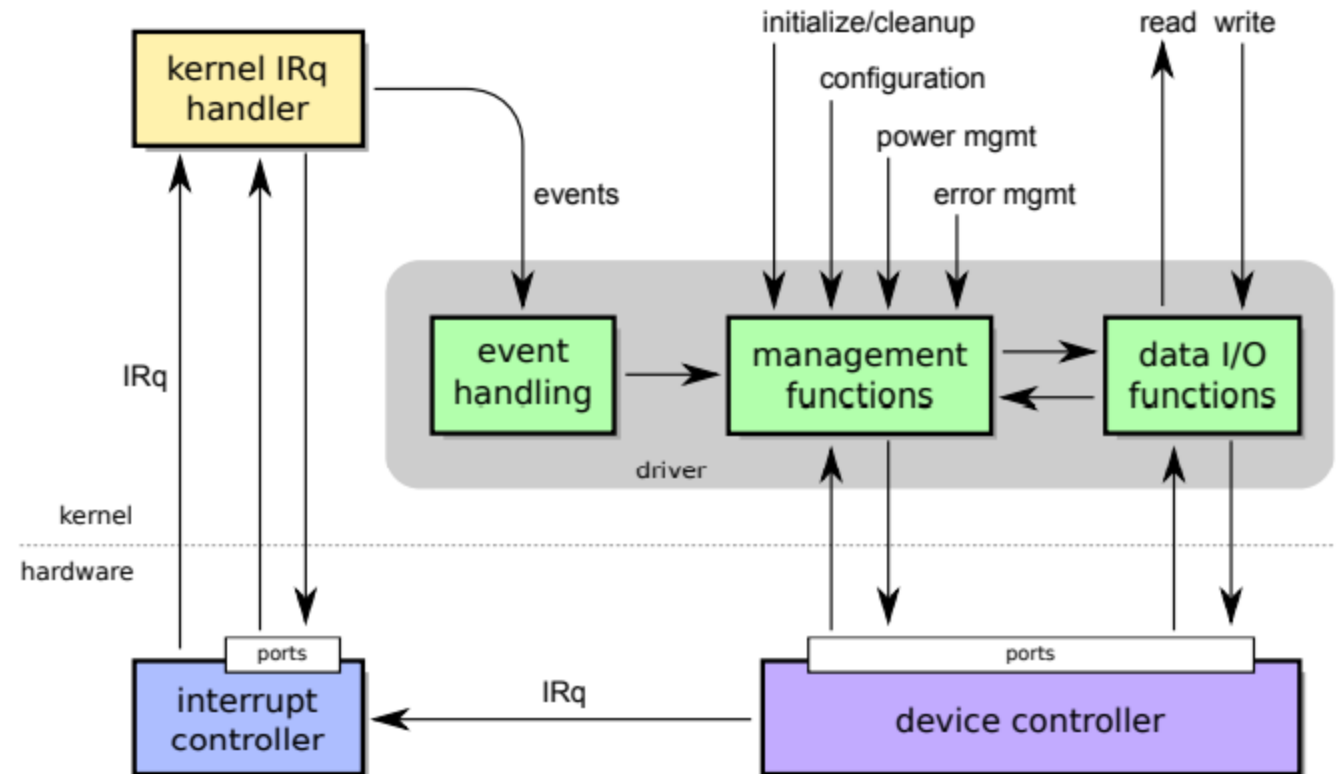




# Software de E/S (Drivers)

## 2. Funções de gerência:

- a. Responsáveis pela gestão do **dispositivo** e do **próprio driver**;
- b. Geralmente são fornecidas funções para configurar o dispositivo, para **desligar** ou colocar em **espera** quando este não for usado, e para tratar **erros** no dispositivo.



# Software de E/S (Drivers)

## 3. Funções de tratamento de eventos:

- a. Estas funções são ativadas quando uma **requisição de interrupção** é gerada pelo dispositivo;
- b. Toda requisição de interrupção gerada pelo dispositivo é encaminhada ao **controlador de interrupções do hardware**;
- c. No núcleo, um tratador de interrupções (*IRq handler*) **reconhece e identifica** a interrupção.

