MC504 - Sistemas Operacionais

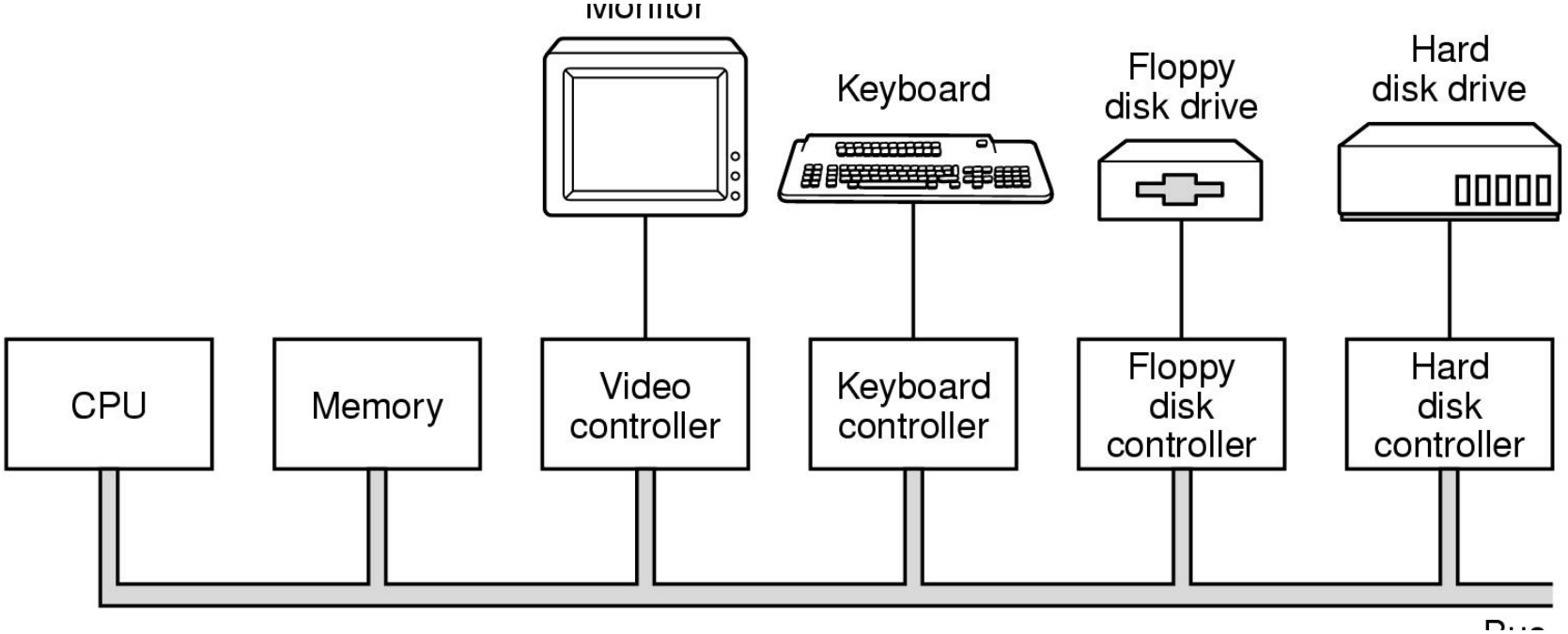
Entrada e Saída
Device drivers

Islene Calciolari Garcia
Instituto de Computação - Unicamp
Primeiro Semestre de 2017

Sumário

Dispositivos de I/O e controladores

Visão simplificada

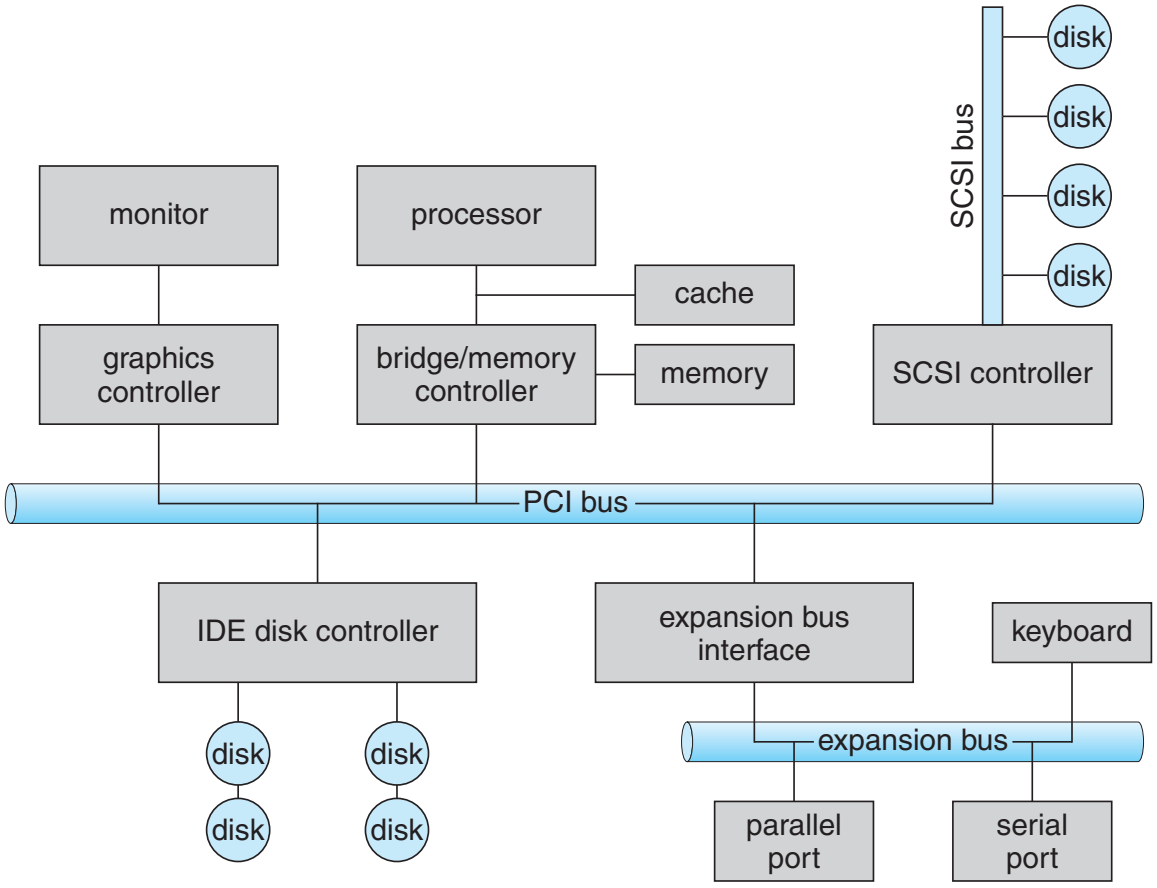


Tanenbaum: Figura 1.5

O sistema operacional deve interagir com os controladores

Dispositivos de I/O e controladores

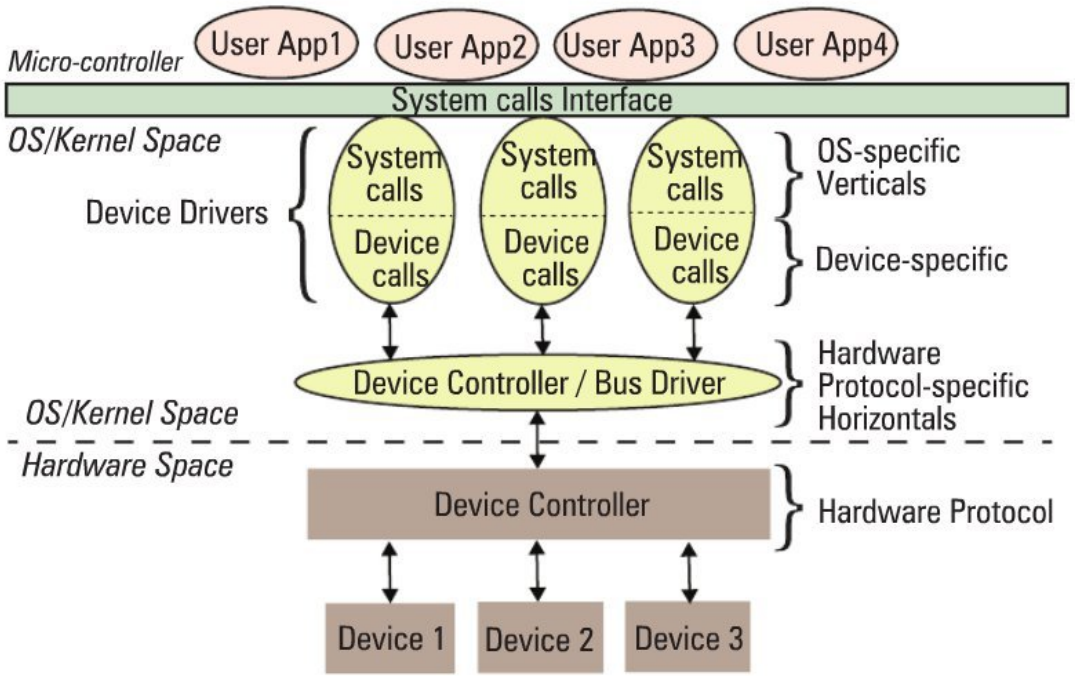
Visão com barramentos



Silberschatz: Figura 13.1

Device drivers

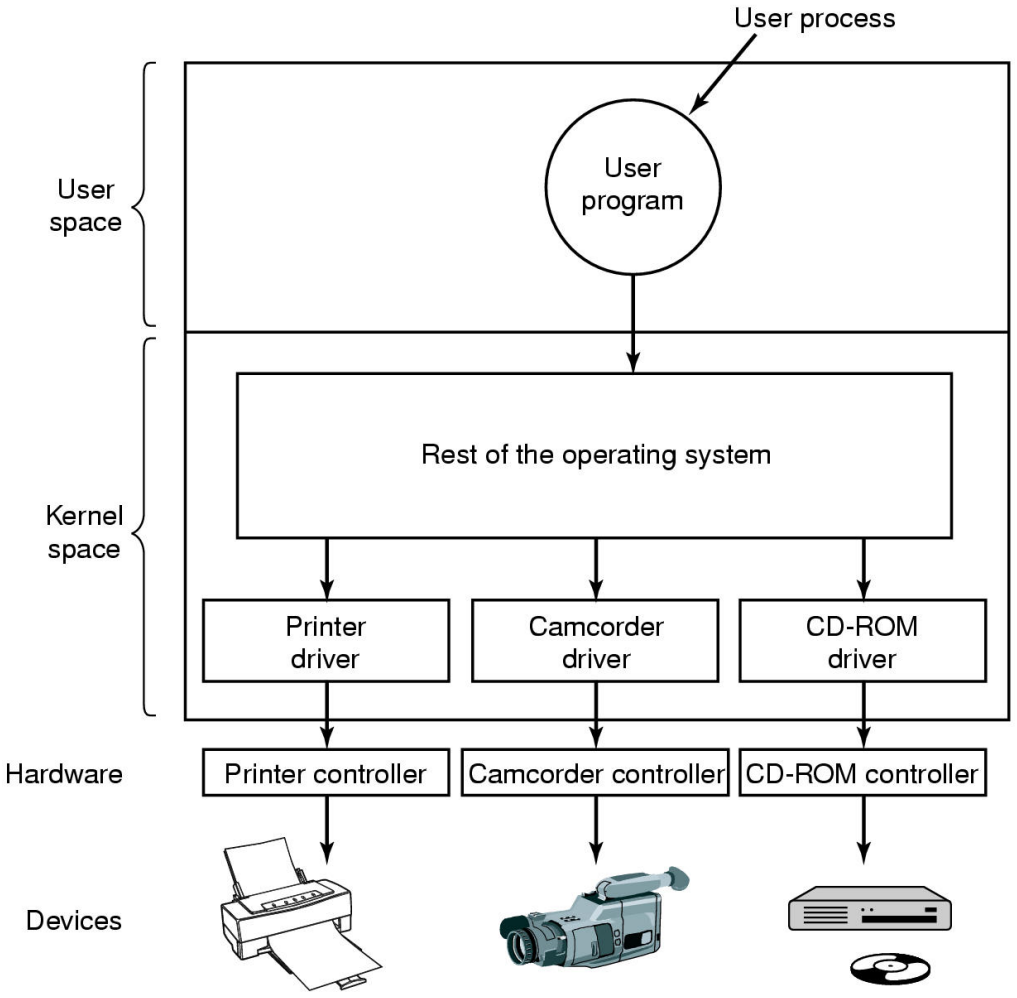
- Software que “conversa” com o controlador
- Fabricantes devem fornecer dados detalhados



Linux Device Drivers Series

Device drivers

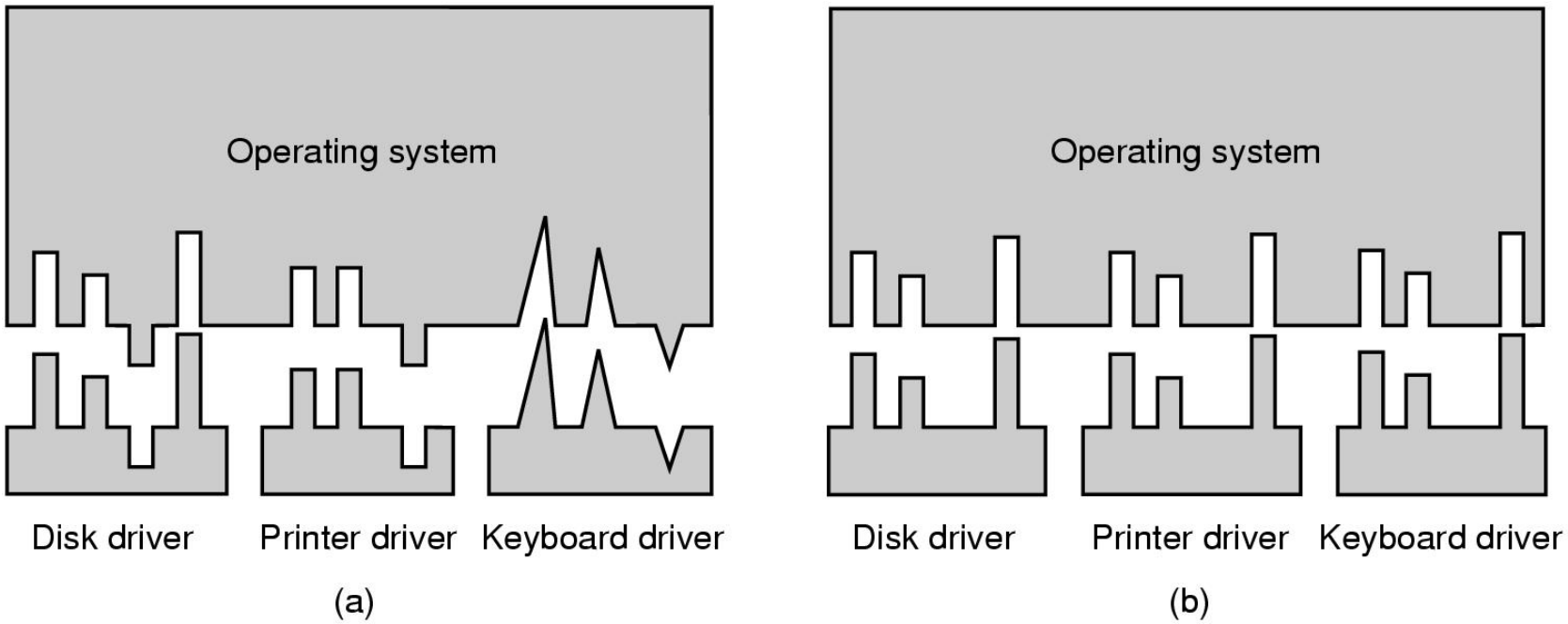
Como fazer a interface?



Tanenbaum: Figura 5.11

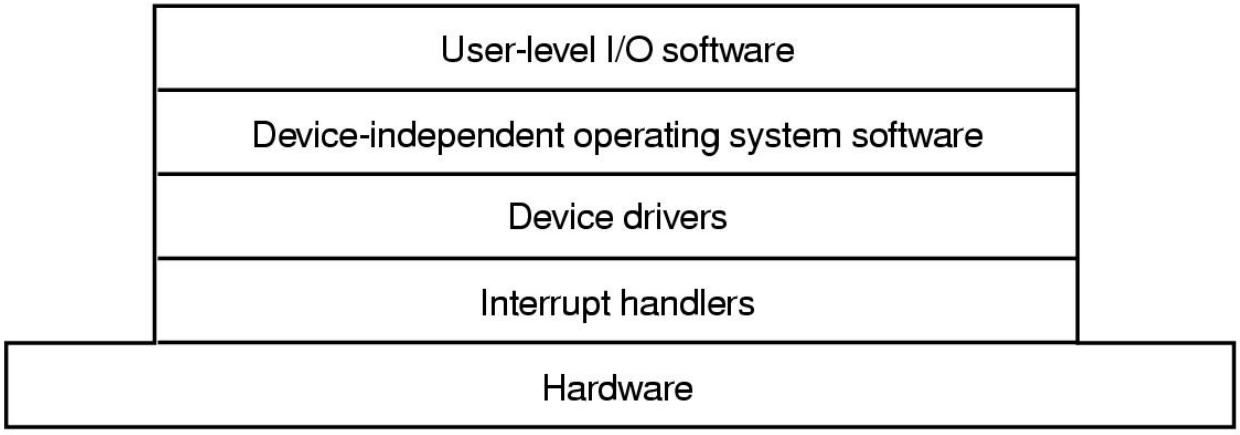
Device drivers

Sem ou com uma interface padrão



Tanenbaum: Figura 5.13

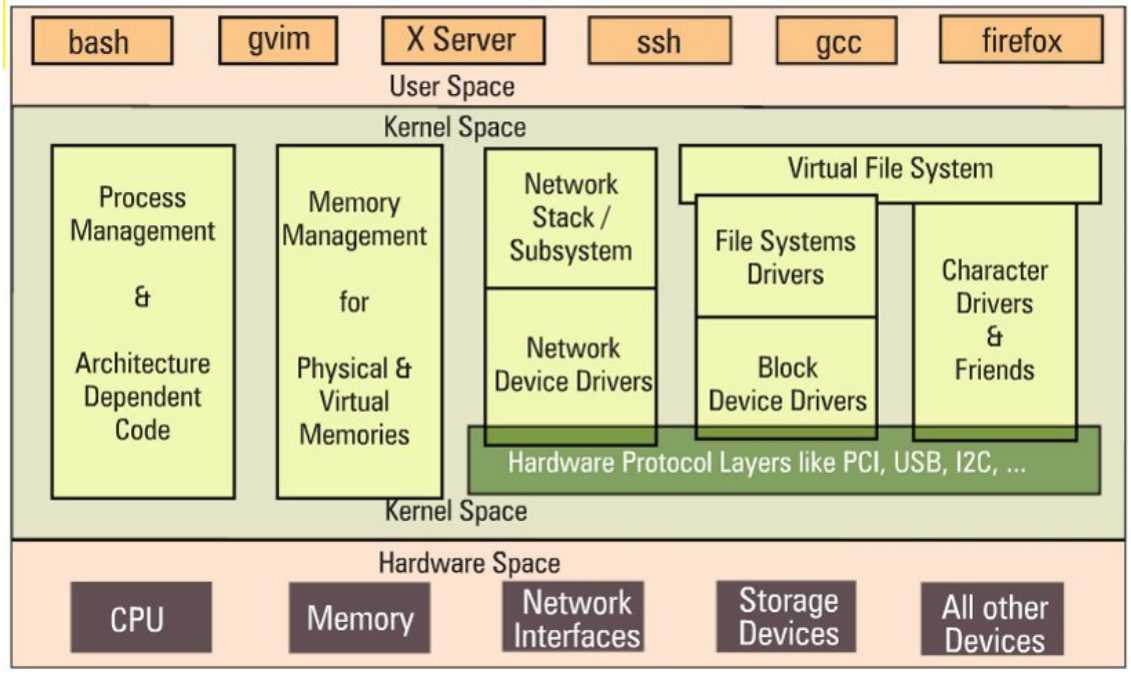
Camadas de software



Tanenbaum: Figura 5.10

Camadas de software

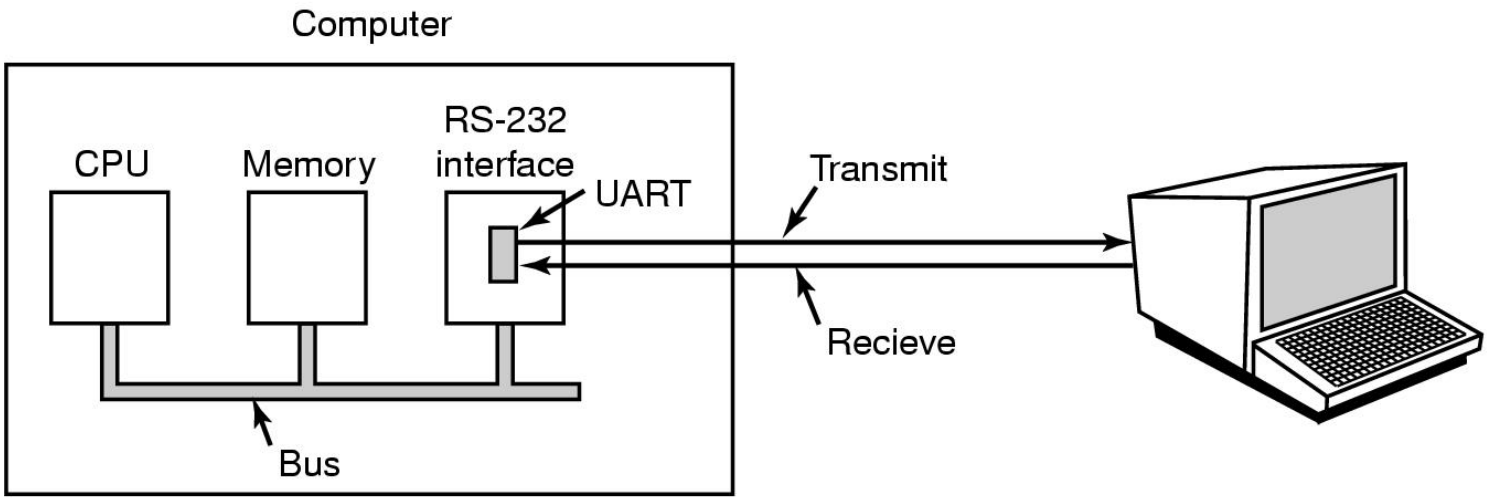
- Tipos diferentes de dispositivo



Linux Device Drivers Series

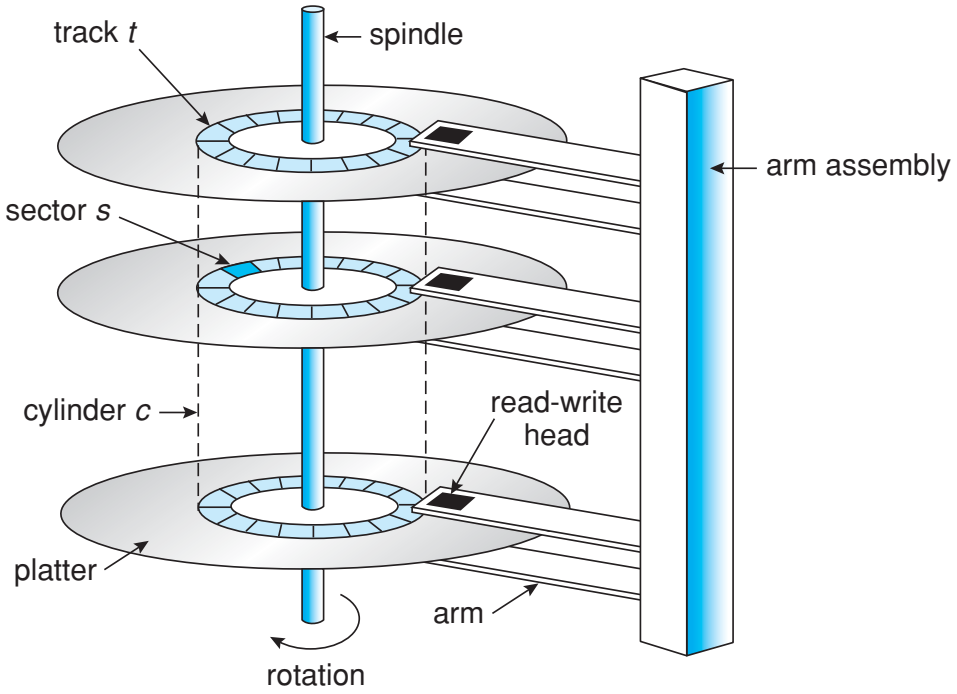
Character device

- Acesso sequencial, caractere a caractere
- Não permite retrocesso...
- Execute `ls -l /dev | grep tty`



Tanenbaum: Figura 5.34

Block device



Silberschatz: Figura 10.1

Acesso não sequencial a blocos de informação

Execute `ls -l /dev | grep sda`

Network device

- Comunicação orientada a pacotes
- Buffer deve poder receber pacotes inteiros
- Trabalhar com redes pode ser mais complexo do que com arquivos...

Endereços de portas

I/O address range (hexadecimal)	device
000–00F	DMA controller
020–021	interrupt controller
040–043	timer
200–20F	game controller
2F8–2FF	serial port (secondary)
320–32F	hard-disk controller
378–37F	parallel port
3D0–3DF	graphics controller
3F0–3F7	diskette-drive controller
3F8–3FF	serial port (primary)

Silberschatz: Figura 13.2

Como programar os dispositivos?

- Instruções especiais

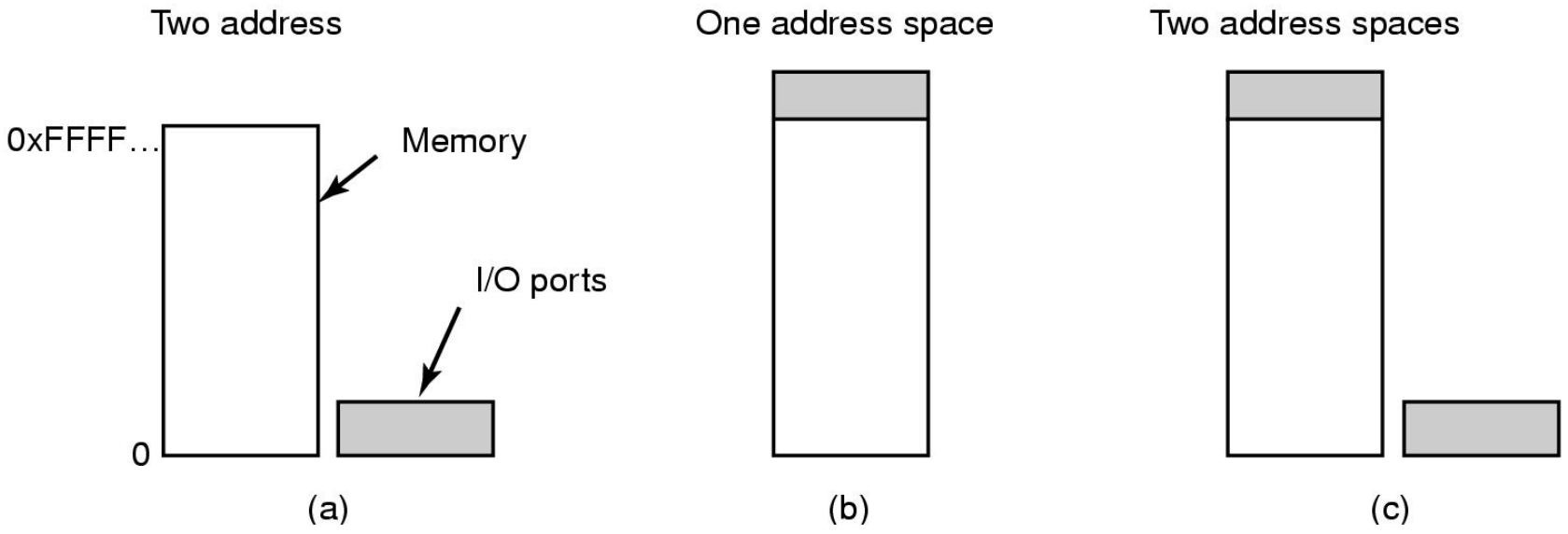
  ```
  IN REG, PORT
  OUT PORT, REG
  ```

- Memory-mapped I/O

  ```
  MOV REG, ADDR
  ```

  Conforme o valor de ADDR, a instrução MOV fará acesso a uma palavra de memória ou dispositivo

Como programar os dispositivos?



Tanenbaum: Figura 5.2

Imprimindo uma string

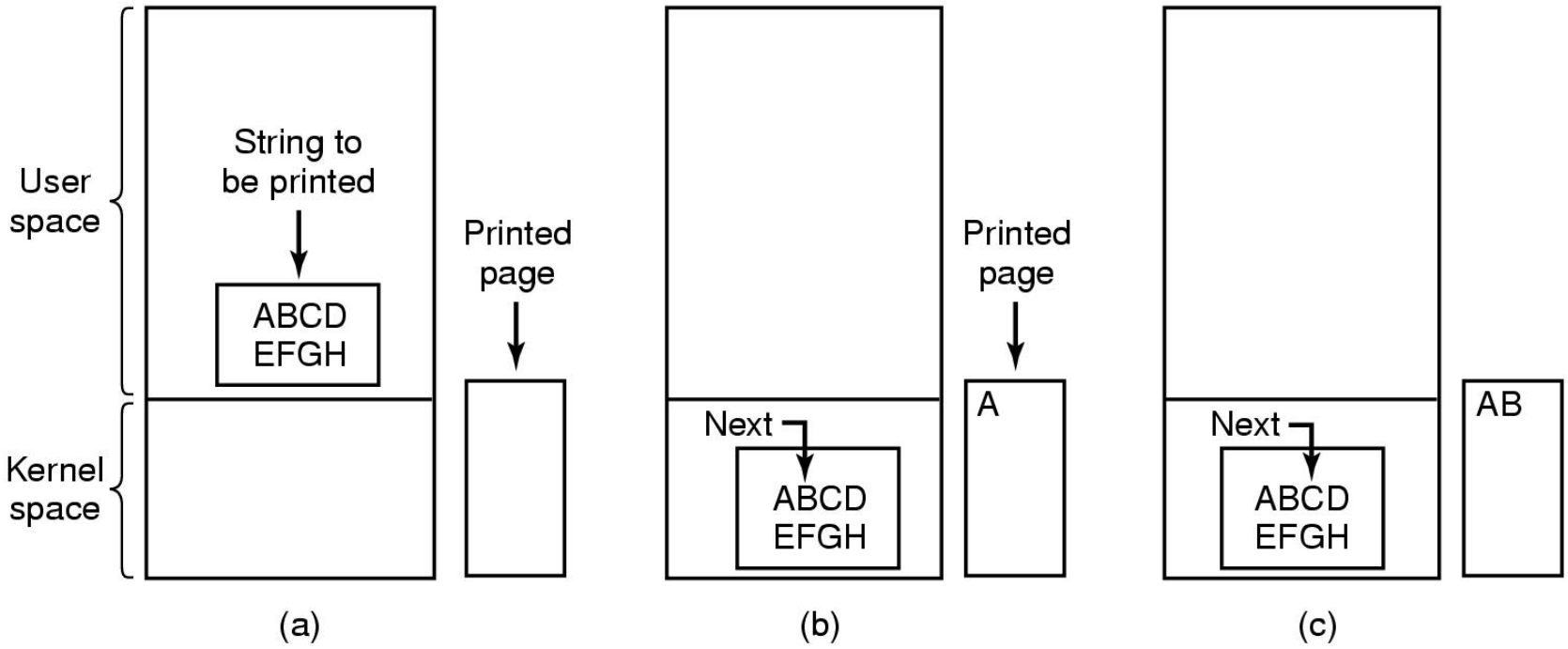
Programmed I/O

```
copy_from_user(buffer, p, count);                /* p is the kernel bufer */
for (i = 0; i < count; i++) {                    /* loop on every character */
        while (*printer_status_reg != READY) ;   /* loop until ready */
        *printer_data_register = p[i];           /* output one character */
}
return_to_user( );
```

Tanenbaum: Figura 5.7

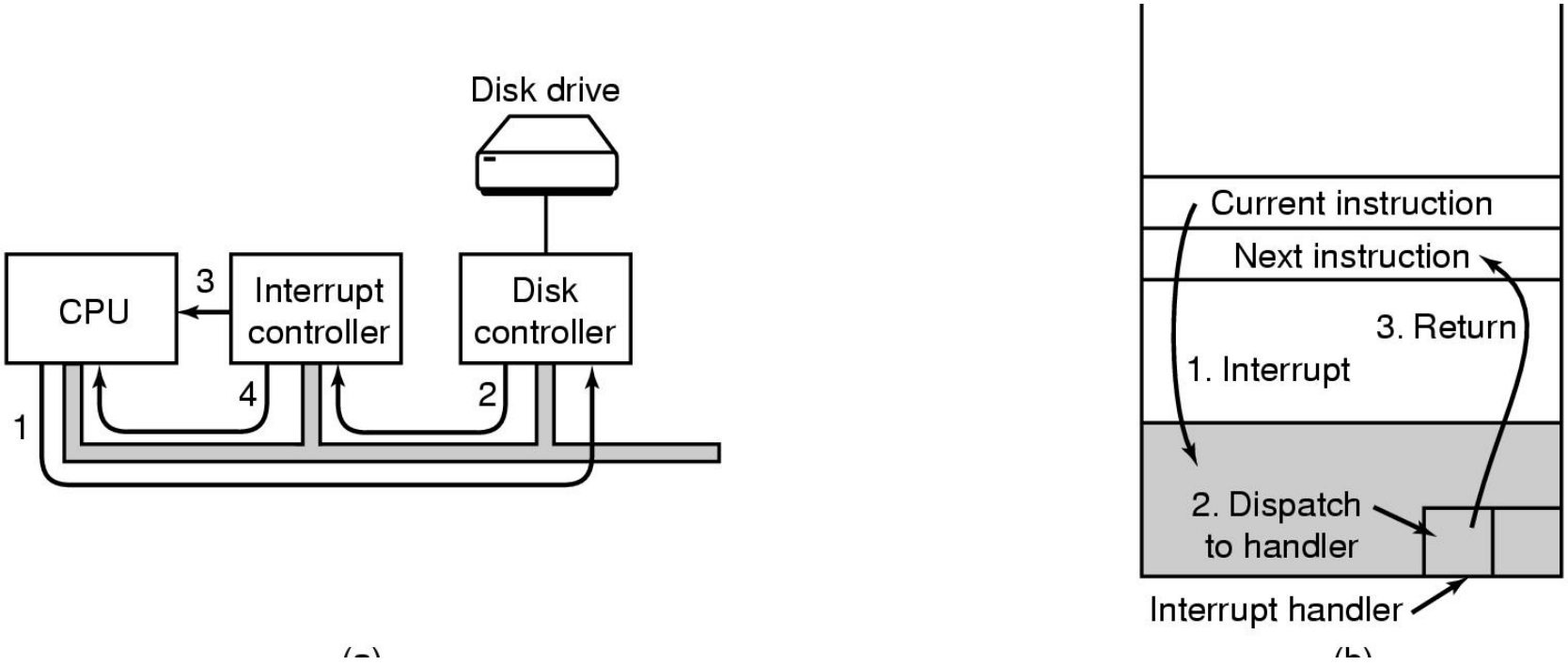
Trecho de código do kernel

Imprimindo uma string



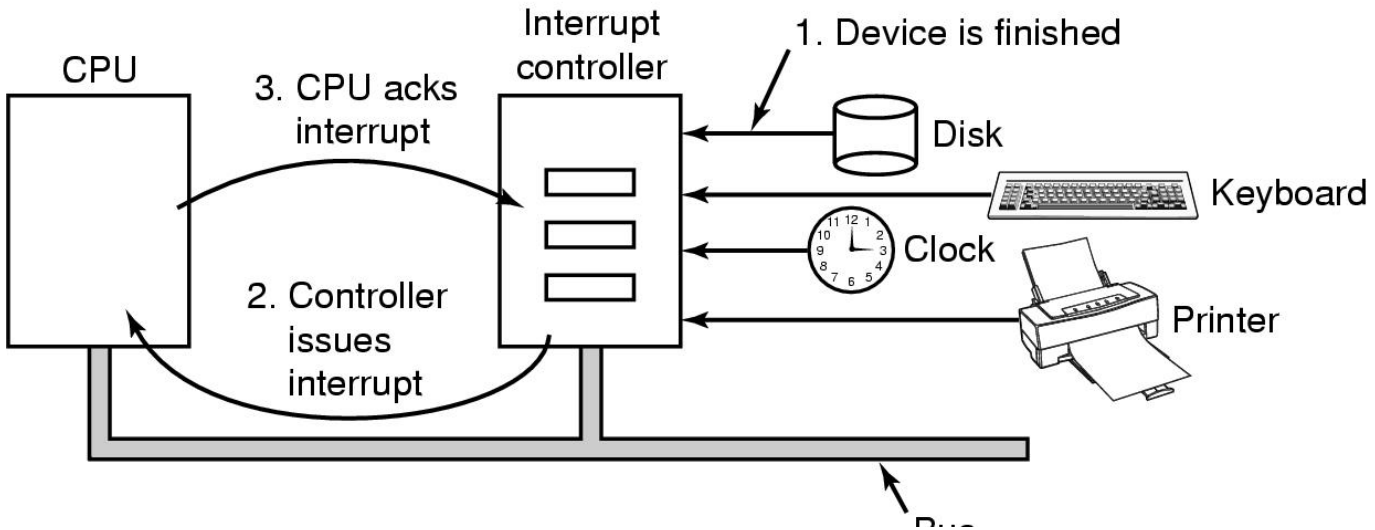
Tanenbaum: Figura 5.6

Tratamento de interrupções



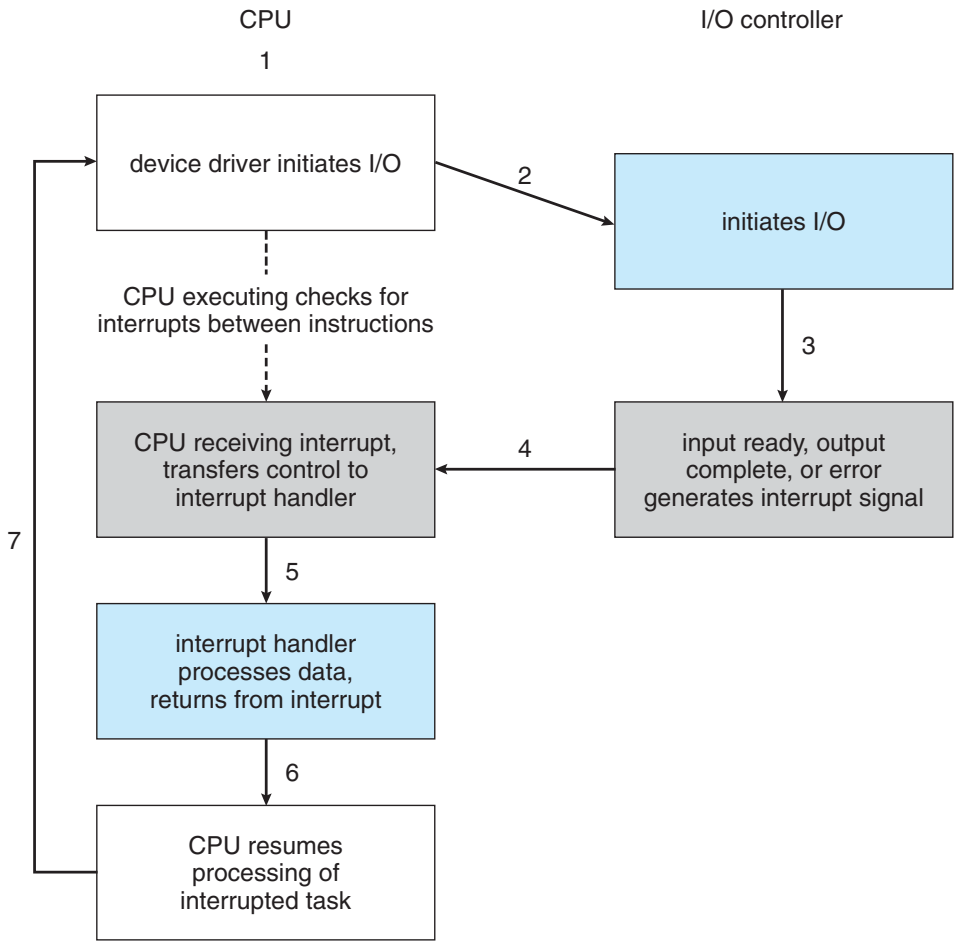
Tanenbaum: Figura 1.10

Tratamento de interrupções



Tanenbaum: Figura 5.5

Tratamento de interrupções



Silberschatz: Figura 13.3

Imprimindo uma string

Interrupt-driven I/O

```
copy_from_user(buffer, p, count);
enable_interrupts( );
while (*printer_status_reg != READY) ;
*printer_data_register = p[0];
scheduler( );
```

(a)

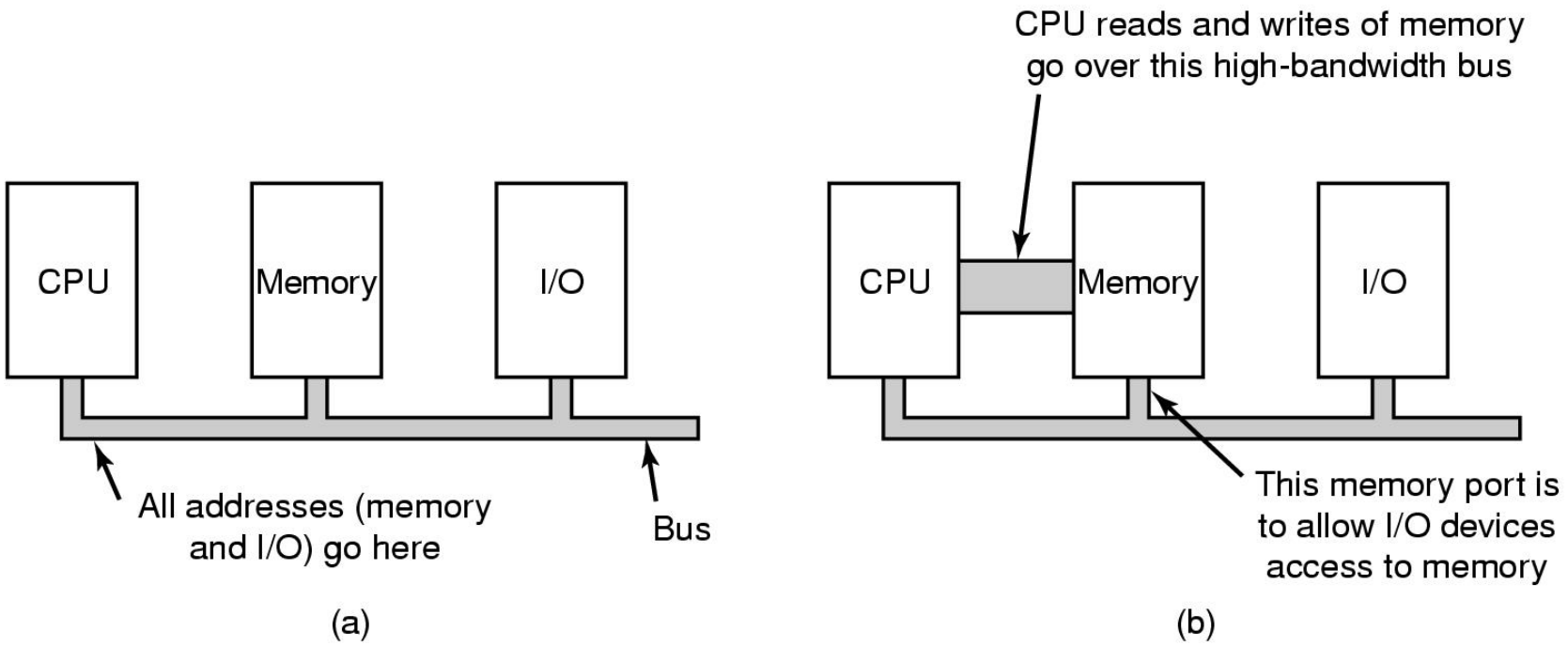
```
if (count == 0) {
    unblock_user( );
} else {
    *printer_data_register = p[i];
    count = count – 1;
     i = i + 1;
}
acknowledge_interrupt( );
return_from_interrupt( );
```

(b)

Tanenbaum: Figura 5.8

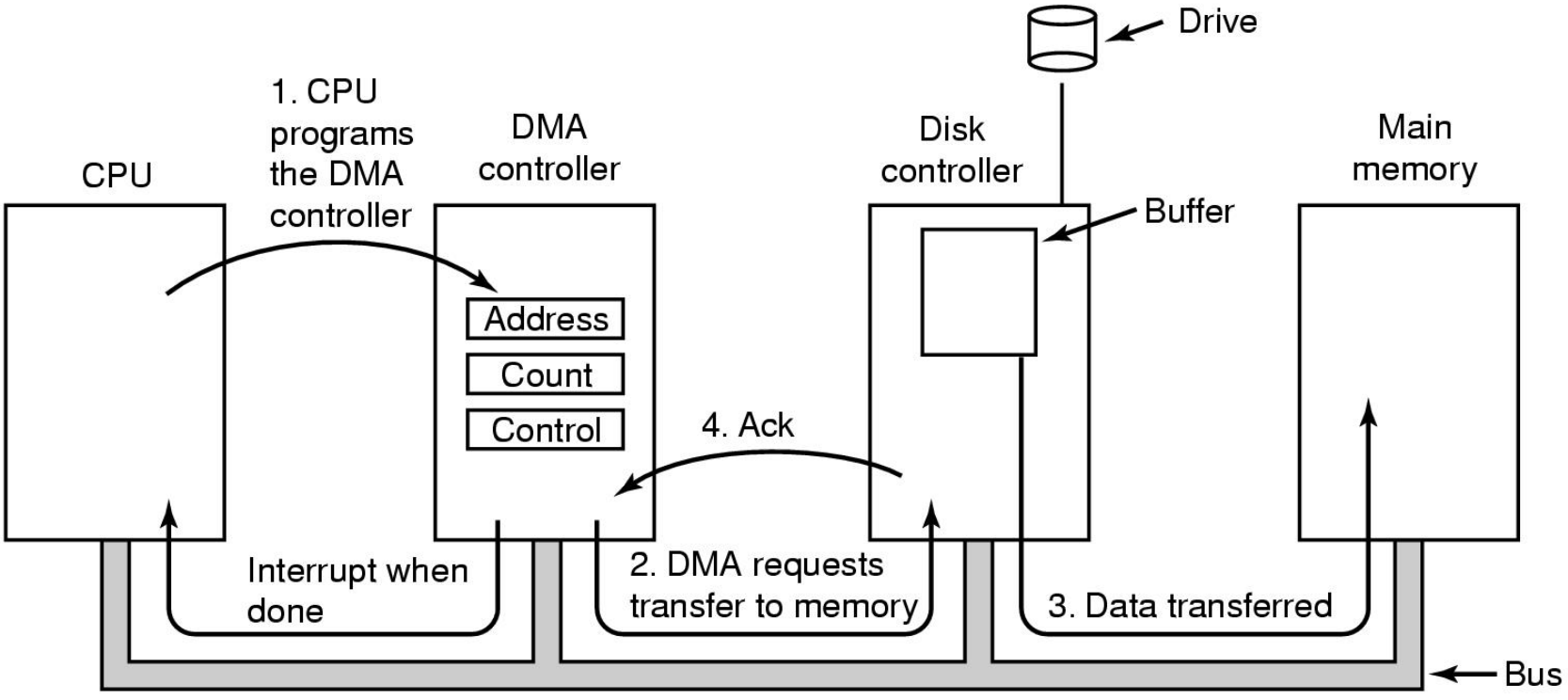
(a) Trecho de código do kernel
(b) Tratador da interrupção

Barramento simples e dual



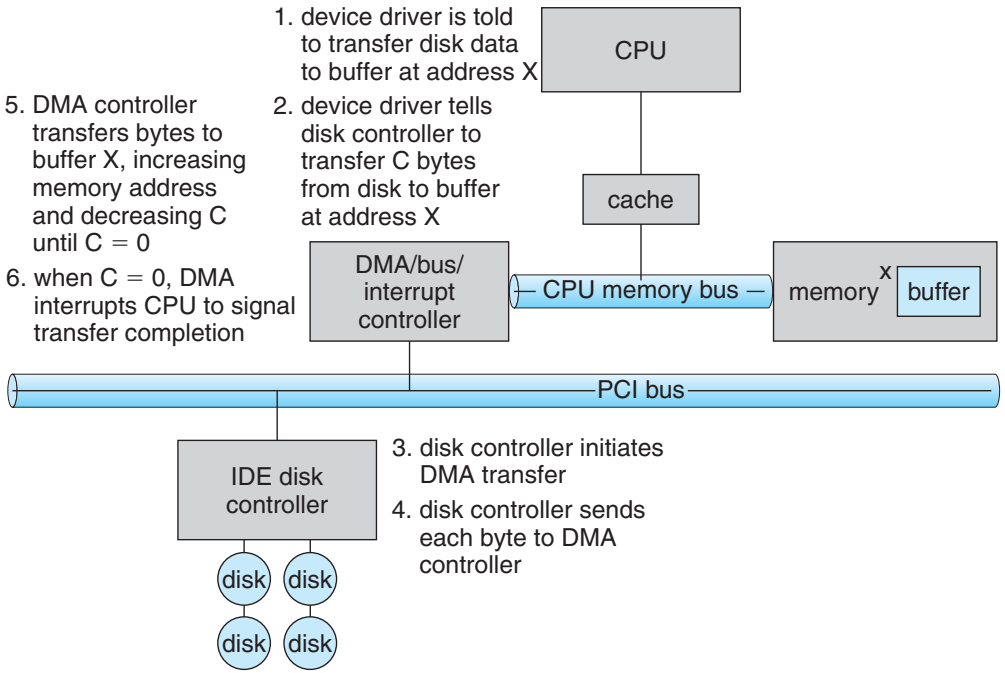
Tanenbaum: Figura 5.3

Direct Memory Access (DMA)



Tanenbaum: Figura 5.4

Tratamento de interrupções



Silberschatz: Figura 13.5

Imprimindo uma string

DMA

```
copy_from_user(buffer, p, count);
set_up_DMA_controller( );
scheduler( );
```

(a)

```
acknowledge_interrupt( );
unblock_user( );
return_from_interrupt( );
```

(b)

(a) Trecho de código do kernel
(b) Tratador de interrupção

Device drivers

- *Major and minor numbers*
  - Tradicionalmente, *major* identifica o driver e *minor* o dispositivo.
  - Múltiplos *drivers* podem compartilham um *major number*.
- Como acoplar um device driver ao kernel:
  - *relink* e *reboot*
  - entrada em um arquivo e reboot
  - on-the-fly
    veja o comando `lsmod`

Programando um device driver

- Veja a série: Device drivers de Anil Kumar Pugalia
- Vejas as instruções para o Projeto 3
- E o Linux é 100% livre?
  - Vamos analisar `drivers/net/appletalk/cops_??drv.h`