

MC514

**Sistemas Operacionais:
Teoria e Prática**

**Islene Calciolari Garcia
Instituto de Computação
UNICAMP**

Conteúdo

- Introdução
- Gerência de Processos
 - Programação concorrente
- Gerência de Memória
- Gerência de Arquivos
- Entrada/Saída
- Estudo de casos

Referência principal

Teoria

Sistemas Operacionais Modernos

Andrew S. Tanenbaum

Segunda edição

Pearson - Prentice Hall

Referência principal

Prática

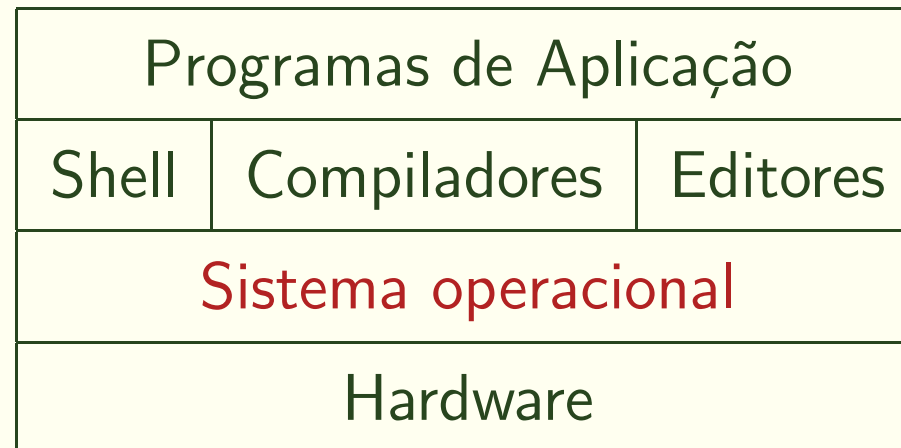
Man pages

Info Libc

Introdução

1. Sistema operacional
2. História dos sistemas operacionais
3. Revisão sobre hardware
4. Conceitos básicos
5. Chamadas aos sistema
6. Estrutura de sistemas operacionais

Sistema operacional



O sistema operacional isola o hardware das camadas superiores em um sistema computacional

Sistema operacional

- Máquina estendida

oferece uma máquina virtual mais simples de programar do que o hardware

- Gerenciador de recursos

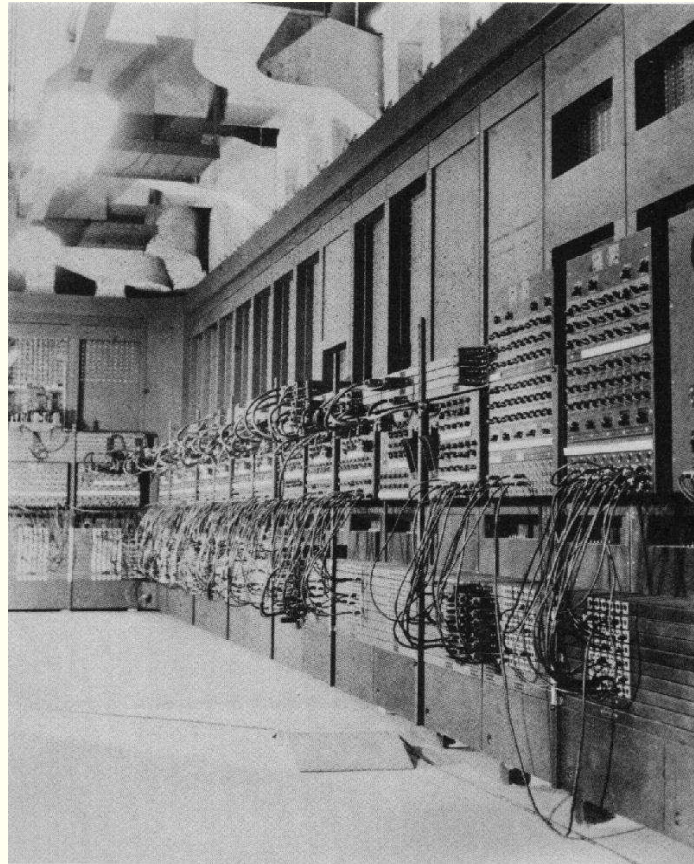
fornece uma alocação controlada de processadores, memória e dispositivos de entrada/saída

História dos sistemas operacionais

- A evolução dos sistemas operacionais está fortemente relacionada ao desenvolvimento do hardware
- Primeira geração 1945–1955
- válvulas e painéis de programação
- computação numérica: geração de tabelas de senos e cosenos.

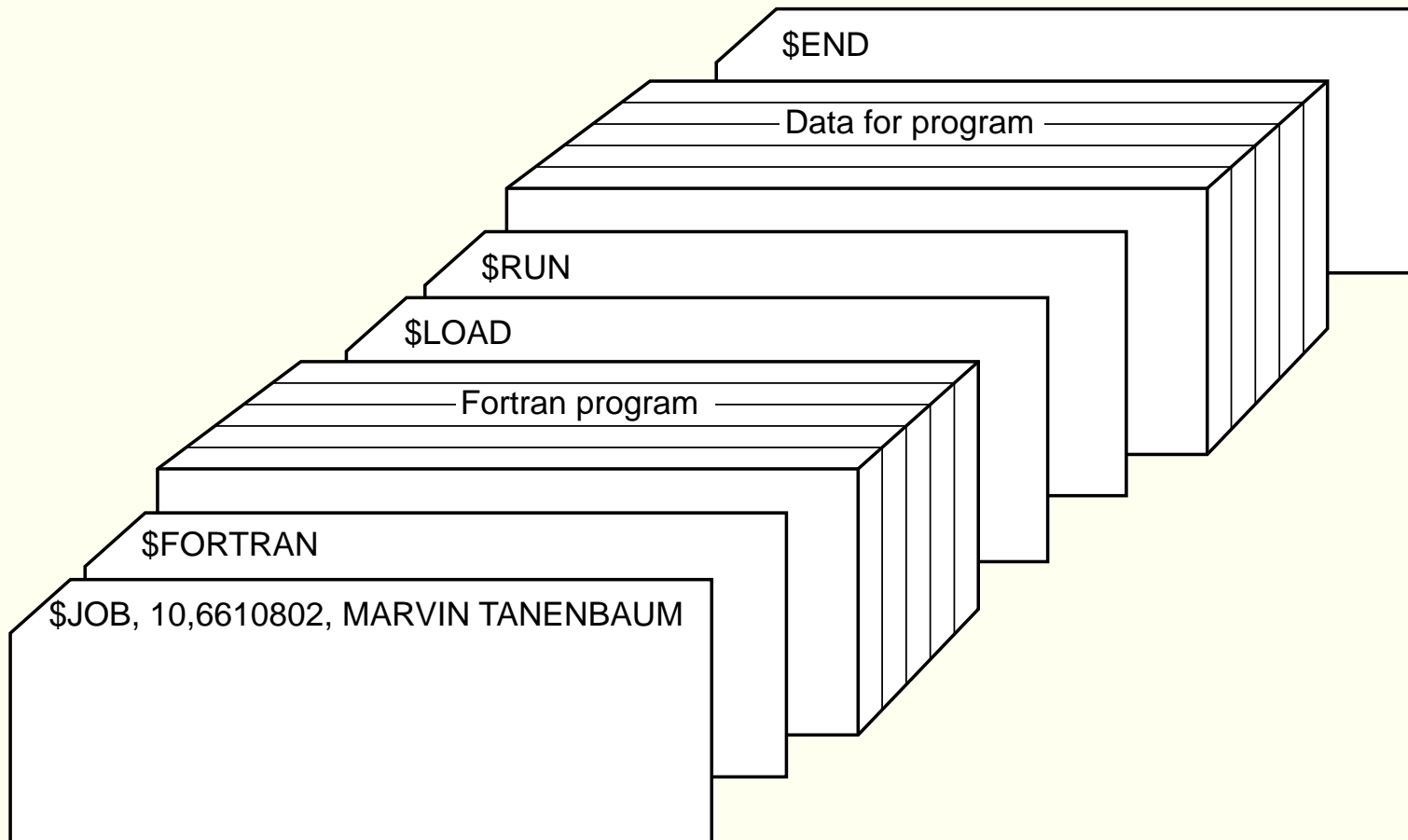
História dos sistemas operacionais

ENIAC



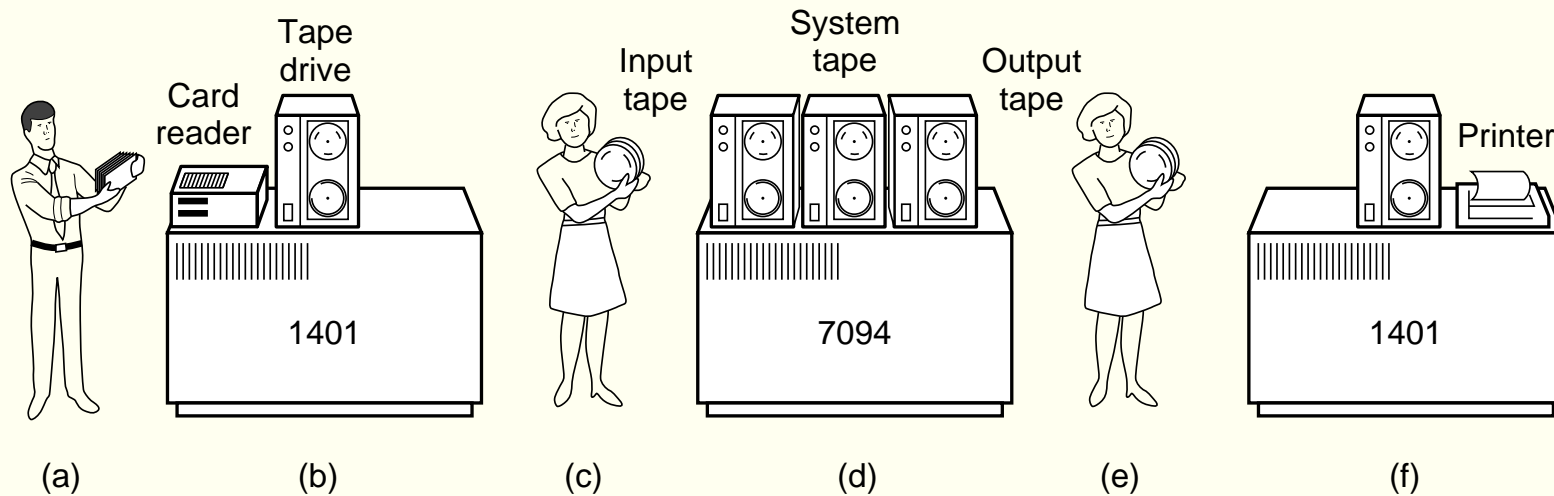
História dos sistemas operacionais

Cartões perfurados



História dos sistemas operacionais

- Segunda geração 1955–1965
- transistores e sistemas batch

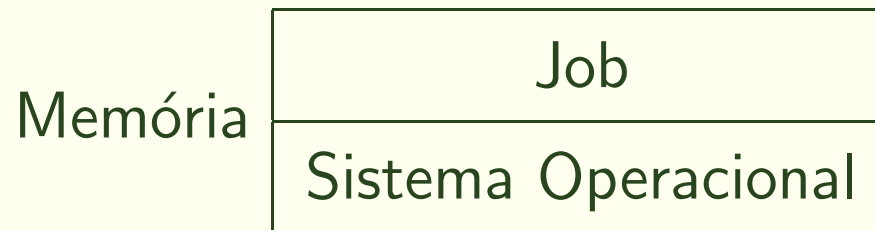


História dos sistemas operacionais

- Terceira geração 1965–1980
- circuitos integrados e multiprogramação
- System/360: família de computadores compatíveis

História dos sistemas operacionais

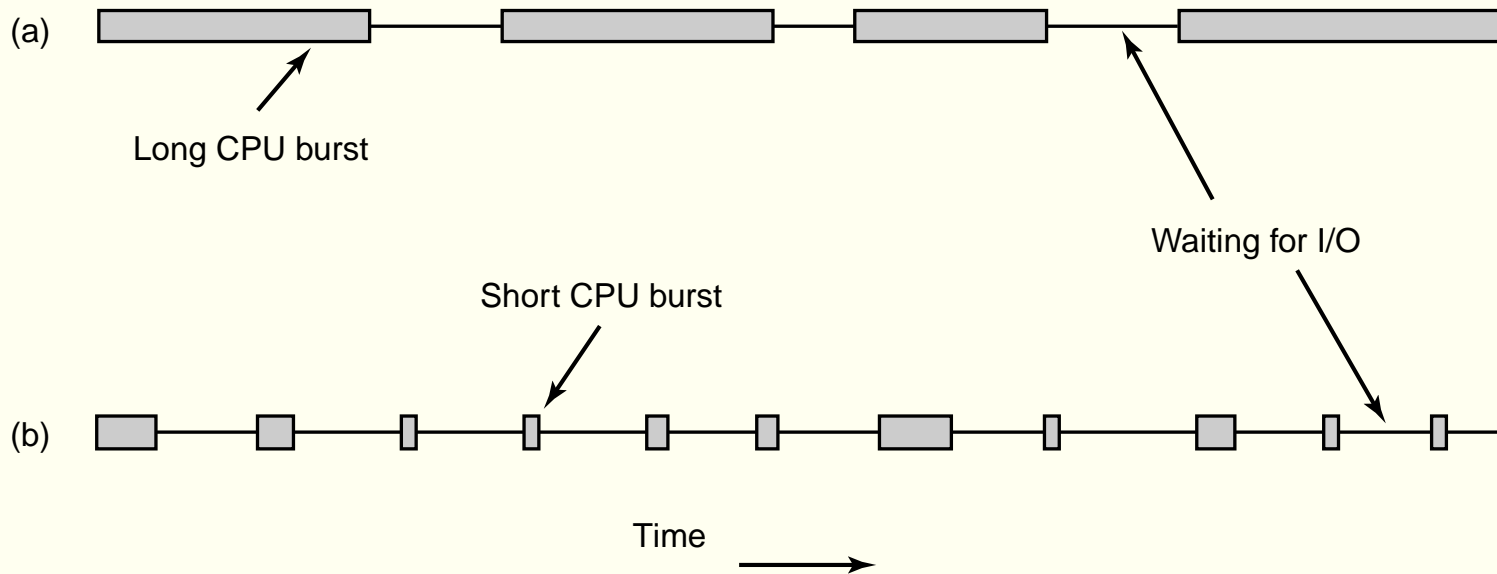
Monoprogramação



Com apenas um job em memória
a CPU fica ociosa durante operações de E/S

História dos sistemas operacionais

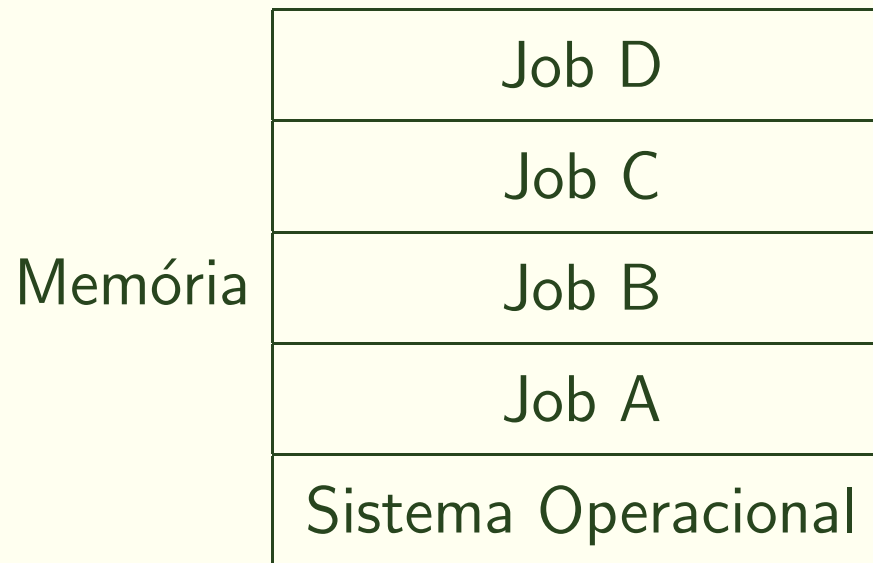
CPU-bound



IO-bound

História dos sistemas operacionais

Multiprogramação

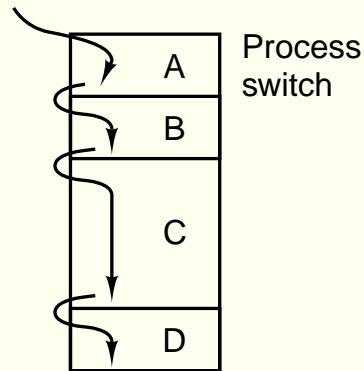


Com vários jobs em memória
a CPU pode ser melhor aproveitada

História dos sistemas operacionais

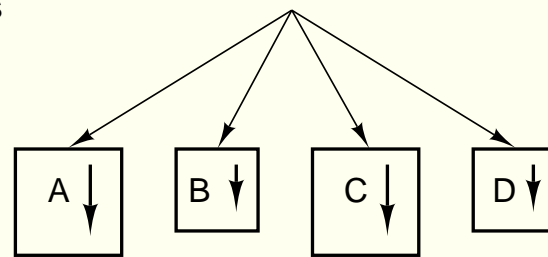
Multiprogramação

One program counter

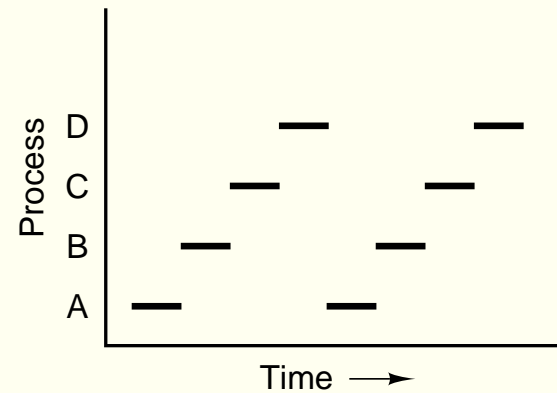


(a)

Four program counters



(b)



(c)

História dos sistemas operacionais

SPOOLing

- Simultaneous Peripheral Operation OnLine
- Leitura dos cartões passou a ser feita em paralelo à execução de outros programas
- Os computadores auxiliares puderam ser aposentados

História dos sistemas operacionais

Tempo-compartilhado



- Vários terminais conectados a um mainframe
- Os usuários exigem resposta rápida

História dos sistemas operacionais

Tempo-compartilhado

- CTSS (Compatible Time Sharing System)—desenvolvido no M.I.T., foi um sucesso comercial
- MULTICS (MULTiplexed Information and Computing Service)—projeto muito ambicioso

História dos sistemas operacionais

Mini-computadores

- Máquinas PDP-X (incompatíveis entre si)
- UNIX (inicialmente UNICS)
 - System V da AT&T
 - BSD (Berkeley Software Distribution)
 - POSIX (Portable Operating System-IX)
 - Minix
 - Linux

História dos sistemas operacionais

- Quarta geração 1980–hoje
- Circuitos integrados de larga escala
- Computadores pessoais
- CP/M (Control Program for Microcomputers)
- MS-DOS (MicroSoft Disk Operating System)

História dos sistemas operacionais GUI (Graphical User Interface)

- Macintosh
- Windows
 - Interface gráfica sobre o MS-DOS
- Unix
 - X Windows System

História dos sistemas operacionais

Diversidade atual

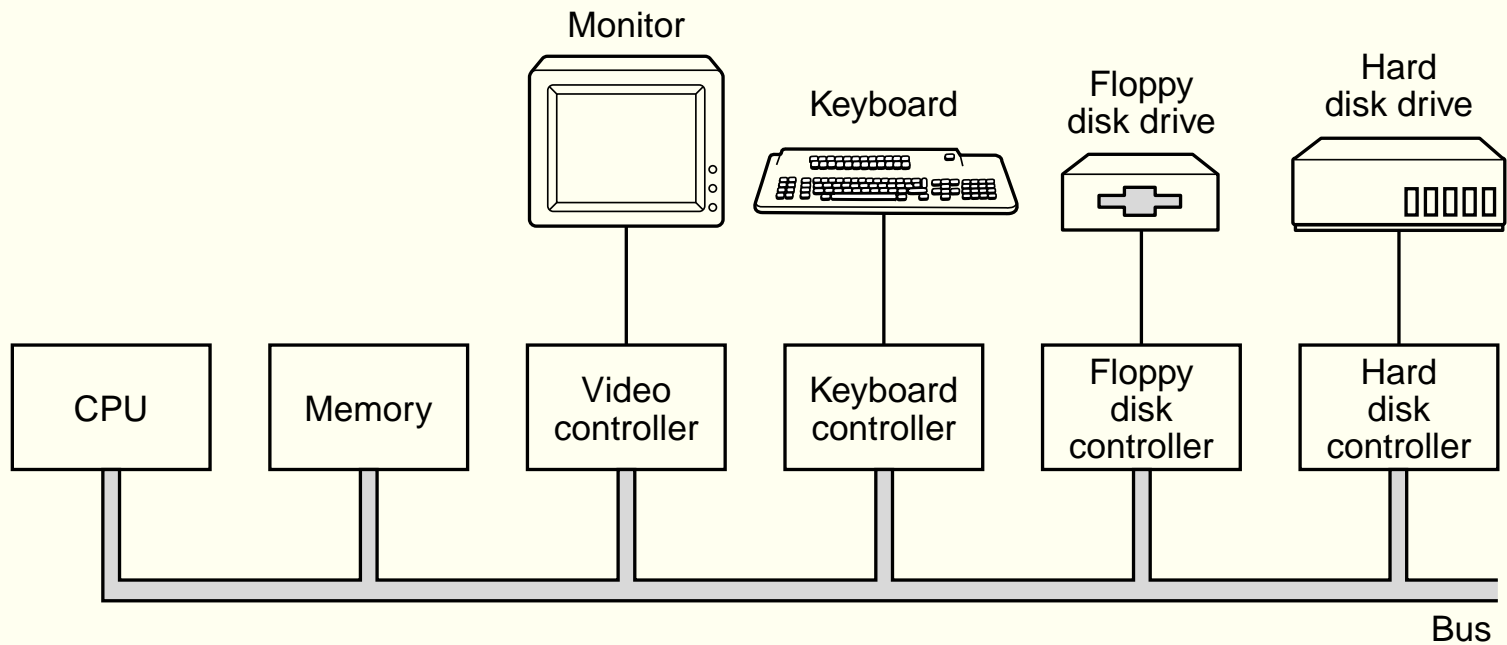
- Mainframe
- Servidores
- Multiprocessados
- Computadores pessoais
- Tempo real
- Embarcados
- Cartões inteligentes

História dos sistemas operacionais

Ciclo de desenvolvimento

| Computadores | Facilidades |
|---|---|
| Mainframe Minicomputadores Computadores pessoais Sistemas embarcados Cartões inteligentes | Linguagens de alto nível Proteção de hardware Suporte a multiprogramação Discos e sistemas de arquivos |

Revisão sobre hardware



Revisão sobre hardware

Processador

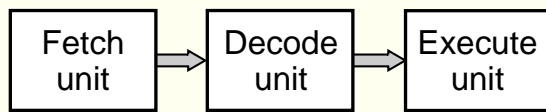
- Responsável pela execução dos programas
- Ciclo de execução: busca → decodifica → executa

PC (contador de programa) →

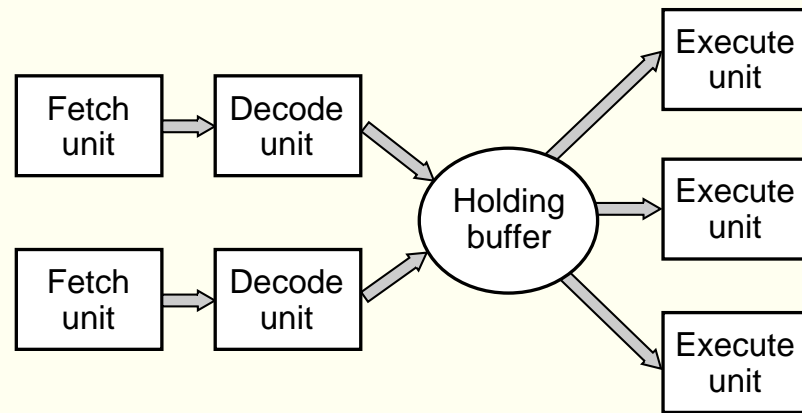
| |
|------------|
| ⋮ |
| Mov R1, A |
| Mov R2, B |
| Add R1, R2 |
| ⋮ |

Revisão sobre hardware

Pipelines e CPUs superescalares



(a)

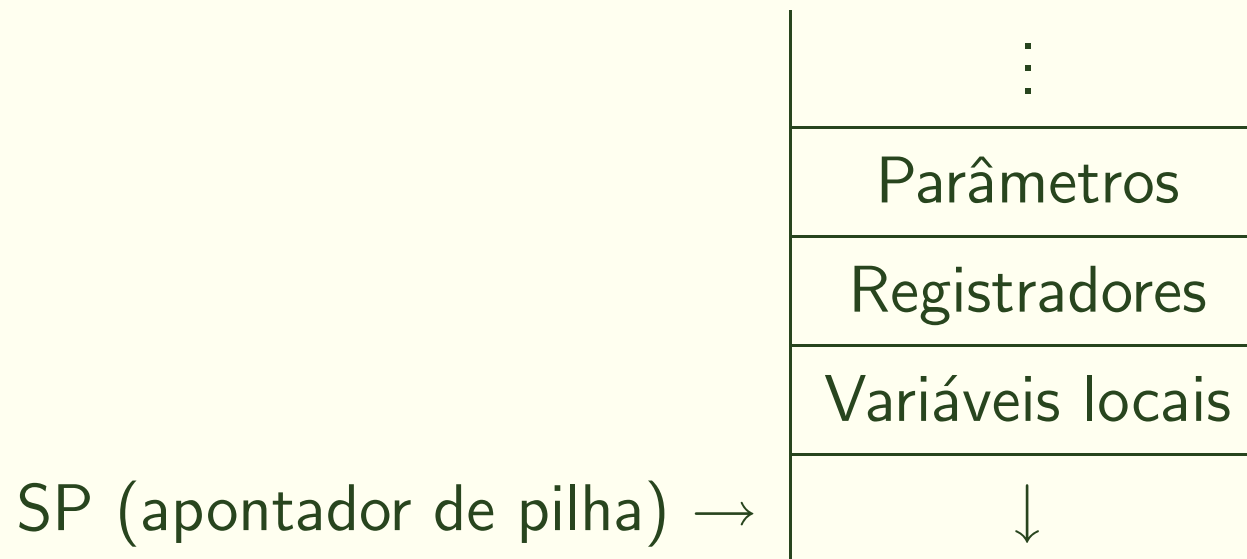


(b)

- Maior eficiência
- Maior complexidade dos compiladores e sistemas operacionais

Revisão sobre hardware

Pilha de execução



Revisão sobre hardware

Modos de execução

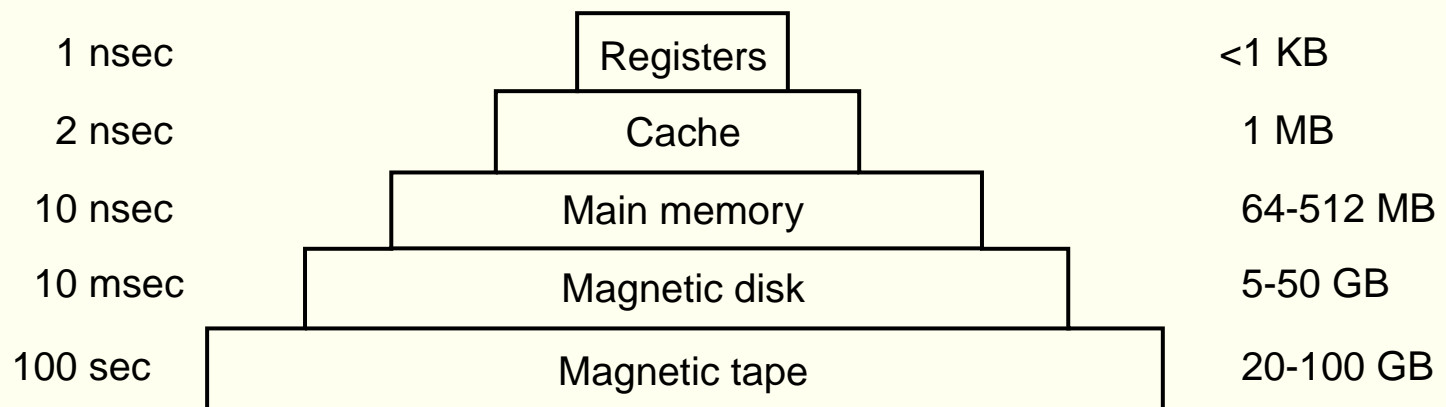
- Modo usuário
 - acesso restrito ao conjunto de instruções
- Modo núcleo
 - acesso total ao conjunto de instruções
- Chamadas ao sistema: permitem a mudança de modo de um programa

Revisão sobre hardware

Hierarquia de memória

Typical access time

Typical capacity



Revisão sobre hardware

Registradores

- Internos à CPU
- Extremamente rápidos
- Programas decidem o que deve ficar armazenado nos registradores

Revisão sobre hardware

Cache

- Internos ou muito próximos à CPU
- Divididos em linhas de cache
- Controlados por hardware
- Cache hit
- Cache miss

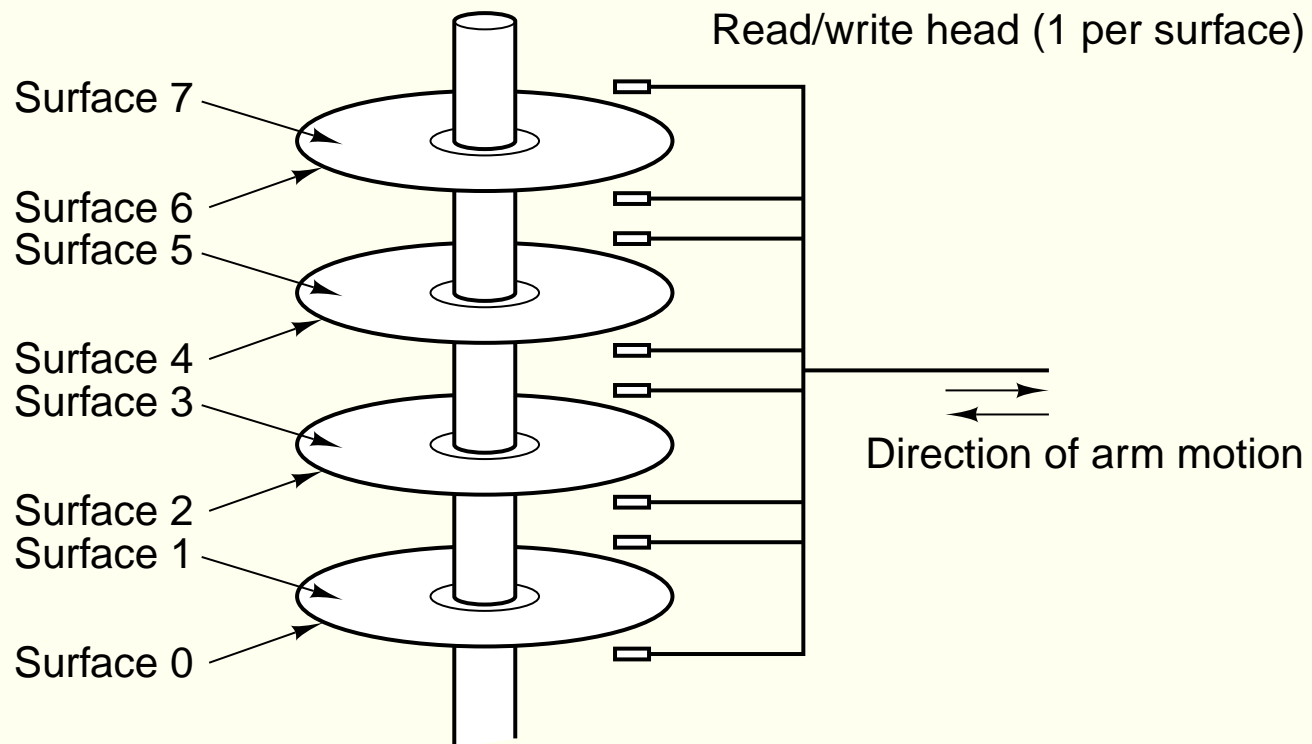
Revisão sobre hardware

Memória Principal

- Random Access Memory (RAM)
- Compromisso entre preço e desempenho
- Armazenamento volátil

Revisão sobre hardware

Disco



Revisão sobre hardware

Fitas magnéticas

- Utilizadas para cópias de segurança (backups)
- Armazenamento de grandes quantidades de dados
- Acesso seqüencial

Revisão sobre hardware

Outros tipos de memória

- ROM (Read Only Memory)
 - rápida e barata
 - bootstrap loader está gravado em ROM
- EEPROM (Electrically Erasable ROM)
 - podem ser apagadas (erros podem ser corrigidos)
- CMOS
 - dependem de uma bateria
 - armazenam relógio e configurações

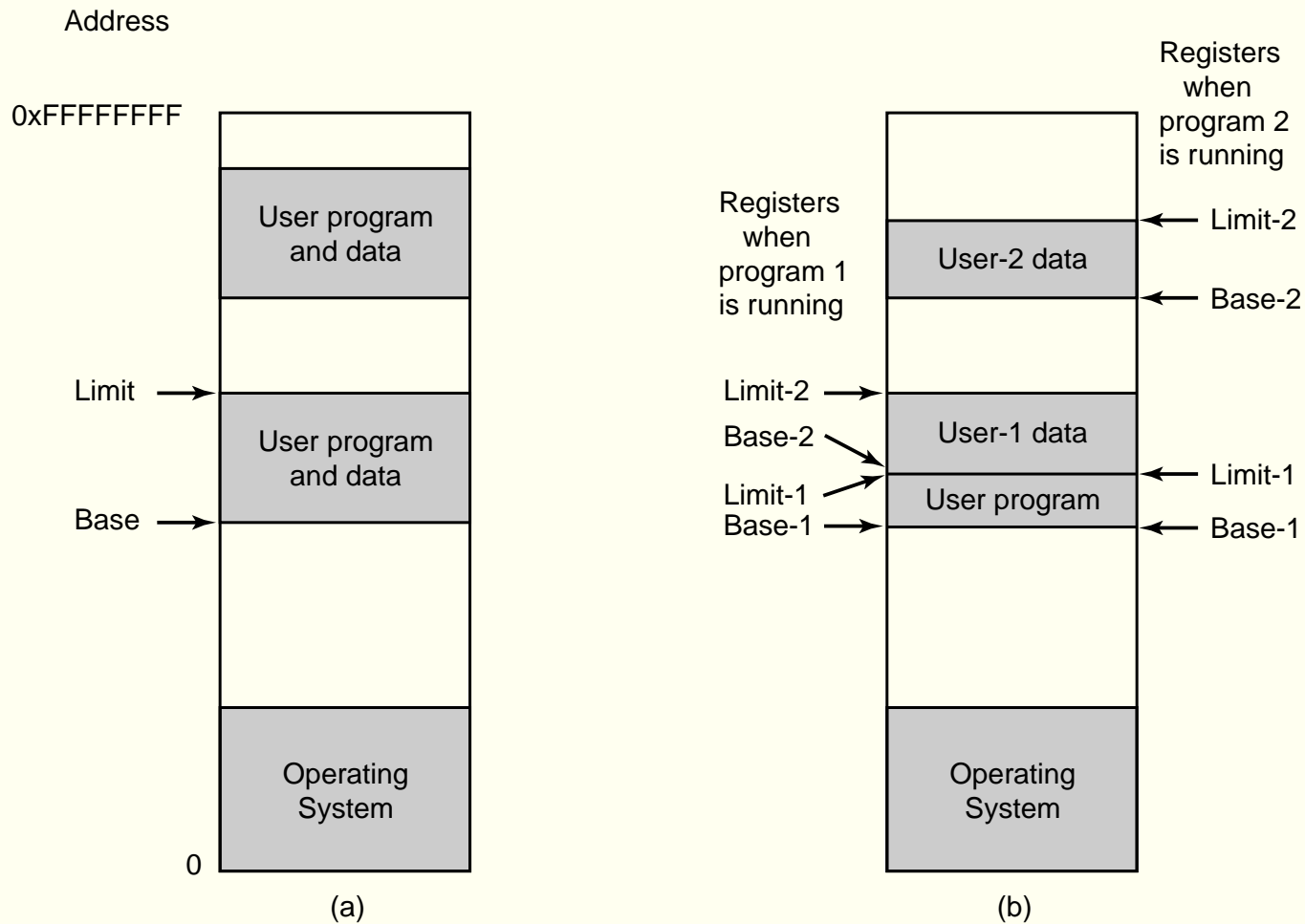
Revisão sobre hardware

Proteção e realocação

- Como proteger um programa de outro
- Como permitir que um programa seja executado em posições distintas da memória

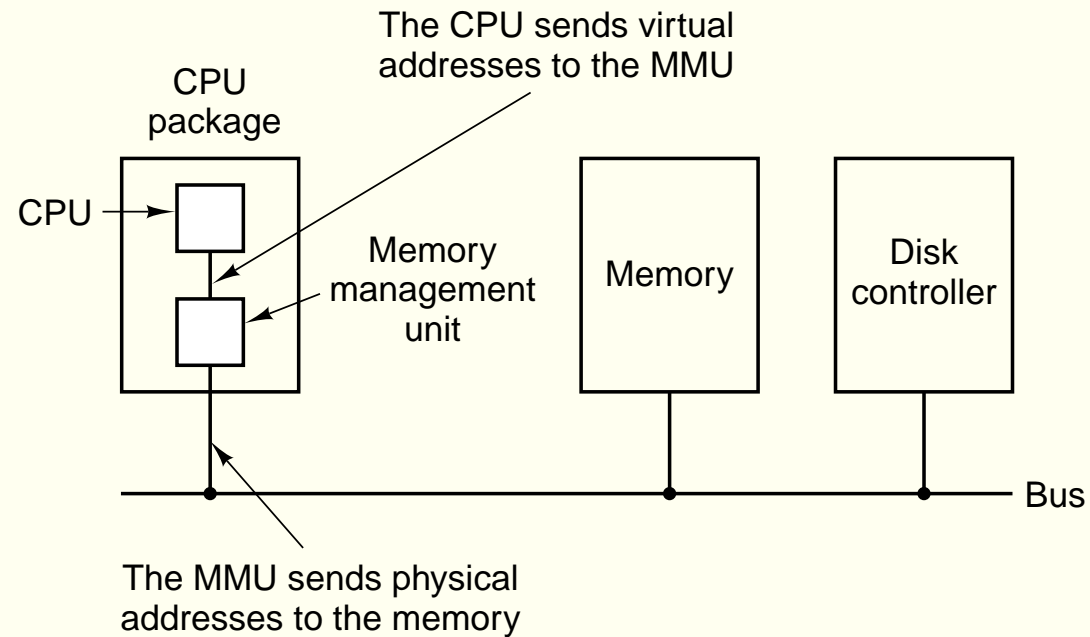
Revisão sobre hardware

Proteção e realocação



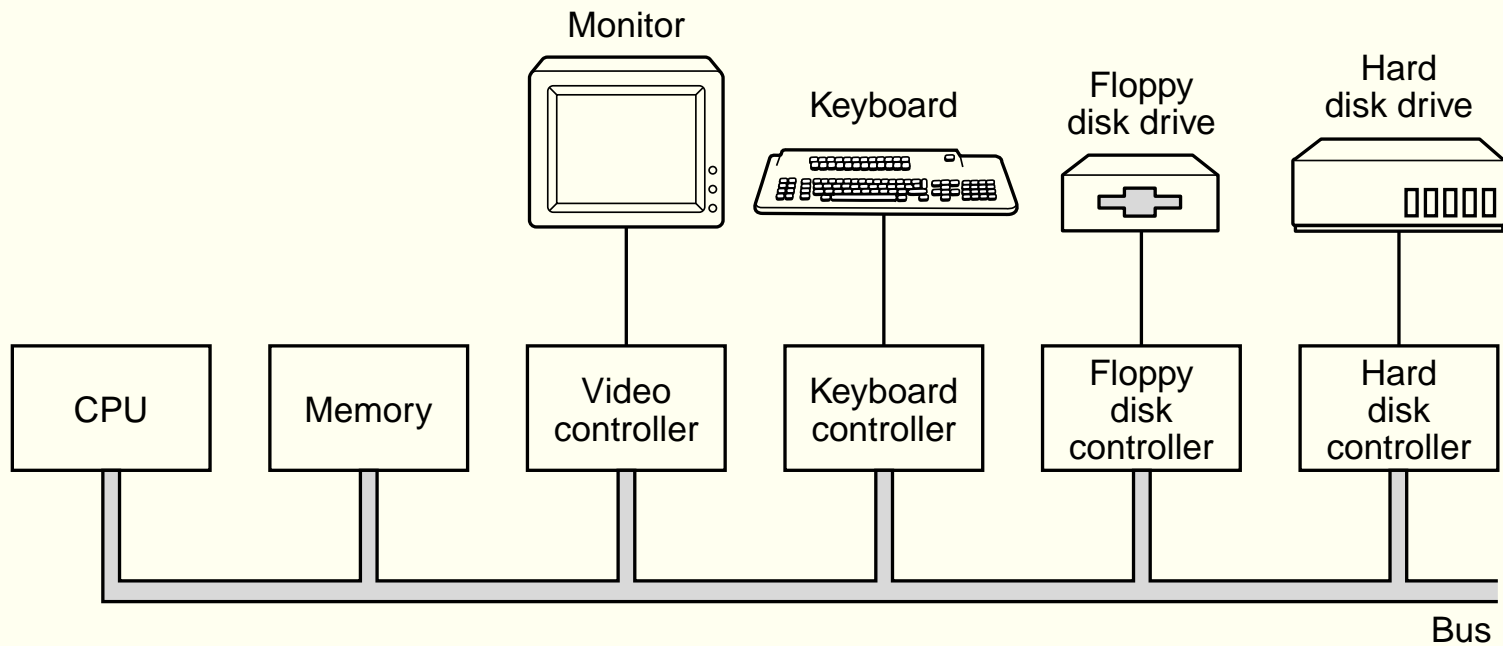
Revisão sobre hardware

Endereços físicos e virtuais



Revisão sobre hardware

Entrada e Saída



- O controlador administra fisicamente o dispositivo
- Driver de dispositivo: programa fornecido pelo fabricante que se comunica com o controlador

Revisão sobre hardware

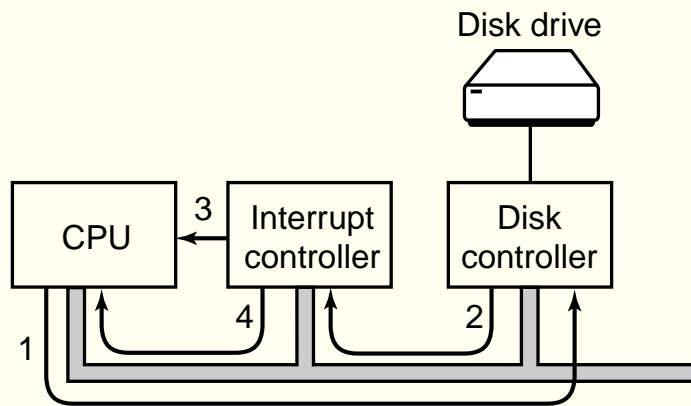
Espera ocupada

- A CPU programa um controlador para leitura
- Entra em um loop de testes consecutivos até o dado estar disponível
- Lê o dado

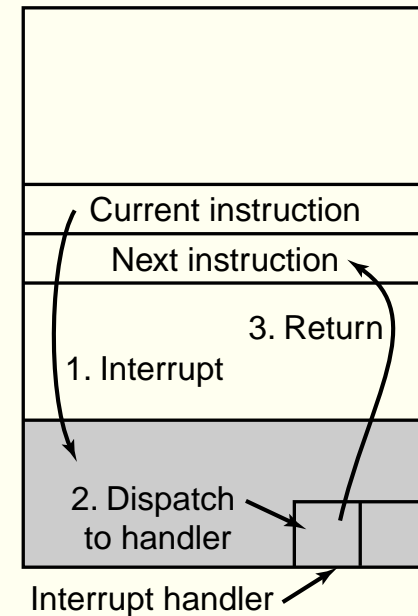
⇒ Desperdício de processamento

Revisão sobre hardware

Interrupções



(a)

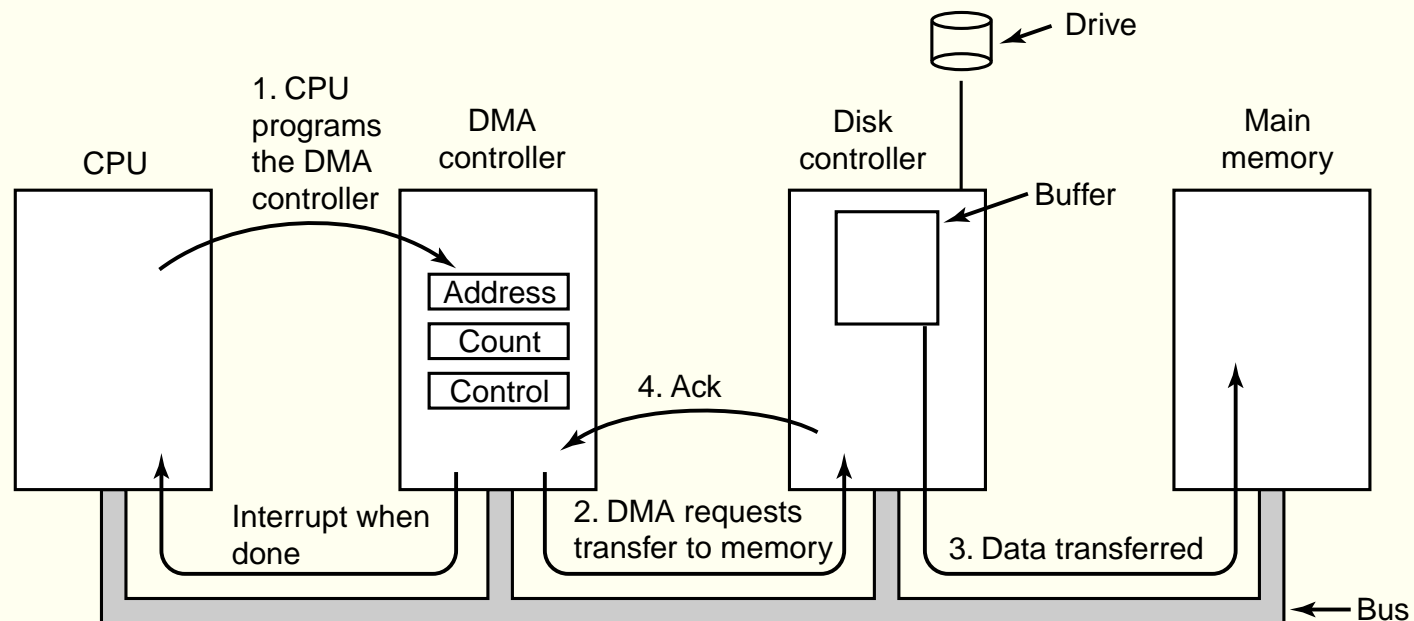


(b)

- A CPU programa o controlador e *fica livre*
- Quando o dado está disponível a CPU é interrompida

Revisão sobre hardware

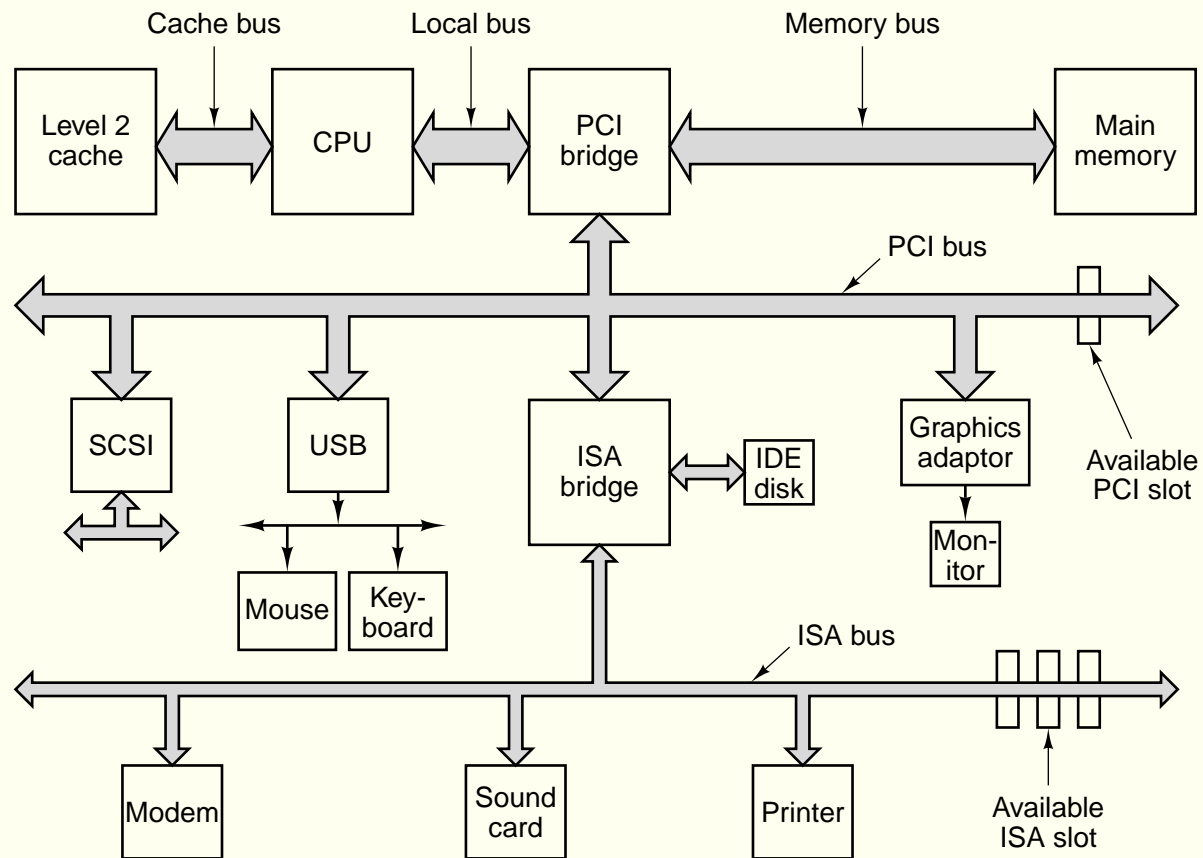
Direct Memory Access



- A CPU programa o DMA e *fica livre*
- Quando o bloco está disponível a CPU é interrompida

Revisão sobre hardware

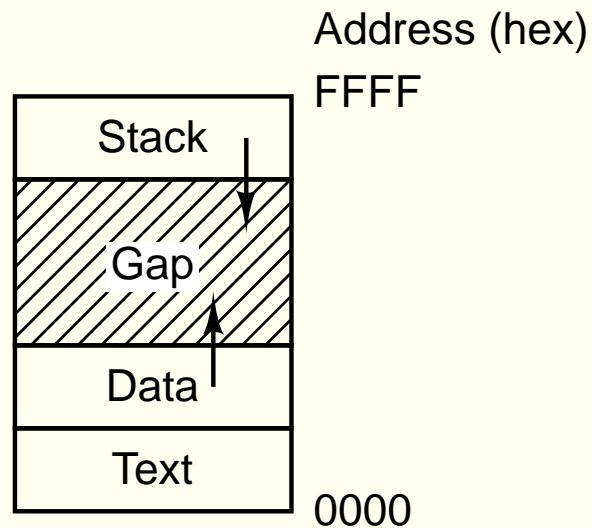
Barramentos (Pentium System)



Conceitos básicos

Processos

- Programa em execução
- Espaço de endereçamento



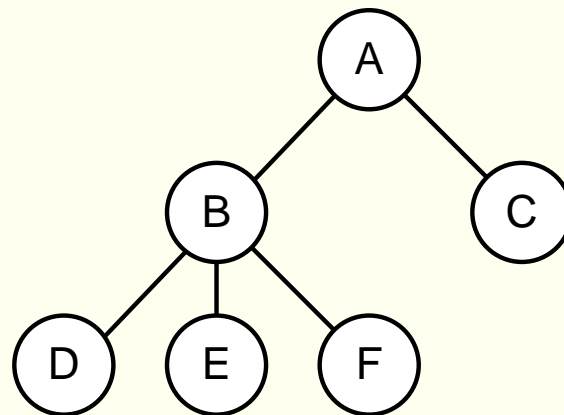
Conceitos básicos

Processos

- Registradores
 - Contador de programa
 - ponteiro de pilha,
 - PSW (program status word)
- Conjunto de arquivos abertos

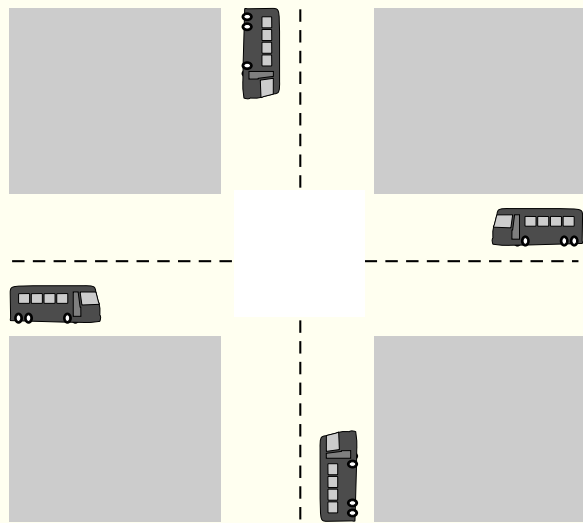
Conceitos básicos

Hierarquia de processos

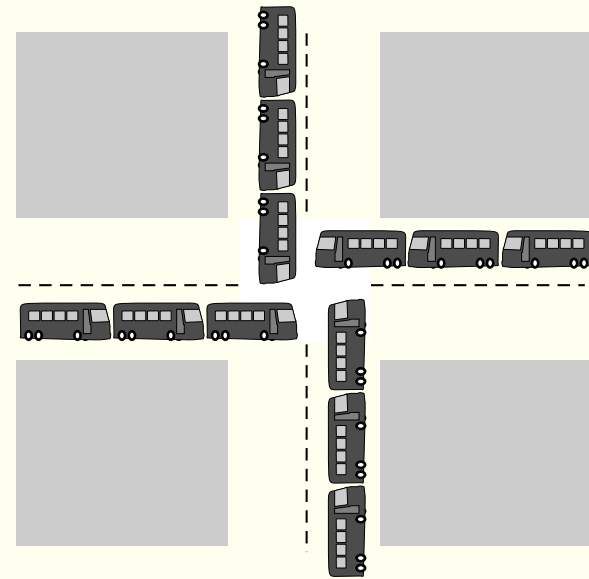


Conceitos básicos

Deadlock



(a)



(b)

Conceitos básicos

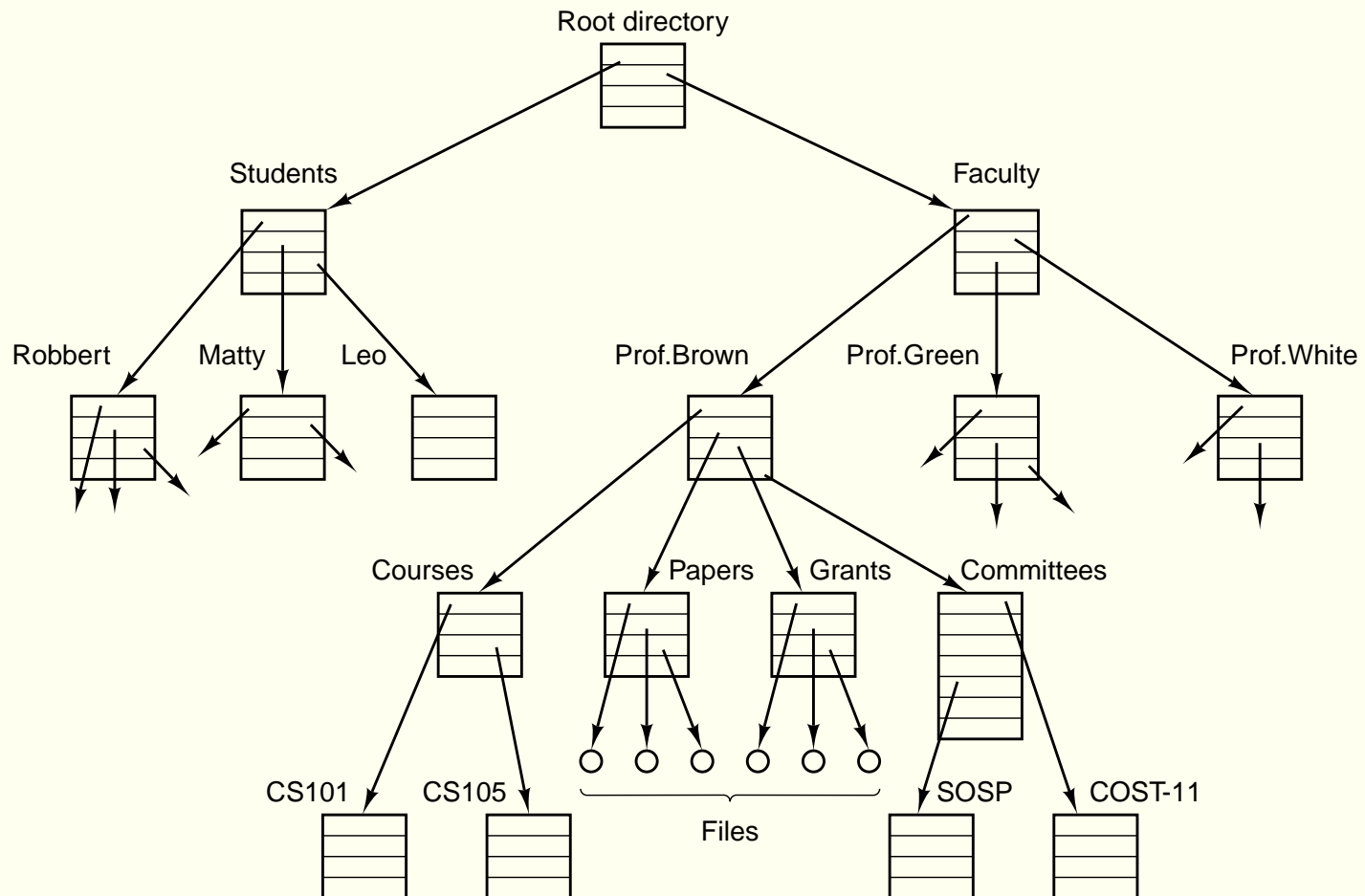
Deadlock

Suponha que dois processos querem gravar um CD-ROM a partir de dados gravados em uma fita magnética

- Processo 1
 - obtém CD-ROM
 - aguarda fita
- Process 2
 - obtém fita
 - aguarda CD-ROM

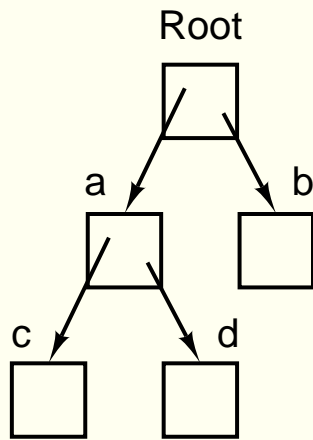
Conceitos básicos

Sistema de arquivos

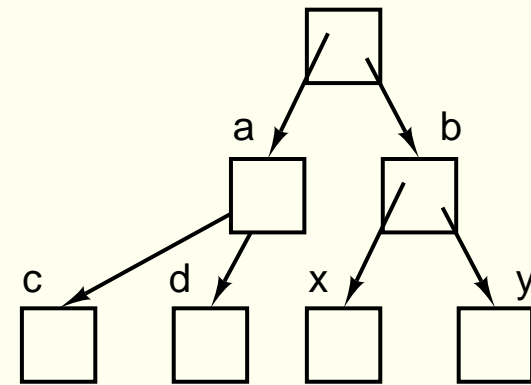
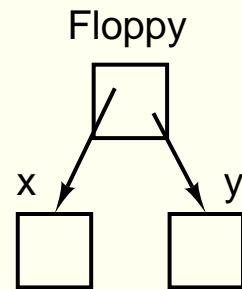


Conceitos básicos

Mount



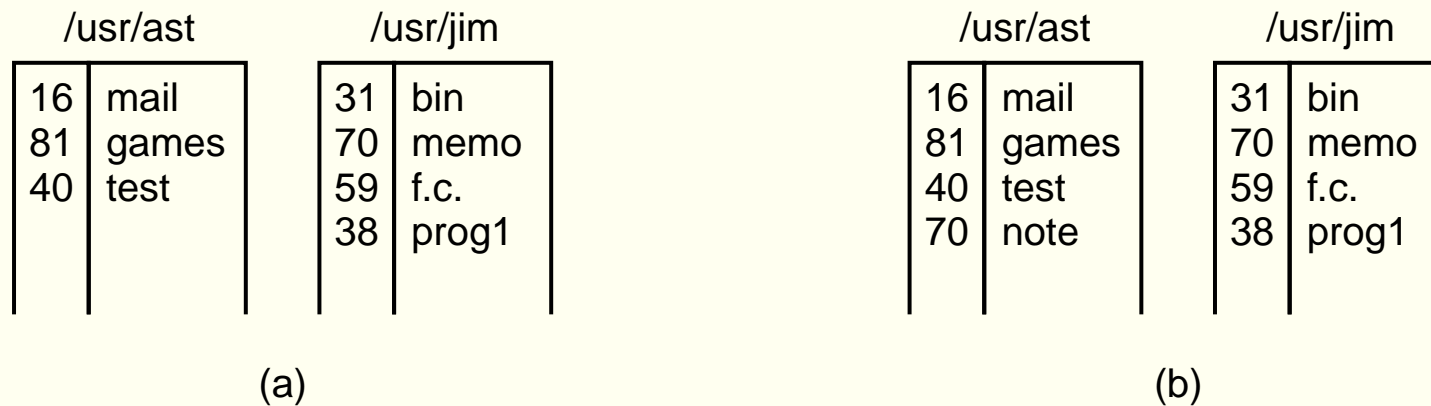
(a)



(b)

Conceitos básicos

Links



```
$ ln /usr/jim/memo /usr/ast/note
```

Ast e Jim podem compartilhar um diretório

Conceitos básicos

Segurança

```
islene@emilia% ll
```

```
-rw-r----- so.txt
```

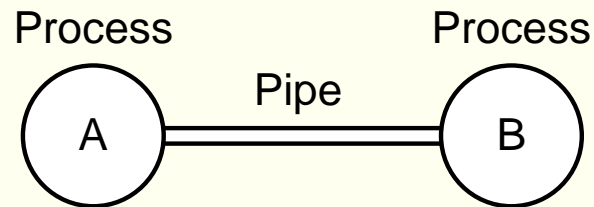
```
drwxr-xr-x threads/
```

```
islene@emilia% ll /usr/bin/passwd
```

```
-r-s--x--x /usr/bin/passwd*
```

Conceitos básicos

Pipe



```
$ grep str file | wc -l
```

Mais eficiente que:

```
$ grep str file > file-str
```

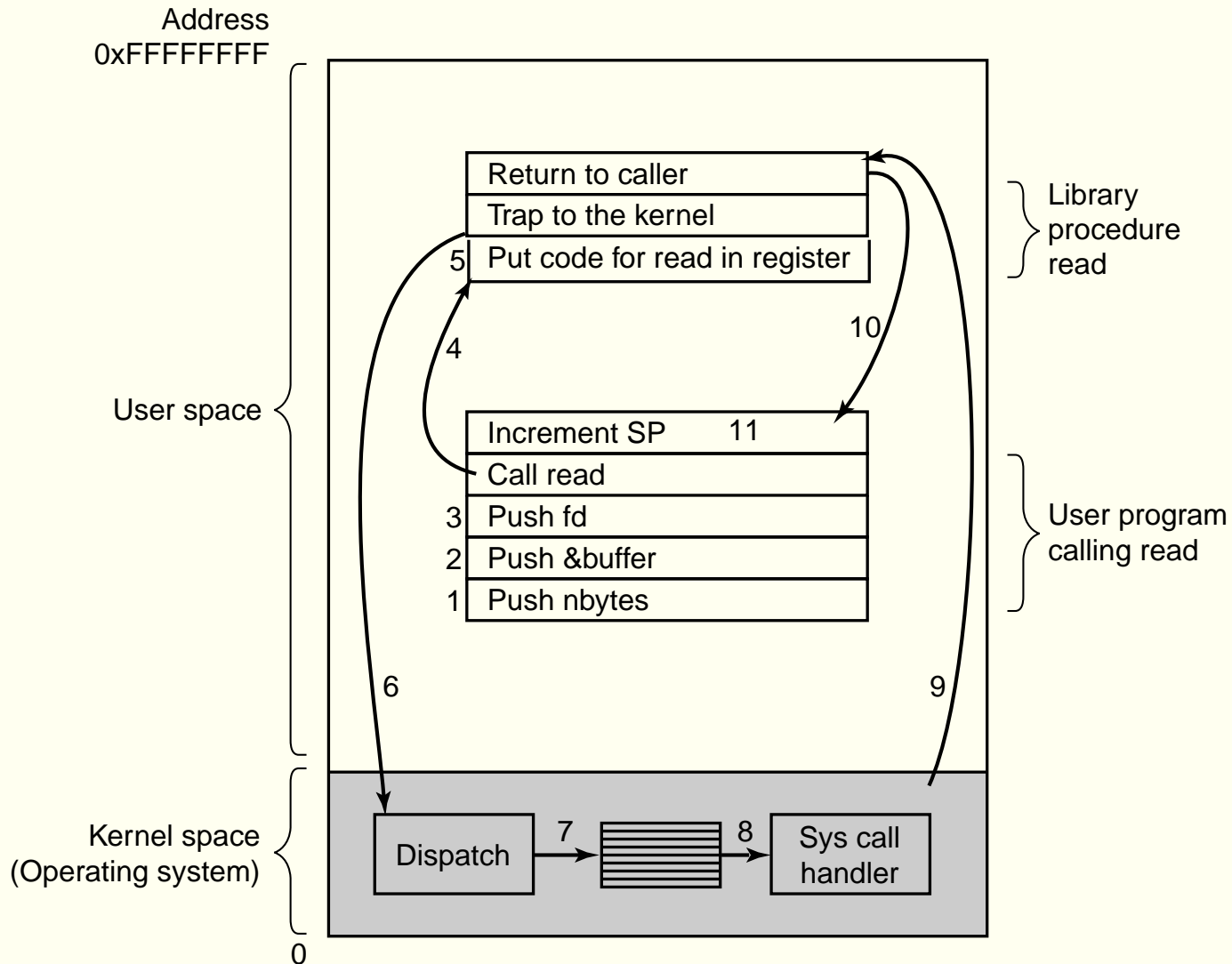
```
$ wc -l file-str
```

```
$ rm file-str
```

Chamada ao sistema

- Tipo especial de chamada de procedimento
- Passa do modo usuário para o modo supervisor
- Instrução TRAP

cont = read(fd, buffer, nbytes)



Exemplos de chamadas ao sistema

Gerência de processos

- `pid = fork()`
- `waitpid(pid, &statloc, options)`
- `s = execve(name, argv, environp)`
- `exit(status)`

Exemplos de chamadas ao sistema

Shell

```
#define TRUE 1

while (TRUE) {                               /* repeat forever */
    type_prompt( );                          /* display prompt on the screen */
    read_command(command, parameters);      /* read input from terminal */

    if (fork() != 0) {                       /* fork off child process */
        /* Parent code. */
        waitpid(-1, &status, 0);           /* wait for child to exit */
    } else {
        /* Child code. */
        execve(command, parameters, 0);    /* execute command */
    }
}
```

Exemplos de chamadas ao sistema

Gerência de arquivos

- `fd = open(file, how)`
- `s = close(fd)`
- `n = read(fd, buffer, nbytes)`
- `n = write(fd, buffer, nbytes)`
- `pos = lseek(fd, offset, whence)`
- `s = stat(name, &buf)`

Exemplos de chamadas ao sistema

Gerência de arquivos e diretórios

- `s = mkdir(name,mode)`
- `s = rmdir(name)`
- `s = link(name1,name2)`
- `s = unlink(name)`
- `s = mount(special,name,flag)`
- `s = umount(special)`

Exemplos de chamadas ao sistema

Diversas

- `s = chdir(dirname)`
- `s = chmod(name,mode)`
- `s = kill(pid,signal)`
- `seconds = time(&seconds)`

Exemplos de chamadas de sistema

| Unix | Win32 |
|---------|---------------------|
| fork | CreateProcess |
| waitpid | WaitForSingleObject |
| execve | |
| open | CreateFile |
| close | CloseHandle |

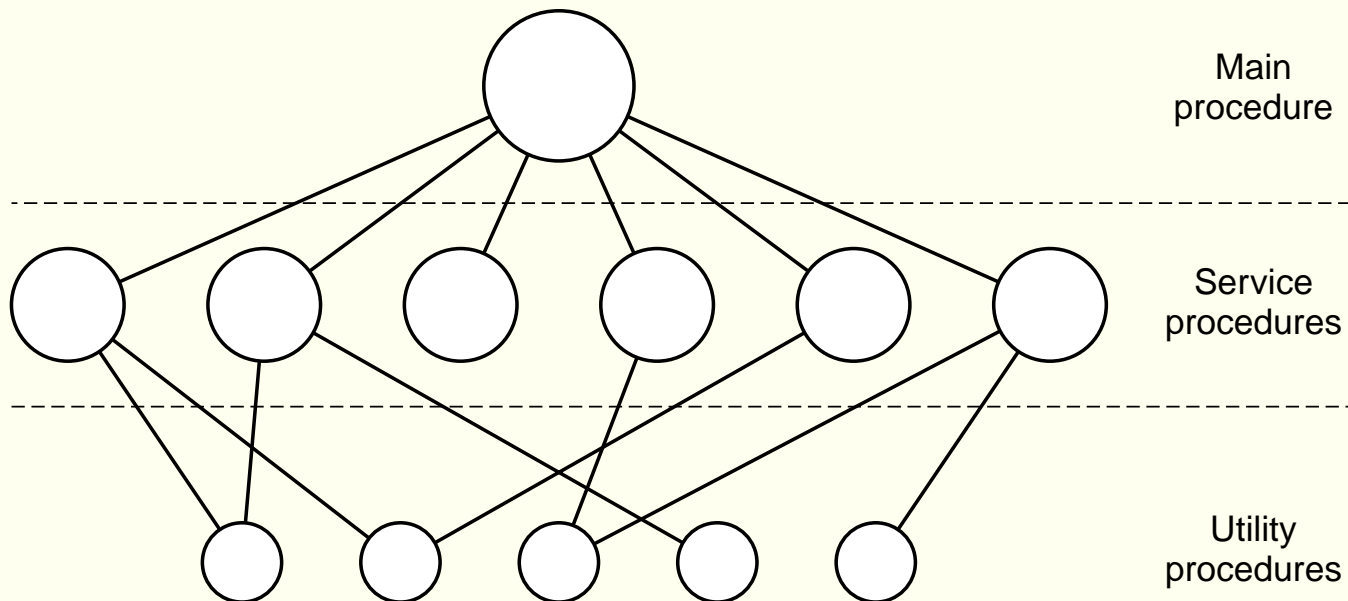
Estrutura de sistemas operacionais

Sistema monolítico

- Conjunto de procedimentos
- Visibilidade total (qualquer procedimento pode invocar outro procedimento)

Estrutura de sistemas operacionais

Sistema monolítico (estrutura simples)



Estrutura de sistemas operacionais

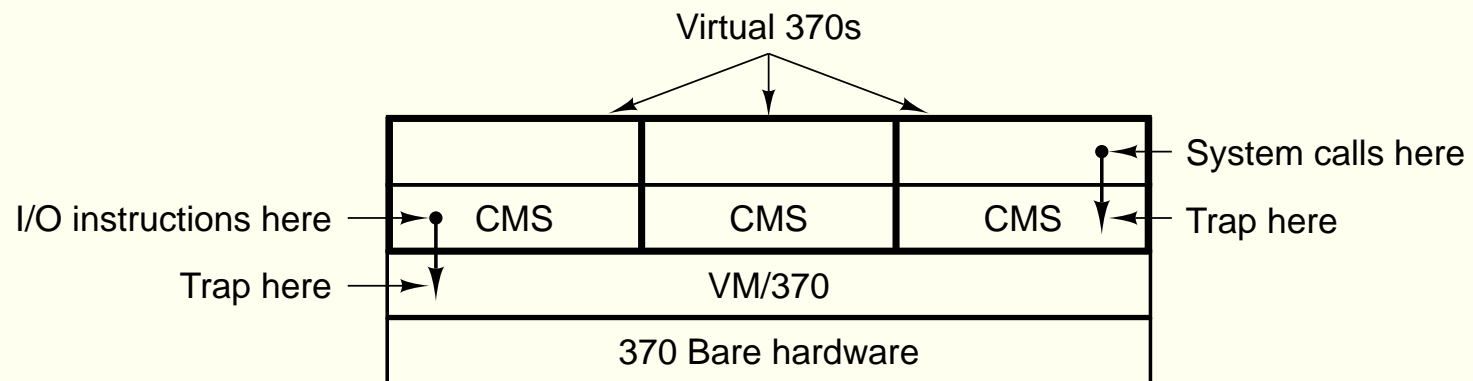
Sistema de camadas

| Layer | Function |
|--------------|---|
| 5 | The operator |
| 4 | User programs |
| 3 | Input/output management |
| 2 | Operator-process communication |
| 1 | Memory and drum management |
| 0 | Processor allocation and multiprogramming |

Estrutura de sistemas operacionais

Máquinas Virtuais

- OS/360: sistema batch
- CMS (Conversational Monitor System)



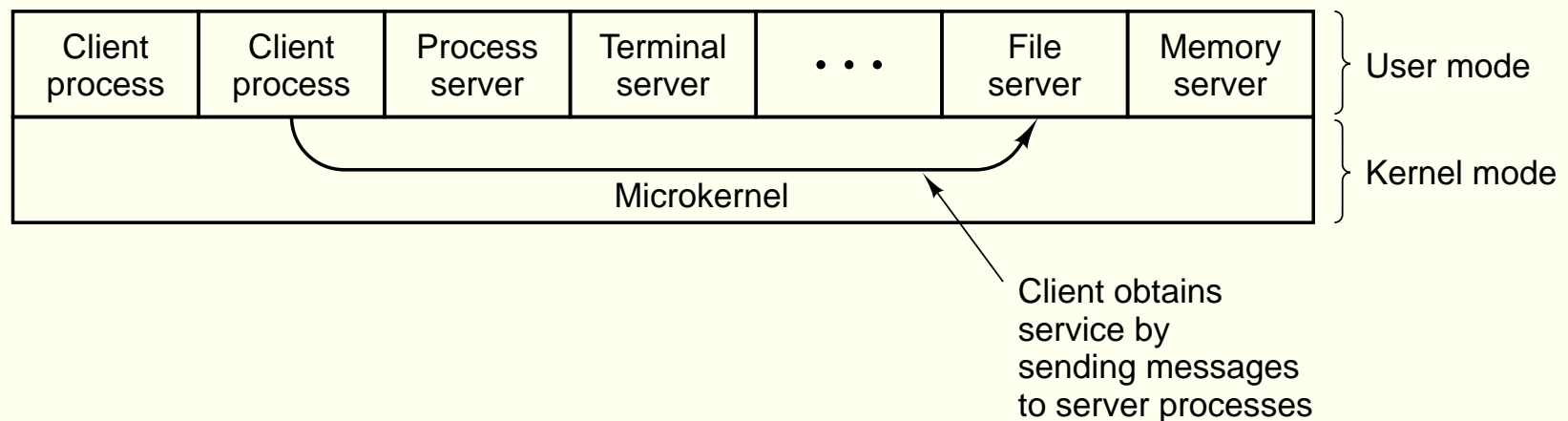
Estrutura de sistemas operacionais

Exonúcleos

- Clone de uma máquina
- Recursos alocados individualmente
- Sem transparência

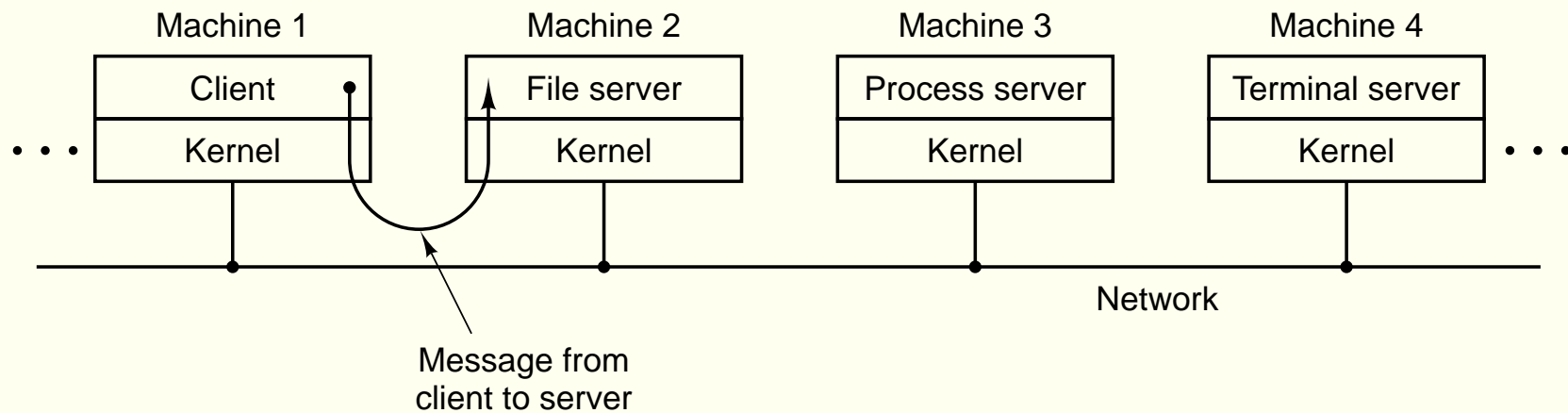
Estrutura de sistemas operacionais

Modelo cliente-servidor



Estrutura de sistemas operacionais

Modelo cliente-servidor em um ambiente distribuído



Resumo

1. Sistema operacional
2. História dos sistemas operacionais
3. Revisão sobre hardware
4. Conceitos básicos
5. Chamadas aos sistema
6. Estrutura de sistemas operacionais

Próximo tópico: Processos e threads