

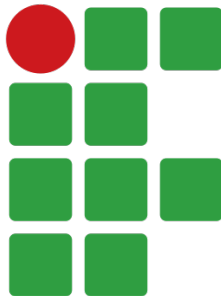
SISTEMAS OPERACIONAIS

AULA 02: ESTRUTURA DO SISTEMA OPERACIONAL

10 de outubro de 2024

Prof. Me. José Paulo Lima

IFPE Garanhuns



**INSTITUTO
FEDERAL**
Pernambuco

SUMÁRIO I



Tipos de Sistemas Operacionais

Monotarefa x Multitarefa

Monousuário x Multiusuário

Monoprocessados x Multiprocessados

Classes de Sistemas Operacionais

SOs de Grande Porte

SOs de Servidores

SOs de Computadores Pessoais

SOs de Computadores Portáteis

SOs de Embarcados

SOs de Tempo Real

SOs de Tempo Real Crítico

SUMÁRIO II



Atribuições do Sistemas Operacionais

- Gerenciamento de *hardware*

 - CPU

 - Memórias

- Gerenciamento de processos

- Gerenciamento de memória

- Gerenciamento de arquivos

- Gerenciamento de dispositivos

- Gerenciamento de proteção

 - Chamadas de sistema

SUMÁRIO III



Estrutura de Sistemas Operacionais

- Sistemas Monolíticos

- Sistema de camadas

- Micronúcleo

- Modelo Cliente-Servidor

- Máquinas virtuais

 - Máquinas virtuais Java

Referências

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

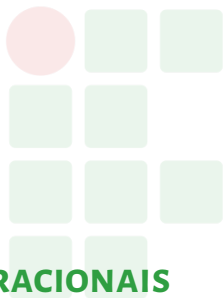


- ▶ Gerenciamento de memória:
 - ▶ Fornecer para as tarefas (processos) um espaço de memória;
 - ▶ Isolado e independente;
 - ▶ Podem utilizar o disco como memória virtual.
- ▶ Gerenciamento de processos:
 - ▶ Distribuir a capacidade de processamento;
 - ▶ Sincronização de processos.
- ▶ Gerenciamento de dispositivos
 - ▶ Controla a E/S (Pen-drives, discos de armazenamento, DVD, etc.);
 - ▶ Controladores.
- ▶ Gerenciamento de arquivos:
 - ▶ Interage com o gerenciamento de dispositivos;
 - ▶ Abstração de arquivos e diretórios.
- ▶ Gerenciamento de proteção:
 - ▶ Políticas de acesso aos recursos;
 - ▶ Um processo só pode acessar os recursos que tem permissão;
 - ▶ Controle de usuários e grupos.

EVOLUÇÃO DOS SISTEMAS OPERACIONAIS



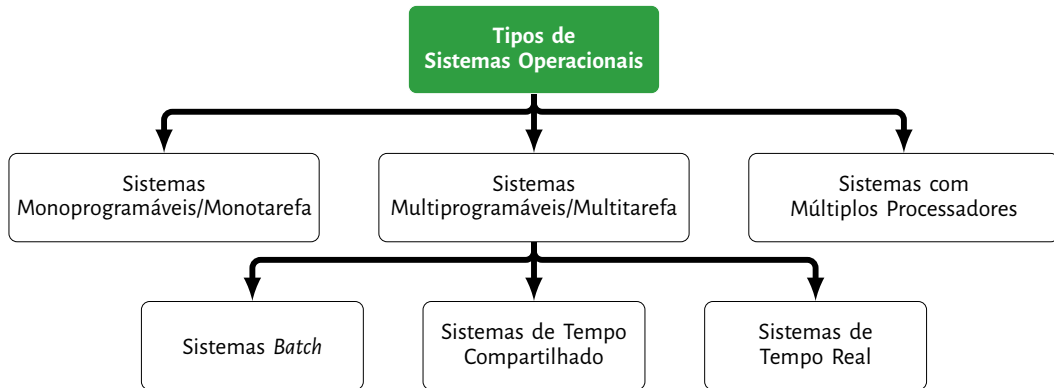
- ▶ Os Sistemas Operacionais tiveram evoluções naturais devido a multiprogramação, que proporcionou:
 - ▶ Compartilhamento de tempo:
 - ▶ Cada usuário possui um terminal próprio e acessa à mesma CPU.
 - ▶ Multiusuário:
 - ▶ Mais de uma sessão podem ser abertas em um computador só, por vários usuários;
 - ▶ Sistemas mais antigos eram monousuários (MS-DOS).
 - ▶ Multitarefa:
 - ▶ Cada usuário pode usar mais de um *job* “simultaneamente”.



TIPOS DE SISTEMAS OPERACIONAIS

**INSTITUTO
FEDERAL**
Pernambuco

TIPOS DE SISTEMAS OPERACIONAIS



TIPOS DE SISTEMAS OPERACIONAIS

MONOTAREFA X MULTITAREFA



► Monotarefa



- São sistemas operacionais que executam uma tarefa por vez;
- Os recursos ficam alocados a um processo (tarefa);
- Exemplo:
 - MS-DOS.

► Multitarefa



- São sistemas operacionais que executam várias tarefas simultaneamente;
- Exemplos:
 - Windows;
 - MacOS.

TIPOS DE SISTEMAS OPERACIONAIS

MONOTAREFA

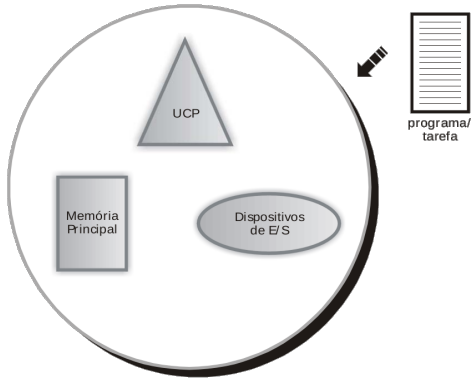


Figura extraída de Machado e Maia (2017, p. 16).

Sinônimos:

Monoprogramáveis = Monotarefa

Definição:

“Os sistemas monoprogramáveis [...] se caracterizam por permitir que o processador, a memória e os periféricos permaneçam exclusivamente dedicados à execução de um único programa. [...] se caracterizam por permitir que todos os recursos do sistema fiquem exclusivamente dedicados a uma única tarefa.” (MACHADO; MAIA, 2017, p. 15).

TIPOS DE SISTEMAS OPERACIONAIS

MULTITAREFA

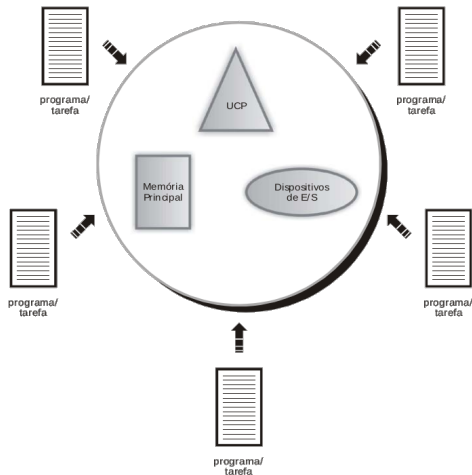


Figura extraída de Machado e Maia (2017, p. 17).

Definição:

“[...] os recursos computacionais são compartilhados entre os diversos usuários e aplicações.

Neste tipo de sistema, por exemplo, enquanto um programa espera por uma operação de leitura ou gravação em disco, outros programas podem estar sendo processados neste mesmo intervalo de tempo.” (MACHADO; MAIA, 2017, p. 16).

TIPOS DE SISTEMAS OPERACIONAIS

MULTITAREFA



- ▶ Sistemas multitarefas ou multiprogramáveis:
 - ▶ Sistemas *batch*
 - ▶ “O processamento *batch* tem a característica de **não exigir a interação do usuário** com a aplicação. Todas as entradas e saídas de dados da aplicação são implementadas por algum tipo de memória secundária, geralmente arquivos em disco.” (MACHADO; MAIA, 2017, p. 18, grifo do professor).
 - ▶ Sistemas de tempo compartilhado
 - ▶ “Os sistemas de tempo compartilhado (*time-sharing*) permite que diversos programas sejam executados a partir da **divisão do tempo do processador** em pequenos intervalos, denominados fatia de tempo (*time-slice*). Caso a fatia de tempo não seja suficiente para a conclusão do programa, ele é interrompido pelo sistema operacional e substituído por um outro [...]” (MACHADO; MAIA, 2017, p. 18, grifo do professor).
 - ▶ Sistemas de tempo real
 - ▶ “[...] os tempos de processamento devem estar dentro de limites rígidos, que devem ser obedecidos, caso contrário poderão ocorrer problemas irreparáveis.” (MACHADO; MAIA, 2017, p. 19).

TIPOS DE SISTEMAS OPERACIONAIS

MONOUSUÁRIO X MULTIUSUÁRIO



- ▶ **Monousuário**
 - ▶ “Sistemas multiprogramáveis monousuários são encontrados em computadores pessoais e estações de trabalho, onde há apenas um único usuário interagindo com o sistema.” (MACHADO; MAIA, 2017, p. 16).
 - ▶ São os sistemas operacionais para um único usuário executar tarefas;
 - ▶ Exemplos: Windows 95, MS-DOS.
- ▶ **Multiusuário**
 - ▶ “Sistemas multiprogramáveis multiusuário são ambientes interativos que possibilitam a diversos usuários conectarem-se ao sistema simultaneamente.” (MACHADO; MAIA, 2017, p. 17).
 - ▶ São sistemas operacionais que permite diversos usuários utilizar simultaneamente os recursos do computador;
 - ▶ Exemplos: Linux, Windows NT.
 - ▶ Todos os sistemas operacionais multiusuários são multitarefas.

TIPOS DE SISTEMAS OPERACIONAIS

MONOPROCESSADOS X MULTIPROCESSADOS



- ▶ Monoprocessados
 - ▶ O sistema operacional utiliza apenas um processador.
 - ▶ Exemplo: Windows 95.
- ▶ Multiprocessados
 - ▶ “Os sistemas com múltiplos processadores caracterizam-se por possuir duas ou mais UCPs interligadas e trabalhando em conjunto. A vantagem deste tipo de sistema é permitir que vários programas sejam executados ao mesmo tempo ou que um mesmo programa seja subdividido em partes para serem executadas simultaneamente em mais de um processador.” (MACHADO; MAIA, 2017, p. 21).
 - ▶ O sistema operacional gerencia mais de um processador simultaneamente;
 - ▶ Exemplos: Windows NT, Linux.

TIPOS DE SISTEMAS OPERACIONAIS

MULTIPROCESSADOS: FORTEMENTE ACOPLADO

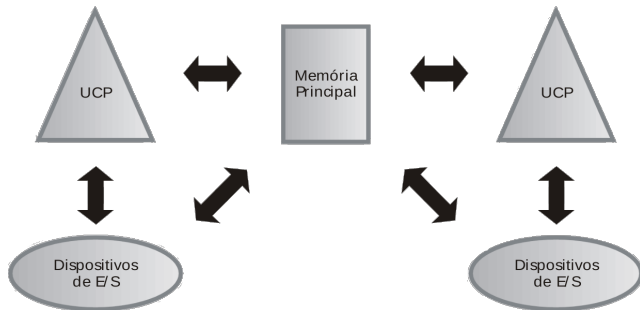


Figura extraída de Machado e Maia (2017, p. 20).

- ▶ Vários processadores;
- ▶ Compartilham uma única memória física e entrada/saída.
 - ▶ Gerenciamento por apenas um sistema operacional;
 - ▶ Multiprocessadores.

TIPOS DE SISTEMAS OPERACIONAIS

MULTIPROCESSADOS: FRACAMENTE ACOPLADO

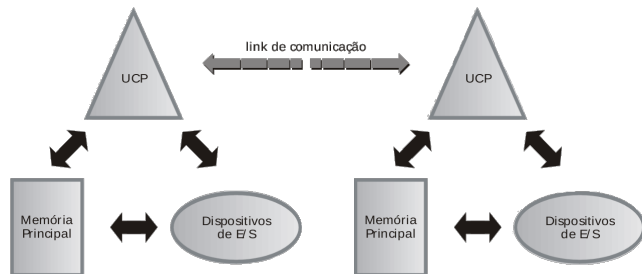
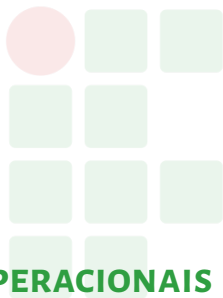


Figura extraída de Machado e Maia (2017, p. 21).

- ▶ Possuem dois ou mais sistemas computacionais;
 - ▶ Conectados através de linha de comunicação;
 - ▶ Cada sistema funciona de forma independente;
 - ▶ Próprio Sistema Operacional;
 - ▶ Gerenciando seus próprios recursos;
 - ▶ Multicomputadores.



CLASSES DE SISTEMAS OPERACIONAIS

**INSTITUTO
FEDERAL**
Pernambuco

CLASSES DE SISTEMAS OPERACIONAIS

SOS DE GRANDE PORTE



- ▶ Multitarefa;
- ▶ Lote (*Batch*);
- ▶ Presente em *mainframes*;
 - ▶ Grande capacidade de E/S;
 - ▶ Grande capacidade de dispositivos;
 - ▶ Processadores, discos, etc.
 - ▶ Exemplos:
 - ▶ OS/360, sistemas Unix.

CLASSES DE SISTEMAS OPERACIONAIS

SOS DE SERVIDORES



- ▶ Multiusuário;
- ▶ Multitarefa;
- ▶ Presente em estações de trabalho;
- ▶ Compartilhamento de recursos e arquivos;
- ▶ Serviços de:
 - ▶ Impressão; Arquivos; Web.
- ▶ Exemplos:
 - ▶ Solaris, FreeBSD, Linux, Windows Server.

CLASSES DE SISTEMAS OPERACIONAIS

SOs DE COMPUTADORES PESSOAIS



- ▶ Multiprogramação;
- ▶ Sistemas Operacionais de Multiprocessadores:
 - ▶ Sistemas focados em processamento em larga escala;
 - ▶ Semelhantes a SOs de Servidores e SOs comuns de PCs.
- ▶ Interface amigável;
- ▶ Utilizados por empresas;
- ▶ Exemplos:
 - ▶ Windows, Linux.

CLASSES DE SISTEMAS OPERACIONAIS

SOS DE COMPUTADORES PORTÁTEIS



- ▶ Multitarefas;
- ▶ Combinam características de um SO do computador pessoal com outros recursos úteis para uso móvel como:
 - ▶ Tela sensível ao toque, celular, Bluetooth, Wi-Fi, GPS, câmera...
- ▶ Presente em Palms, Celulares, Tablets.

CLASSES DE SISTEMAS OPERACIONAIS

SOS DE EMBARCADOS



- ▶ São sistemas que controlam dispositivos diferente de computadores pessoais;
 - ▶ Controlam dispositivos que “não são considerados computadores”;
 - ▶ Garante que nenhum *software* não confiável será instalado nele;
 - ▶ Exemplos de dispositivos:
 - ▶ Forno de Micro-ondas, Smart TVs, aparelhos de DVD, videogames, celulares mais antigos.
- ▶ Multitarefa;
- ▶ Dedicado ao recurso.

CLASSES DE SISTEMAS OPERACIONAIS

SOS DE TEMPO REAL



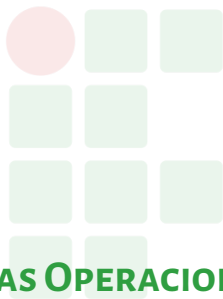
- ▶ São sistemas que utilizam o tempo como parâmetro fundamental;
- ▶ Presente em Processos Industriais;
 - ▶ Tempo é fundamental;
 - ▶ Chance mínima de falhas.
- ▶ Sistema de Tempo Real Crítico vs. Sistema de Tempo Real Não-Crítico;
 - ▶ Executa apenas *software* instalados pelos projetistas, assim como os SOs embarcados.

CLASSES DE SISTEMAS OPERACIONAIS

SOS DE TEMPO REAL CRÍTICO



- ▶ Além do tempo ser fundamental, as precisam ações precisam ocorrer em um determinado instante de tempo;
- ▶ Presente em Aviões, Foguetes.



ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

**INSTITUTO
FEDERAL**
Pernambuco

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

| Funções | System calls |
|---|--|
| Gerência de processos de <i>threads</i> | Criação e eliminação de processos e <i>threads</i> |
| | Alteração das características de processos e <i>threads</i> |
| | Sincronização e comunicação entre processos e <i>threads</i> |
| | Obtenção de informações sobre processos e <i>threads</i> |
| Gerência de memória | Alocação e desalocação de memória |
| Gerência do sistema de arquivos | Criação e eliminação de arquivos e diretórios |
| | Abrir das características de arquivos e diretórios |
| | Abrir e fechar arquivos |
| | Leitura e gravação em arquivos |
| | Obtenção de informações sobre arquivos e diretórios |
| Gerência de dispositivos | Alocação e desalocação de dispositivos |
| | Operações de entrada/saída em dispositivos |
| | Obtenção de informações sobre dispositivos |

Tabela extraída de Machado e Maia (2017, p. 55).

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE HARDWARE

“Um sistema operacional está intimamente ligado ao *hardware* do computador no qual ele é executado. Ele estende o conjunto de instruções do computador e gerencia seus recursos.” (TANENBAUM; BOS, 2016, p. 14).

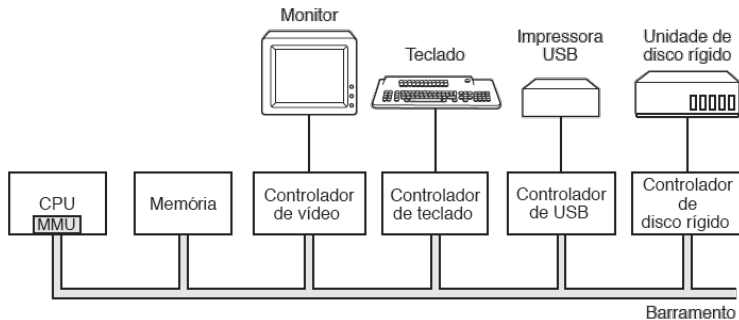


Figura extraída de Tanenbaum e Bos (2016, p. 14).

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE HARDWARE: CPU



“O ‘cérebro’ do computador é a CPU. Ela busca instruções da memória e as executa. O ciclo básico de toda CPU é buscar a primeira instrução da memória, decodificá-la para determinar o seu tipo e operandos, executá-la, e então buscar, decodificar e executar as instruções subsequentes. O ciclo é repetido até o programa terminar.” (TANENBAUM; BOS, 2016, p. 15).

1. Registradores internos para armazenamento de variáveis e resultados temporários:
 - ▶ Contador de programa: contém o endereço de memória da próxima instrução a ser buscada;
 - ▶ Ponteiro de pilha: aponta para o topo da pilha atual na memória;
 - ▶ PSW (*Program Status Word* - palavra de estado do programa): contém os bits do código de condições, que são estabelecidos por instruções de comparação, a prioridade da CPU, o modo de execução (usuário ou núcleo) e vários outros bits de controle.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE HARDWARE: CPU



2. CPUs modernas têm recursos para executar mais de uma instrução ao mesmo tempo:

2.1 *Pipelining*

- Uma CPU pode ter unidades de busca, decodificação e execução separadas, assim enquanto ela está executando a instrução n , poderia também estar decodificando a instrução $n + 1$ e buscando a instrução $n + 2$.



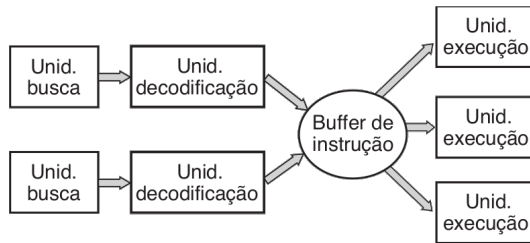
ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE HARDWARE: CPU



2. CPUs modernas têm recursos para executar mais de uma instrução ao mesmo tempo:

2.2 CPU Superescalar



- ▶ Unidades múltiplas de execução estão presentes;
- ▶ Duas ou mais instruções são buscadas ao mesmo tempo, decodificadas e jogadas em um buffer de instrução até que possam ser executadas;
- ▶ Tão logo uma unidade de execução fica disponível, ela procura no buffer de instrução para ver se há uma instrução que ela pode executar e, se assim for, ela remove a instrução do buffer e a executa.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE HARDWARE: CPU

2. CPUs modernas têm recursos para executar mais de uma instrução ao mesmo tempo:

2.3 Chamadas de Sistema

- Tipo especial de procedimento de instrução de chamada que tem a propriedade adicional de chavear do modo usuário para o modo núcleo.

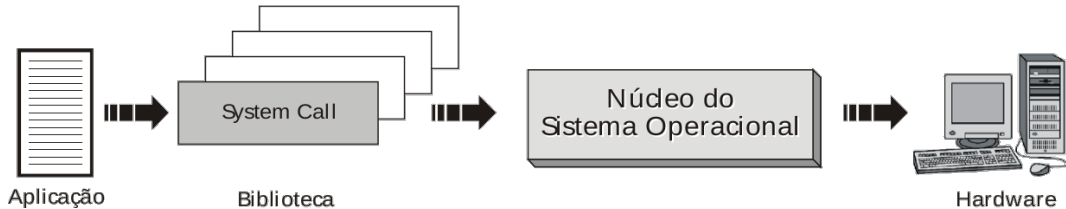


Figura extraída de Machado e Maia (2017, p. 54).

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE HARDWARE: CPU



3. Chips multithread e multinúcleo

3.1 *Multithread*

- ▶ A medida que o número de transistores aumenta, é necessário colocar memórias *cache* maiores no chip da CPU;
- ▶ Replicar não apenas as unidades funcionais, mas também parte da lógica de controle.
- ▶ Pentium 4 da Intel:
 - Permite que a CPU mantenha o estado de dois *threads* diferentes e então faça o chaveamento entre um e outro em uma escala de tempo de nanossegundos;
 - Por exemplo, se um dos processos precisa ler uma palavra da memória (o que leva muitos ciclos de relógio), uma CPU *multithread* pode simplesmente fazer o chaveamento para outro *thread*;
 - O *multithreading* não proporciona paralelismo real. Apenas um processo de cada vez é executado, mas o tempo de chaveamento de *thread* é reduzido.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE HARDWARE: CPU



3. Chips *multithread* e multinúcleo

3.2 Multinúcleo

- ▶ Os chips multinúcleo trazem quatro cada um com sua CPU independente;
- ▶ Alguns processadores, como o Intel Xeon Phi e o Tileria TilePro, já apresentam mais de 60 núcleos em um único chip;
- ▶ Fazer uso de um chip com múltiplos núcleos como esse definitivamente exigirá um sistema operacional de multiprocessador.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE *HARDWARE*: MEMÓRIAS



São o segundo principal componente em qualquer computador, o qual deve ser rápido ao extremo (mais rápida do que executar uma instrução, de maneira que a CPU não seja atrasada pela memória).

| Tempo típico de acesso | | Capacidade típica |
|------------------------|-------------------|-------------------|
| 1 ns | Registradores | <1 KB |
| 2 ns | Cache | 4 MB |
| 10 ns | Memória principal | 1-8 GB |
| 10 ms | Disco magnético | 1-4 TB |

Figura extraída de Tanenbaum e Bos (2016, p. 17).

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE *HARDWARE*: MEMÓRIAS



1. Registradores

- ▶ Eles são feitos do mesmo material que a CPU e são, desse modo, tão rápidos quanto ela;
- ▶ A capacidade de armazenamento disponível neles é tipicamente 32 x 32 bits em uma CPU de 32 bits e 64 x 64 bits em uma CPU de 64 bits. Menos de 1 KB em ambos os casos;
- ▶ Os programas devem gerenciar os próprios registradores (isto é, decidir o que manter neles) no *software*.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE *HARDWARE*: MEMÓRIAS



2. *Cache*

- ▶ É controlada principalmente pelo *hardware*;
- ▶ A memória principal é dividida em linhas de *cache*, tipicamente 64 bytes, com endereços 0 a 63 na linha de *cache* 0, 64 a 127 na linha de *cache* 1 e assim por diante;
- ▶ As linhas de *cache* mais utilizadas são mantidas em uma *cache* de alta velocidade localizada dentro ou muito próximo da CPU.
- ▶ Se ela estiver presente na *cache* (*cache hit*), a requisição é atendida e nenhuma requisição de memória é feita para a memória principal sobre o barramento;
- ▶ Se a linha requisitada estiver ausente da *cache* (*cache miss*), uma requisição adicional é feita à memória, com uma penalidade de tempo substancial;
- ▶ A memória da *cache* é limitada em tamanho por causa do alto custo.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE HARDWARE: MEMÓRIAS



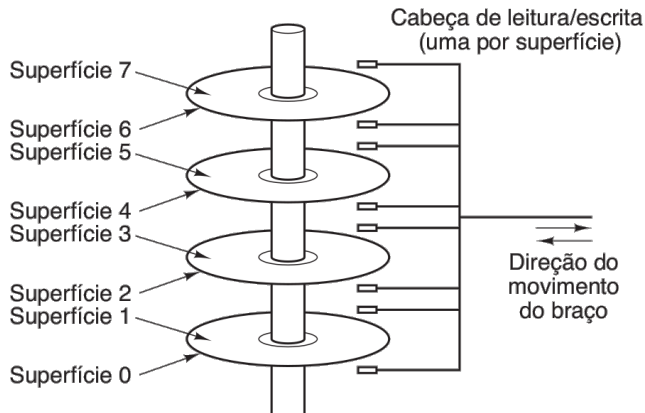
3. Memória principal

- ▶ Memória RAM (*Random Access Memory* - memória de acesso aleatório):
 - ▶ Memória principal;
 - ▶ Memória volátil;
 - ▶ Todas as requisições da CPU que não podem ser atendidas pela *cache* vão para a memória principal.
- ▶ Memória ROM (*Read Only Memory* - memória somente de leitura):
 - ▶ Não volátil, ou seja, a memória não perde o seu conteúdo quando a energia é desligada;
 - ▶ Programada na fábrica e não pode ser modificada depois;
 - ▶ Ela é rápida e barata.
- ▶ EEPROM (Electrically Erasable PROM - ROM eletricamente apagável) e a memória *flash*:
 - ▶ Também são não voláteis, mas, diferentemente da ROM, podem ser apagadas e reescritas;
 - ▶ A memória *flash* é bastante usada como um meio de armazenamento em dispositivos eletrônicos portáteis.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE HARDWARE: MEMÓRIAS

4. Disco magnético



extraída de Tanenbaum e Bos (2016, p. 19).

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE PROCESSOS



| Chamada | Descrição |
|--|---|
| <code>pid = fork()</code> | Cria um processo filho idêntico ao pai |
| <code>pid = waitpid(pid, &statloc, options)</code> | Espera que um processo filho seja concluído |
| <code>s = execve(name, argv, environp)</code> | Substitui a imagem do núcleo de um processo |
| <code>exit(status)</code> | Conclui a execução do processo e devolve status |

Tabela extraída de Tanenbaum e Bos (2016, p. 38).

- ▶ Distribuir a capacidade de processamento;
- ▶ Sincronização de processos.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE MEMÓRIA



- ▶ Fornecer para as tarefas (processos) um espaço de memória;
- ▶ Isolado e independente;
- ▶ Podem utilizar o disco como memória virtual.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE ARQUIVOS



| Chamada | Descrição |
|---|---|
| <code>fd = open(file, how, ...)</code> | Abre um arquivo para leitura, escrita ou ambos |
| <code>s = close(fd)</code> | Fecha um arquivo aberto |
| <code>n = read(fd, buffer, nbytes)</code> | Lê dados a partir de um arquivo em um buffer |
| <code>n = write(fd, buffer, nbytes)</code> | Escreve dados a partir de um buffer em um arquivo |
| <code>position = lseek(fd, offset, whence)</code> | Movimenta o ponteiro do arquivo |
| <code>s = stat(name, &buf)</code> | Obtém informações sobre um arquivo |

Tabela extraída de Tanenbaum e Bos (2016, p. 38).

- ▶ Interage com o gerenciamento de dispositivos;
- ▶ Abstração de arquivos e diretórios.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE DIRETÓRIOS E ARQUIVOS



| Chamada | Descrição |
|---|--|
| <code>s = mkdir(name, mode)</code> | Cria um novo diretório |
| <code>s = rmdir(name)</code> | Remove um diretório vazio |
| <code>s = link(name1, name2)</code> | Cria uma nova entrada, name2, apontando para name1 |
| <code>s = unlink(name)</code> | Remove uma entrada de diretório |
| <code>s = mount(special, name, flag)</code> | Monta um sistema de arquivos |
| <code>s = umount(special)</code> | Desmonta um sistema de arquivos |

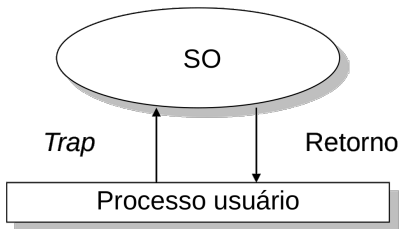
Tabela extraída de Tanenbaum e Bos (2016, p. 38).

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE DISPOSITIVOS



- ▶ Controla a E/S (Pen-drives, discos de armazenamento, DVD, etc.);
- ▶ Controladores;
- ▶ As instruções de E/S são privilegiadas;
 - ▶ Para proteger os periféricos:
 - ▶ Um processo usuário não pode acessá-los;
 - ▶ Exemplos: Escrita em disco, Leitura de um CD.
- ▶ O usuário deve passar pelo SO através de uma chamada de sistema, que gera uma interrupção (trap).



- ▶ Chamada de sistema oferece o serviço ao processo usuário, em modo “seguro”;
- ▶ Gera uma interrupção;
 - ▶ Troca de contexto.
- ▶ O processo deve deixar o lugar para o código do núcleo.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

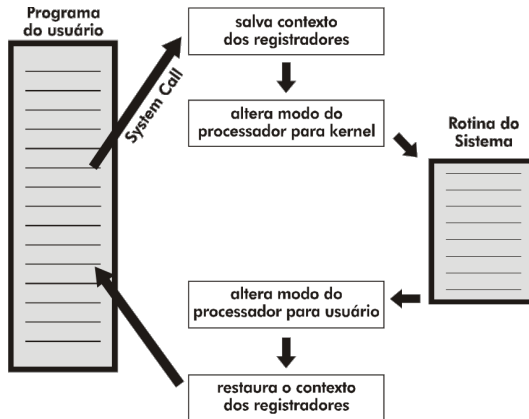
GERENCIAMENTO DE PROTEÇÃO



- ▶ Políticas de acesso aos recursos;
 - ▶ Tipos de instruções:
 - ▶ Privilegiadas;
 - ▶ Não-privilegiadas.
- ▶ Um processo só pode acessar os recursos que tem permissão;
 - ▶ Modos de acesso - Usuário:
 - ▶ Uso limitado;
 - ▶ Os processos usuários operam neste modo;
 - ▶ Alterna para o modo *kernel* por interrupção.
 - ▶ Modo de acesso - *Kernel* ou supervisor (núcleo):
 - ▶ Todo o conjunto de operações é disponível;
 - ▶ É o modo do Sistema Operacional;
 - ▶ Privilegiado, protegido;
 - ▶ Retorna para o modo usuário por instruções clássicas: *yield*, *return*, ...
- ▶ Controle de usuários e grupos.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE PROTEÇÃO: CHAMADAS DE SISTEMA



ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE PROTEÇÃO: CHAMADAS DE SISTEMA

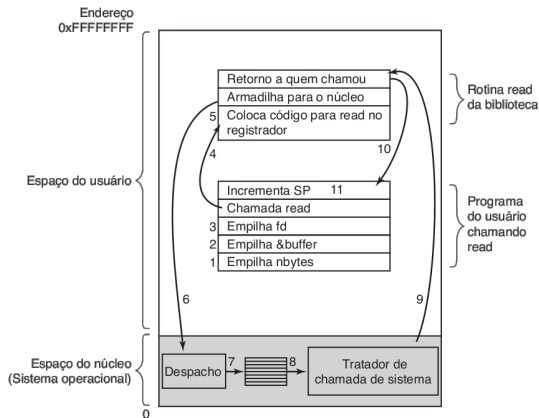


Figura extraída de Machado e Maia (2017, p. 36).

1. a 3. Preparação para chamar a rotina de biblioteca `read`, empilhando os parâmetros.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE PROTEÇÃO: CHAMADAS DE SISTEMA

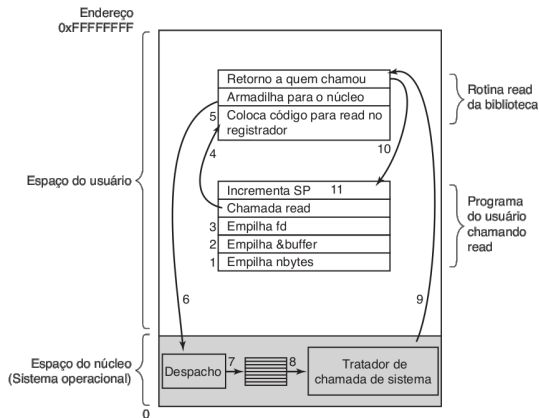


Figura extraída de Machado e Maia (2017, p. 36).

4. Chamada real para a rotina de biblioteca read.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE PROTEÇÃO: CHAMADAS DE SISTEMA

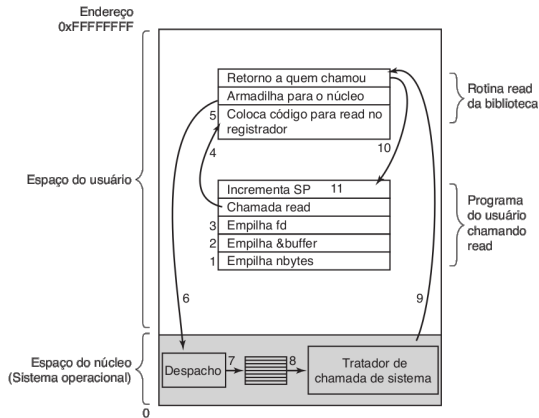


Figura extraída de Machado e Maia (2017, p. 36).

- Coloca o número da chamada de sistema em um lugar onde o sistema operacional a espera, como um registro.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE PROTEÇÃO: CHAMADAS DE SISTEMA

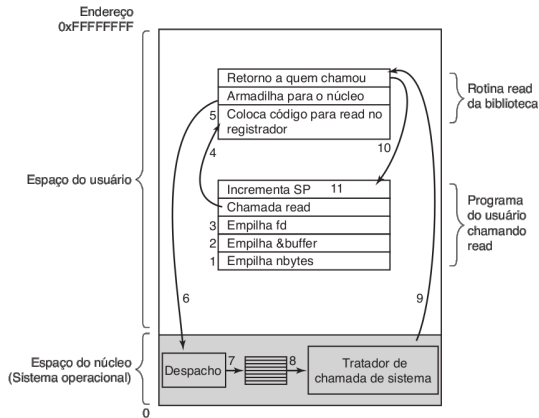


Figura extraída de Machado e Maia (2017, p. 36).

- Executa uma instrução TRAP para passar do modo usuário ao modo núcleo e começa a execução num endereço fixo no núcleo.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE PROTEÇÃO: CHAMADAS DE SISTEMA

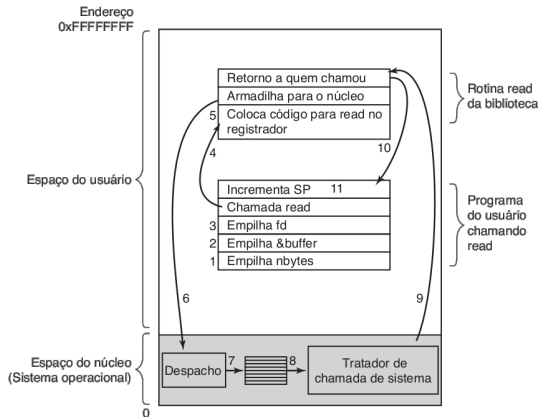


Figura extraída de Machado e Maia (2017, p. 36).

7. Examina o número da chamada de sistema e então o despacha para o tratador correto da chamada de sistema.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE PROTEÇÃO: CHAMADAS DE SISTEMA

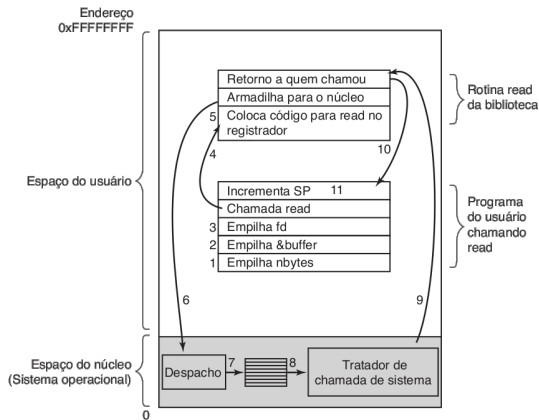


Figura extraída de Machado e Maia (2017, p. 36).

8. Executado o tratamento de chamada de sistema.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE PROTEÇÃO: CHAMADAS DE SISTEMA

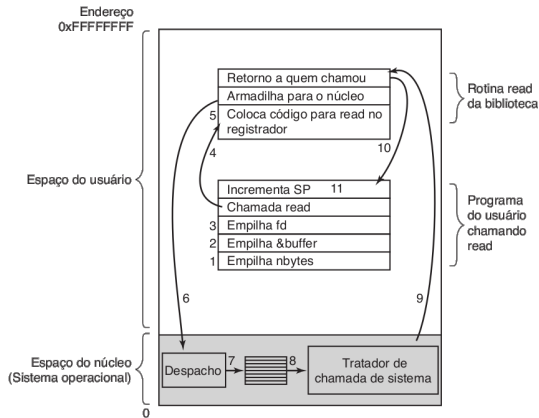


Figura extraída de Machado e Maia (2017, p. 36).

- O controle pode ser retornado para a rotina de biblioteca no espaço do usuário na instrução após a instrução TRAP.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE PROTEÇÃO: CHAMADAS DE SISTEMA

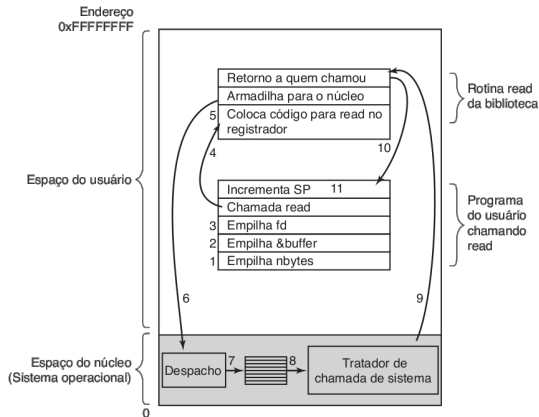


Figura extraída de Machado e Maia (2017, p. 36).

10. Essa rotina retorna para o programa do usuário da maneira usual que as chamadas de rotina retornam.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE PROTEÇÃO: CHAMADAS DE SISTEMA

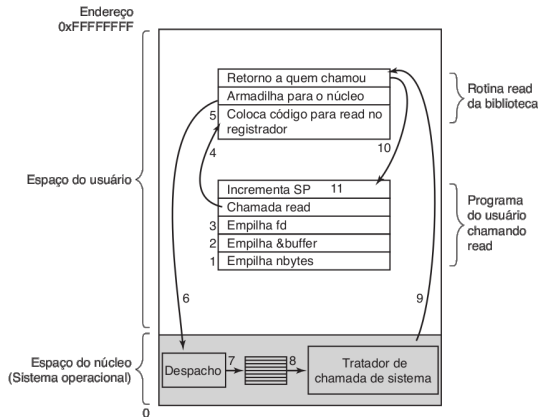


Figura extraída de Machado e Maia (2017, p. 36).

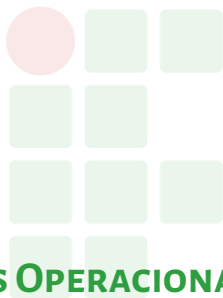
11. Para terminar a tarefa, o programa do usuário tem de limpar a pilha, como ele faz após qualquer chamada de rotina.

ATRIBUIÇÕES DO SISTEMAS OPERACIONAIS

GERENCIAMENTO DE PROTEÇÃO: CHAMADAS DE SISTEMA



- ▶ Exemplos de chamadas de sistemas (*system calls*):
 - ▶ Fork:
 - ▶ Copia a imagem do pai;
 - ▶ As variáveis de ambiente;
 - ▶ File descriptors (Pode redirecionar antes do `execve`).
 - ▶ Execve:
 - ▶ Substitui a imagem.
 - ▶ Win32:
 - ▶ `CreateProcess`.



ESTRUTURA DE SISTEMAS OPERACIONAIS

**INSTITUTO
FEDERAL**
Pernambuco

ESTRUTURA DE SISTEMAS OPERACIONAIS

SISTEMAS MONOLÍTICOS

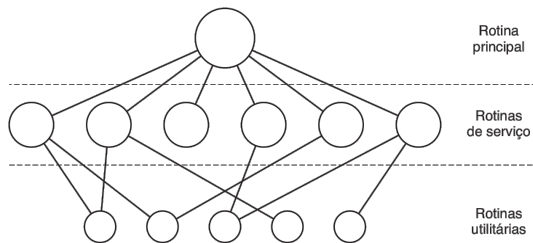


Figura extraída de Tanenbaum e Bos (2016, p. 44).

- ▶ Estrutura básica:
 - ▶ Programa principal que invoca a rotina do serviço requisitado;
 - ▶ Conjunto de rotinas de serviços que executam as chamadas de sistemas;
 - ▶ Conjunto de rotinas utilitárias que auxiliam as rotinas de serviço.
- ▶ Sistema Operacional executado como um único programa no modo núcleo;
- ▶ Mais comum.

ESTRUTURA DE SISTEMAS OPERACIONAIS

SISTEMA DE CAMADAS



- ▶ Organizado como uma hierarquia de camadas;
- ▶ Cada camada criada sobre a camada imediatamente inferior;
- ▶ Primeiro sistema de camadas THE - *Technische Hogeschool Eindhoven* (1968):
 - ▶ Estrutura do Sistema Operacional THE:

| Camada | Função |
|--------|--|
| 5 | O operador |
| 4 | Programas de usuário |
| 3 | Gerenciamento de entrada/saída |
| 2 | Comunicação operador–processo |
| 1 | Memória e gerenciamento de tambor |
| 0 | Alocação do processador e multiprogramação |

ESTRUTURA DE SISTEMAS OPERACIONAIS

MICRONÚCLEO



- ▶ Modularização:
 - ▶ Divisão do sistema operacional em módulos pequenos;
 - ▶ Executado em modo núcleo e o restante em modo usuário;
 - ▶ Diversos módulos executados em modo de usuário.
- ▶ Alcançar alta confiabilidade;
 - ▶ Falhas não derrubam todo o sistema.
 - ▶ Erros pode para de funcionar apenas aquele componente e não o sistema inteiro.
 - ▶ Densidade de erros proporcional ao tamanho do módulo.
 - ▶ Sistemas industriais sérios: 10 erros para cada 1000 linhas de código.
 - ▶ Sistemas Operacionais monolíticos com núcleo de 5 milhões de linhas = 50 mil erros no núcleo.
- ▶ Enxugar ao máximo o núcleo;
- ▶ No entanto, por não ter acesso a recursos de mais baixo nível, não pode dar detalhes de *debug* do nível de, por exemplo, endereço exato da instrução com erro.

ESTRUTURA DE SISTEMAS OPERACIONAIS

MODELO CLIENTE-SERVIDOR



- ▶ O modelo cliente-servidor é uma abstração que pode ser usada em uma única máquina ou em uma rede de máquinas;
 - ▶ Classes de processos servidores que prestam serviços e clientes que consomem esses serviços;
 - ▶ Processos locais ou remotos.
- ▶ Comunicação por meio da troca de mensagens:
 - ▶ O cliente constrói uma mensagem dizendo o que deseja e envia ao servidor;
 - ▶ O servidor recebe a mensagem, processa e responde ao cliente emissor.
- ▶ Variação do micronúcleo.

ESTRUTURA DE SISTEMAS OPERACIONAIS

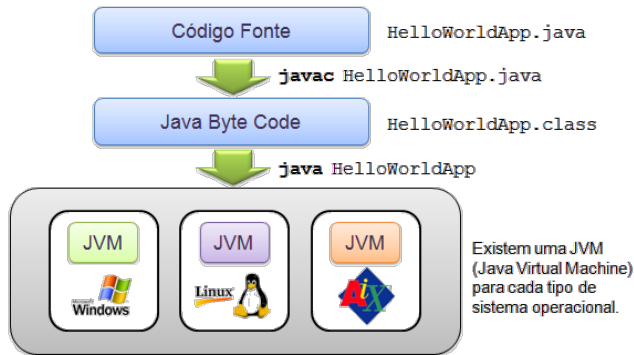
MÁQUINAS VIRTUAIS



- ▶ “É visto como uma duplicata eficiente e isolada de uma máquina real” (POPEK; GOLDBERG, 1974);
- ▶ A abstração é construída por um monitor de máquina virtual (VMM - *Virtual Machine Monitor*);
 - ▶ Fornece um ambiente de execução idêntico ao da máquina real;
 - ▶ Controlar totalmente os recursos do sistema real.
- ▶ Propriedades:
 - ▶ Equivalência:
 - ▶ Um ambiente idêntico ao da máquina real.
 - ▶ Controle de Recursos:
 - ▶ Controle total dos recursos da máquina real.
 - ▶ Eficiência:
 - ▶ As instruções da UCP virtual deve ser executada pela UCP real.
 - ▶ Isolamento:
 - ▶ Aplicações de uma máquina virtual não pode interagir diretamente com a máquina real.

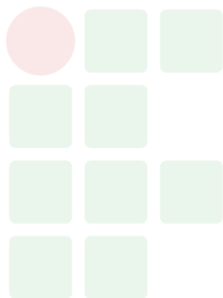
ESTRUTURA DE SISTEMAS OPERACIONAIS

MÁQUINAS VIRTUAIS JAVA



- ▶ Pode ser executado em qualquer plataforma que implementa a JVM;
- ▶ A virtualização é o fator responsável pela independência dos programas em JAVA do *hardware* e Sistemas operacionais.

REFERÊNCIAS





INSTITUTO
FEDERAL
Pernambuco

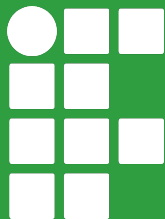
REFERÊNCIAS I



 MACHADO, Francis Berenger; MAIA, Luiz Paulo. **Arquitetura de Sistemas Operacionais**. 5. ed. Rio de Janeiro: LTC, 2017. ISBN 978-85-216-2210-9.

 POPEK, Gerald J.; GOLDBERG, Robert P. Formal requirements for virtualizable third generation architectures. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 17, n. 7, p. 412–421, jul. 1974. Disponível em: <<https://doi.org/10.1145/361011.361073>>. Acesso em: 17 jun. 2020.

 TANENBAUM, Andrew S.; BOS, Herbert. **Sistemas Operacionais Modernos**. 4. ed. São Paulo: Pearson Education do Brasil, 2016.



INSTITUTO FEDERAL

Pernambuco