

## Cap. 1 – Introdução

**Prof. Marcelo Moreno**

[moreno@ice.ufjf.br](mailto:moreno@ice.ufjf.br)

<https://sites.google.com/a/ice.ufjf.br/dcc062>

## Um sistema computacional moderno

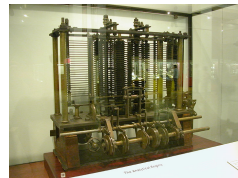
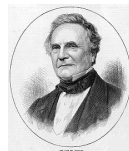
Sistema bancário	Reserva de passagens aéreas	Visualizador Web	Programas de aplicação
Compiladores	Editores	Interpretador de comandos	
Sistema operacional			Programas do sistema
Linguagem de máquina			
Microarquitetura			Hardware
Dispositivos físicos			

## Um pouco de história...

Era uma vez...	1a. Geração	2a. Geração	3a. Geração	4a. Geração
----------------	-------------	-------------	-------------	-------------

### Projeto do primeiro computador digital:

- 1837 - Máquina Analítica de C. Babbage



- Ada Lovelace colabora com Babbage

- Traduz e adiciona notas tão extensas quanto artigo.
- Explicava a sequência de passos para a geração de números de Bernoulli.



- Não havia sistema operacional

## Um pouco de história...

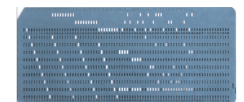
Era uma vez...	1a. Geração	2a. Geração	3a. Geração	4a. Geração
----------------	-------------	-------------	-------------	-------------

### Válvulas e painéis de programação (1945-55)

### Surgem as primeiras máquinas de calcular

- Ciclos medidos em segundos
- Relés substituídos por válvulas
- Programação feita via plugs em painel de programação
- Não havia sistema operacional

### Difundem-se os cartões perfurados




## Um pouco de história...

Era uma vez...	1a. Geração	2a. Geração	3a. Geração	4a. Geração
----------------	-------------	-------------	-------------	-------------

- Transistores e Sistemas em Lote (Batch) - 1955-65
- Mainframes
  - “Jobs” escritos em Fortran, transcritos em cartões
  - Programas demandam “tempo de transmissão”!!!
- Sistema em Lote (Batch) carregava o compilador e os jobs automaticamente a partir de fitas
- Surgem os primeiros Sistemas Operacionais
  - FMS – Fortran Monitor System
  - IBSYS (da IBM para o 7094)

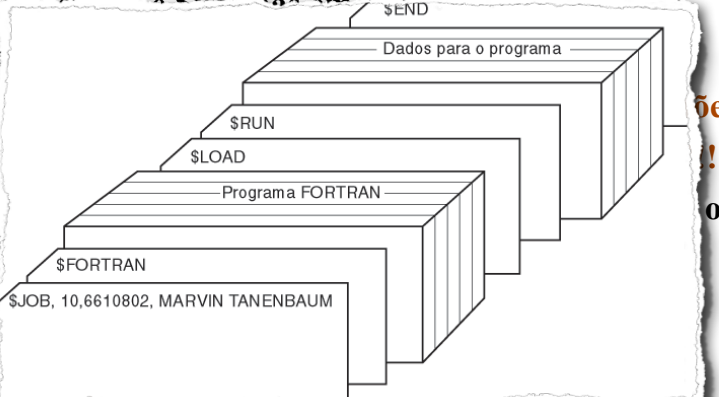
## Um pouco de história...

Era uma vez...	1a. Geração	2a. Geração	3a. Geração	4a. Geração
----------------	-------------	-------------	-------------	-------------

- Transistores e Sistemas em Lote (Batch) - 1955-65
  - Mainframes
    - “Jobs” escritos em Fortran, transcritos em cartões
- 
- de fita magnética  
Leitora de cartões  
Fita de entrada  
Fita do sistema  
Fita de saída  
Impressora
- 1401 7094 1401
- FMS – Fortran Monitor System
- IBSYS (da IBM para o 7094)

## Um pouco de história...

Era uma vez...	1a. Geração	2a. Geração	3a. Geração	4a. Geração
----------------	-------------	-------------	-------------	-------------

- Transistores e Sistemas em Lote (Batch) - 1955-65
  - Mainframes
    - “Jobs” escritos em Fortran, transcritos em cartões
    - Programas demandam “tempo de transmissão”!!!
  - Sistema em Lote (Batch) carregava o compilador e os jobs automaticamente a partir de fitas
  - Surgem os primeiros Sistemas Operacionais
    - FMS – Fortran Monitor System
    - IBSYS (da IBM para o 7094)
- 
- \$END  
Dados para o programa  
\$RUN  
\$LOAD  
Programa FORTRAN  
\$FORTRAN  
\$JOB, 10,6610802, MARVIN TANENBAUM

## Um pouco de história...

Era uma vez...	1a. Geração	2a. Geração	3a. Geração	4a. Geração
----------------	-------------	-------------	-------------	-------------

- Circuitos Integrados e Multiprogramação (1965-80)
- Grande número de computadores incompatíveis
- Manter várias linhas de produção gerava grande custo
  - IBM lança série System/360. Softwares compatíveis
  - O sistema operacional para a série era o OS/360
- Desempenho limitado levou à concepção de técnicas de multiprogramação.
  - Timesharing; Spooling
- Criação do Unix

## Um pouco de história...

Era uma vez...	1a. Geração	2a. Geração	3a. Geração	4a. Geração
----------------	-------------	-------------	-------------	-------------

- **Computadores Pessoais (1980-presente)**
- **Lançamento do processador Intel 8080 em 1974**
  - **Microcomputador com disco flexível. Sist. Operacional CP/M**
- **IBM lança padrão IBM PC. Parceria com Bill Gates**
- **Sistemas operacionais com shell linha de comando**
- **POSIX, Minix, Linux**
- **Concepção da Interface Gráfica do Usuário (GUI)**
  - **Xerox PARC concebe mas não vê importância. Jobs vê.**
  - **Apple Macintosh. Unix X-Window. Microsoft Windows**



## Um pouco de história...

Era uma vez...	1a. Geração	2a. Geração	3a. Geração	4a. Geração
----------------	-------------	-------------	-------------	-------------

- **Sistemas Operacionais de Rede**
- **Sistemas Operacionais Distribuídos**
- **Sistemas Operacionais para Mainframes**
- **Sistemas Operacionais para Servidores**
- **Sistemas Operacionais para Multiprocessamento**
- **Sistemas Operacionais para Computadores Pessoais**
- **Sistemas Operacionais para Dispositivos Móveis**
- **Sistemas Operacionais Embarcados**
- **Sistemas Operacionais de Tempo Real**



## O que é um sistema operacional?

- **Software responsável por gerenciar os componentes de um sistema computacional e fornecer aos programas uma interface com o hardware mais simples.**
- **O que é S.O. e o que não é S.O.?**
  - **Execução em modo usuário**
  - **Execução em modo núcleo**
- **Delimitação difícil porque S.O. têm duas funções:**
  - **Estender a máquina para os programas**
  - **Gerenciar os recursos da máquina**

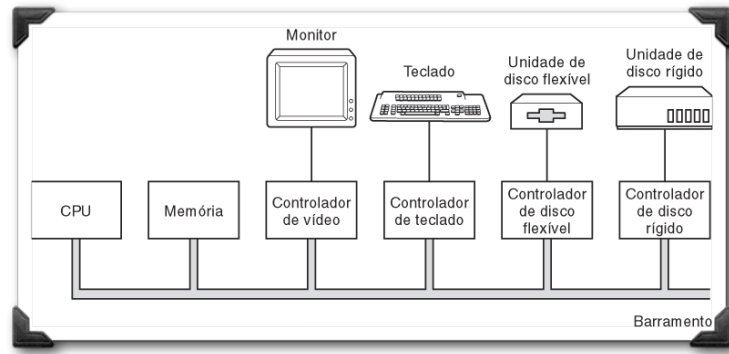


## Funções do S.O.

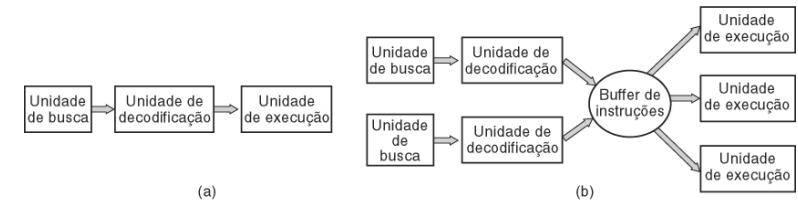
- **S.O. como uma máquina estendida**
  - **Ocultar os detalhes complicados que têm que ser executados**
  - **Apresentar ao usuário uma máquina “virtual”, mais fácil de usar**
- **S.O. como um gerenciador de recursos**
  - **Cada programa tem um tempo com o recurso**
  - **Cada programa tem um espaço no recurso**



## Revisão sobre hardware de computadores

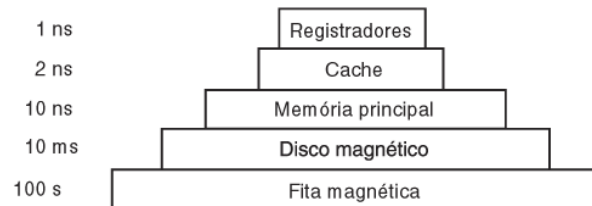


## Revisão sobre hardware de computadores

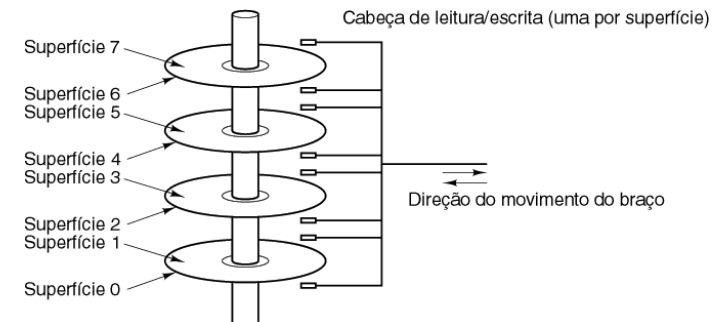


## Revisão sobre hardware de computadores

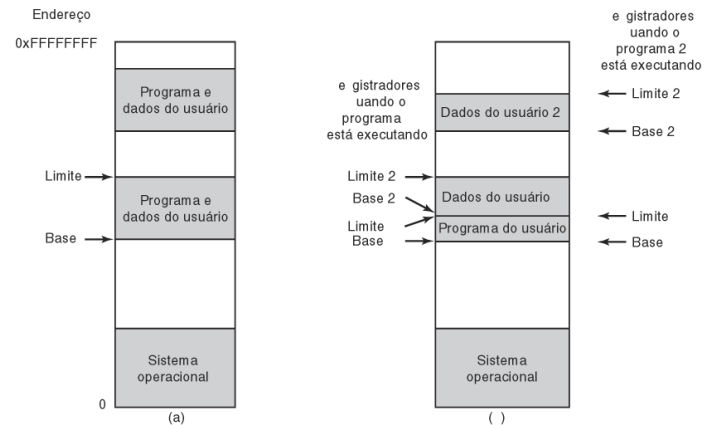
Tempo de acesso típico



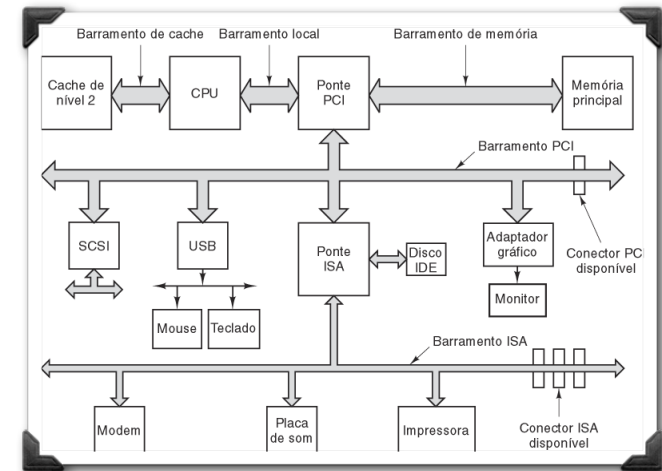
## Revisão sobre hardware de computadores



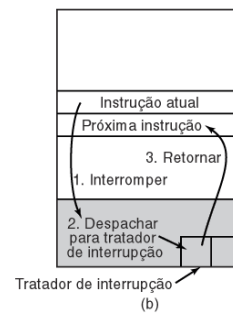
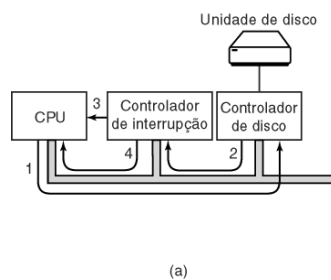
## Revisão sobre hardware de computadores



## Revisão sobre hardware de computadores



## Revisão sobre hardware de computadores

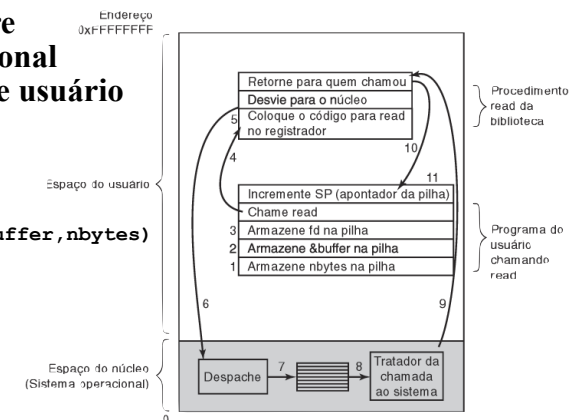


## Chamadas de Sistema

- É a interface entre o sistema operacional e os programas de usuário

- Exemplo: `read()`

`counter = read(fd,buffer,nbytes)`



## Chamadas de Sistema POSIX

Gerenciamento de processos

Chamada	Descrição
pid = fork( )	Crie um processo filho idêntico ao processo pai
pid = waitpid(pid, &statloc, options)	Aguarde um processo filho terminar
s = execve(name, argv, environp)	Substitua o espaço de endereçamento do processo
exit(status)	Termine a execução do processo e retorne o estado

```
#define TRUE 1
```

```
while (TRUE) {
    type_prompt( );           /* repita para sempre */
    read_command(command, parameters); /* mostra prompt na tela */
                                /* lê entrada do terminal */

    if (fork( ) !=0) {         /* cria processo filho */
        /* Parent code. */
        waitpid(-1, *status, 0); /* aguarda o processo filho acabar */
    } else {
        /* Child code. */
        execve(command, parameters, 0); /*executa o comando */
    }
}
```



## Chamadas de Sistema POSIX

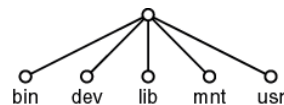
Gerenciamento de arquivos

Chamada	Descrição
fd = open(file, how, ...)	Abra um arquivo para leitura, escrita ou ambas
s = close(fd)	Feche um arquivo aberto
n = read(fd, buffer, nbytes)	Leia dados de um arquivo para um buffer
n = write(fd, buffer, nbytes)	Escreva dados de um buffer para um arquivo
position = lseek(fd, offset, whence)	Mova o ponteiro de posição do arquivo
s = stat(name, &buf)	Obtenha a informação de estado do arquivo

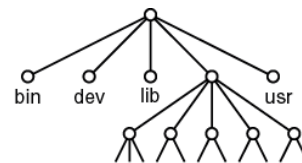
## Chamadas de Sistema POSIX

Gerenciamento do sistema de diretório e arquivo

Chamada	Descrição
s = mkdir(name, mode)	Crie um novo diretório
s = rmdir(name)	Remova um diretório vazio
s = link(name1, name2)	Crie uma nova entrada, name2, apontando para name1
s = unlink(name)	Remova uma entrada de diretório
s = mount(special, name, flag)	Monte um sistema de arquivo
s = umount(special)	Desmonte um sistema de arquivo



(a)



(b)

## Chamadas de Sistema POSIX

Diversas

Chamada	Descrição
s = chdir(dirname)	Altere o diretório de trabalho
s = chmod(name, mode)	Altere os bits de proteção do arquivo
s = kill(pid, signal)	Envie um sinal a um processo
seconds = time(&seconds)	Obtenha o tempo decorrido desde 1º de janeiro de 1970



# Chamadas de Sistema Win32

Unix	Win32	Descrição
fork	CreateProcess	Crie um novo processo
waitpid	WaitForSingleObject	Pode esperar um processo sair
execve	(none)	CrieProcesso = fork + execve
exit	ExitProcess	Termine a execução
open	CreateFile	Crie um arquivo ou abra um arquivo existente
close	CloseHandle	Feche um arquivo
read	ReadFile	Leia dados de um arquivo
write	WriteFile	Escreva dados para um arquivo
lseek	SetFilePointer	Mova o ponteiro de posição do arquivo
stat	GetFileAttributesEx	Obtenha os atributos do arquivo
mkdir	CreateDirectory	Crie um novo diretório
rm dir	RemoveDirectory	Remova um diretório vazio
link	(none)	Win32 não suporta ligações (link)
unlink	DeleteFile	Destrua um arquivo existente
mount	(none)	Win32 não suporta mount
umount	(none)	Win32 não suporta mount
chdir	SetCurrentDirectory	Altere o diretório de trabalho atual
chmod	(none)	Win32 não suporta segurança (embora NT suporte)
kill	(none)	Win32 não suporta sinais
time	GetLocalTime	Obtenha o horário atual