UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS DE RUSSAS

Rus0013 - Sistemas Operacionais

Aula 02: Chamadas ao Sistema & Estruturas de Sistemas Operacionais

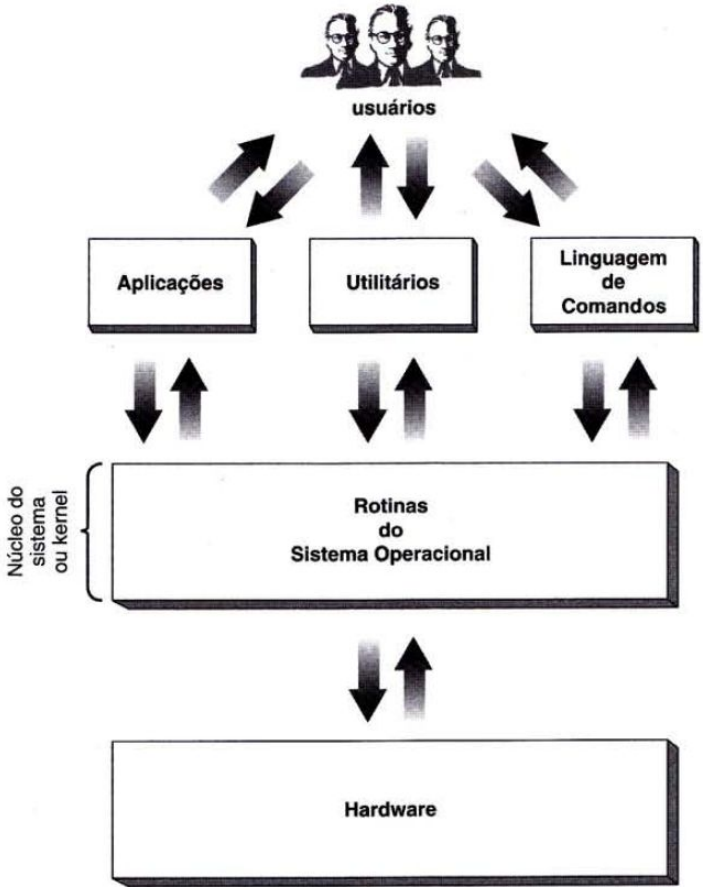
Professor Pablo Soares

2022.2

Sumario

- Proteção do Núcleo
- Chamadas ao Sistema
- Estrutura de Sistemas Operacionais
  - Sistemas Monolíticos
  - Sistemas de Camadas
  - Máquinas Virtuais
  - Exonúcleos
  - O Modelo cliente-Servidor

Estrutura de Um SO



Proteção do Núcleo

- Um SO deve gerenciar os recursos de Hardware
  - Integridade dessa gerência
    - Aplicações não consigam acessar o hardware diretamente
    - Pedir ao SO
  - Como Impedir?
    - Níveis de privilegio
    - Núcleo e drivers → acesso direto
    - Utilitários e aplicativos → acesso restrito
      - Desestabilizando o sistema

Proteção do Núcleo

- SO tem dois ou mais níveis de privilegio
  - Controlados por *flags* especiais no processador
  - Controle estritamente pelo processador
- Nível Núcleo
  - Supervisor
  - Kernel
- Nível usuário
  - Userspace

Proteção do Núcleo

- Nível Núcleo
  - Nesse nível, todo o processador está acessível
  - Registradores e portas de entrada/saída
  - Áreas da memória podem ser acessadas
  - Todas as instruções do processador podem ser executadas

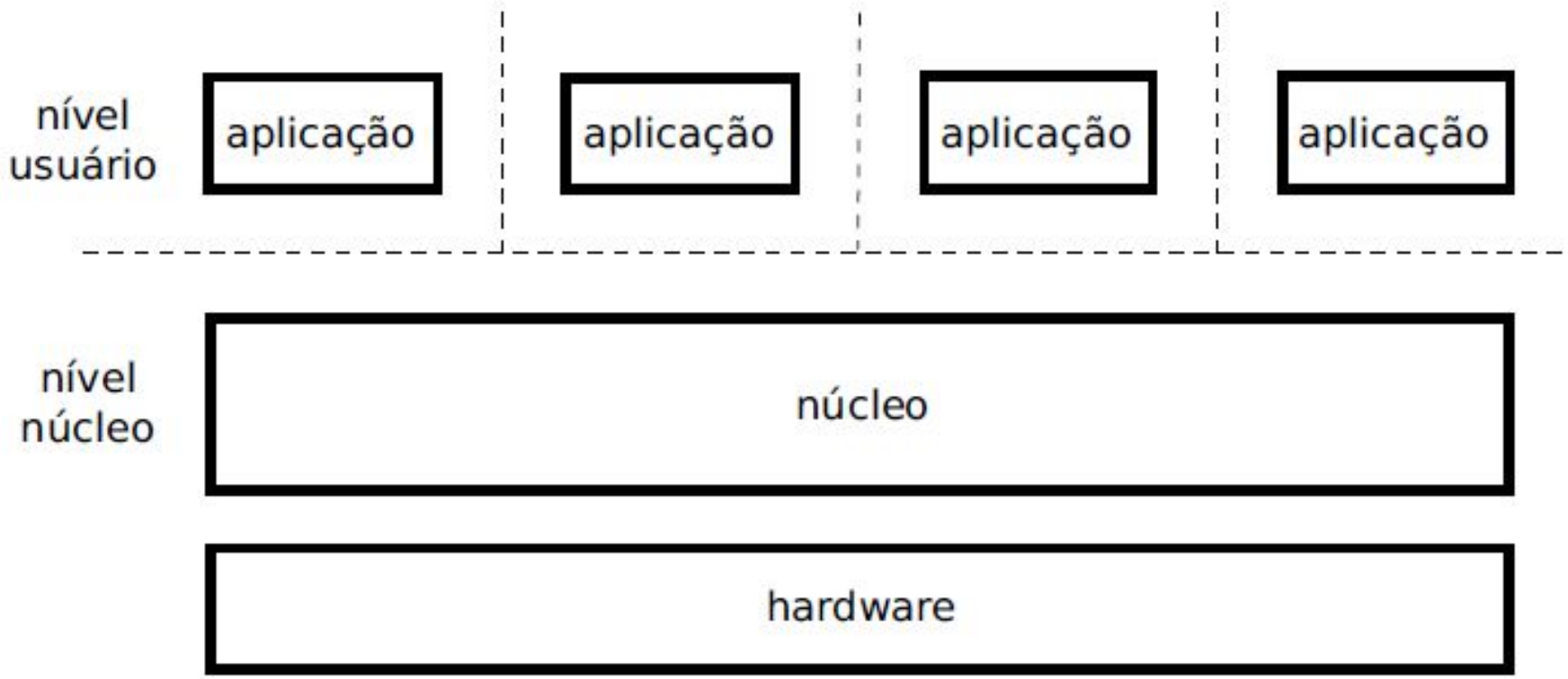
Funções do Núcleo

- Tratamentos de interrupções;
- Criação e eliminação de processos e threads;
- Sincronização de processos e threads
- Escalonamento e controle dos processos e threads
- Gerência de memória
- Gerência de arquivos
- Gerência de E/S
- Suporte a redes locais
- Segurança do Sistema

Proteção do Núcleo

- Nível Usuário
  - Nesse nível, subconjunto de instruções estão disponíveis
    - Registradores
    - Portas de E/S
  - Instruções perigosas são proibidas
    - HALT (parar o processador)
    - RESET(reiniciar o processador)
  - Restringi o uso de memória
    - Áreas previamente definidas
  - Se um código tentar executar um instrução proibida
    - Núcleo faz um tratamento
      - “este programa executou uma instrução ilegal e será finalizado”,

Proteção do Núcleo



Separação entre o núcleo e as aplicações

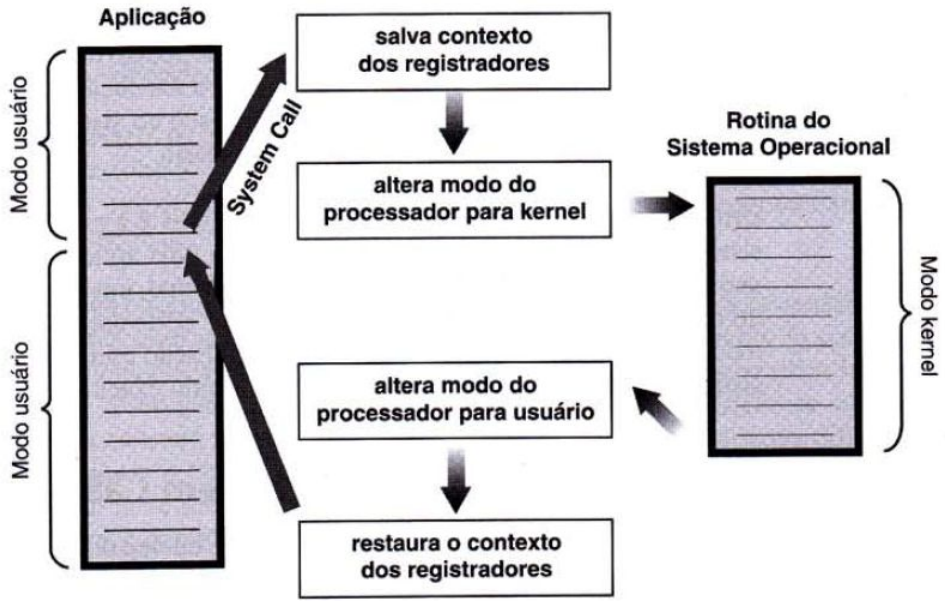
Chamadas ao Sistema

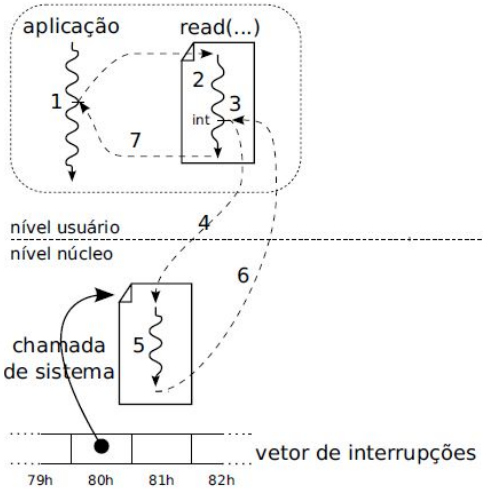
- Confinamento de uma aplicação em sua área de memória pelo (MMU - *Memory Managment Unit*)
  - Robustez
  - Confiabilidade do sistema (Proteção)
- Como uma aplicação pode acessar a placa de rede para enviar/receber dados, se não tem privilégio para acessar as portas de entrada/saída correspondentes nem pode invocar o código do núcleo que implementa esse acesso (pois esse código reside em outra área de memória)?
  - Chamada ao Sistema

Chamadas ao Sistema

- SO definem chamadas ao sistema
  - Acesso a recursos de baixo nível
    - Periféricos
    - Arquivos
    - Alocação de Memória
    - Criação/finalização de tarefas
  - Chamadas ao sistema são oferecidas para aplicações de modo usuário
    - Biblioteca do sistema
      - Parâmetros
      - Interrupções
      - Retorna a aplicação

Chamadas ao Sistema





Chamadas ao Sistema

- Gerenciamento de Processos
- Gerenciamento de Arquivos

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

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 ao Sistema

- Gerenciamento de sistemas de dirétorio
- Diversas

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

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

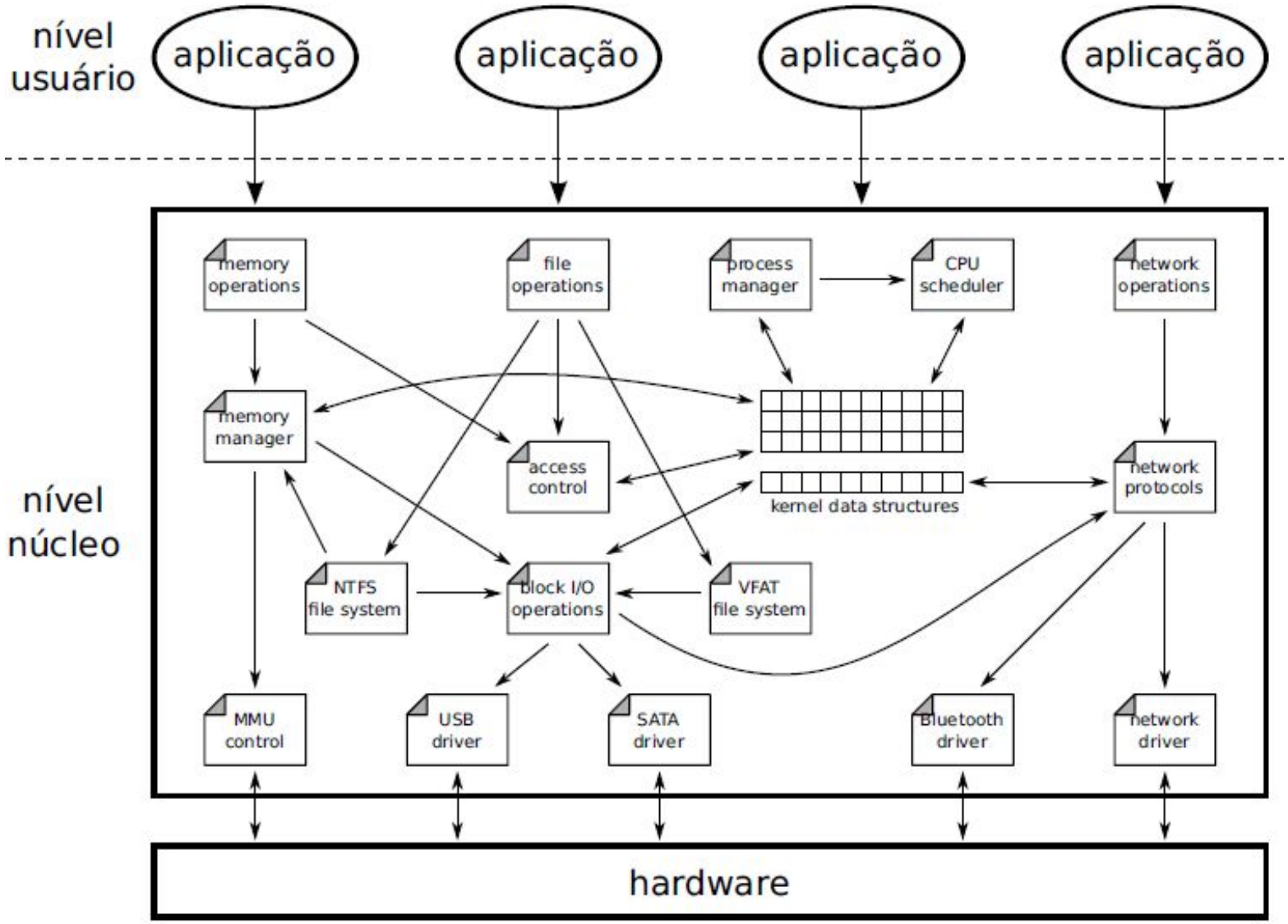
Estruturas de Sistemas Operacionais

- Estrutura interna de um SO
- Projetos
  - Sistemas Monolíticos
  - Sistemas de Camadas
  - Máquinas Virtuais
  - ExoNúcleo
  - MicroKernel

Sistema Monolíticos

- Organização
  - É a mais comum
  - Não há estruturação(A grande Bagunça)
  - Coleção de Procedimentos
    - Todos podem chamar os demais
  - Vantagem
    - Desempenho
  - Desvantagem
    - Caso algum componente perca o controle devido a algum erro → Travamento do sistema

Sistema Monolíticos



Sistema de Camadas

- Organização
  - Hierarquia de camadas
  - Construídas sobre a camada inferior

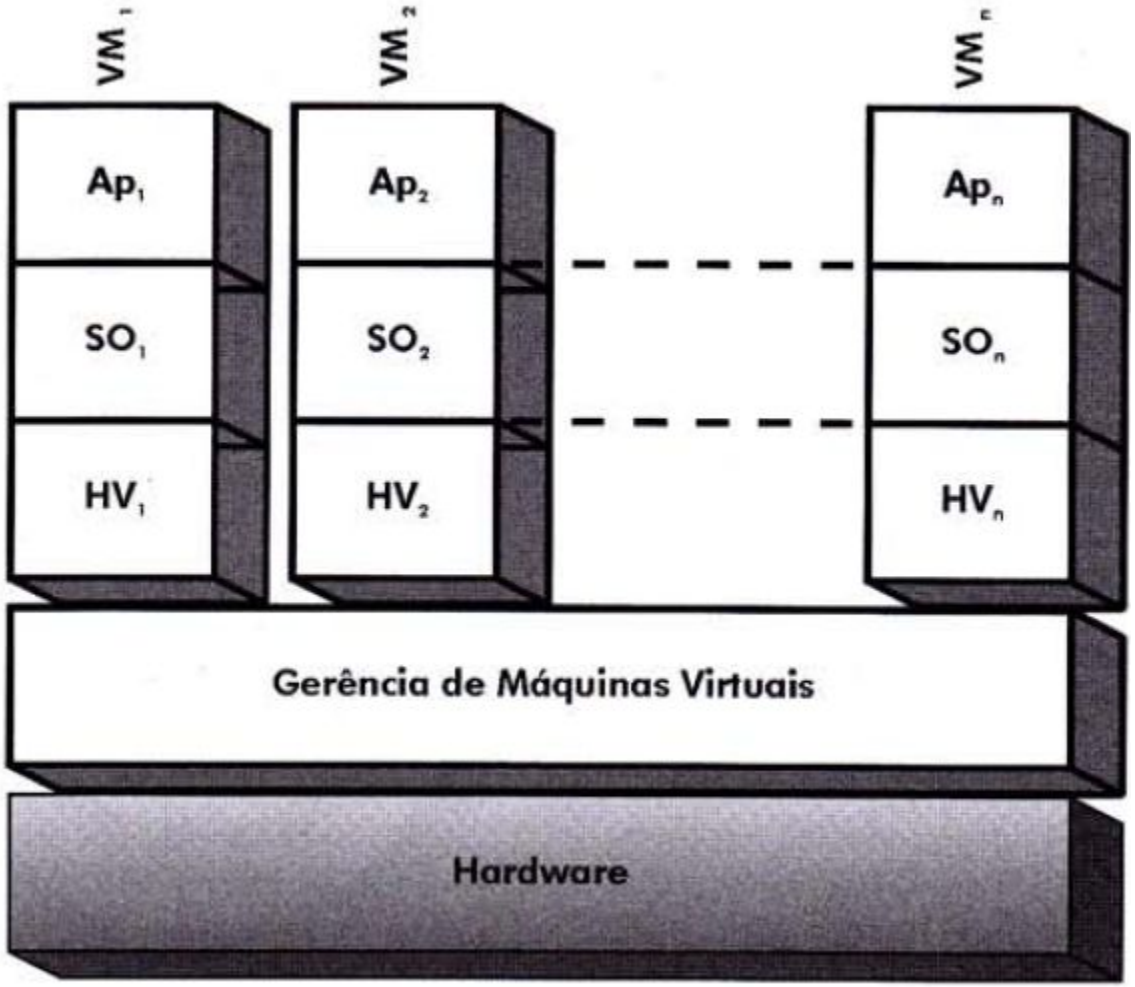
Camada	Função
5	Operador
4	Programa usuário
3	Gerenciamento de entrada/saída
2	Comunicação operador-processo
1	Gerenciamento de memória e tambor
0	Alocação do processador e multiprogramação

A estrutura do sistema operacional THE.

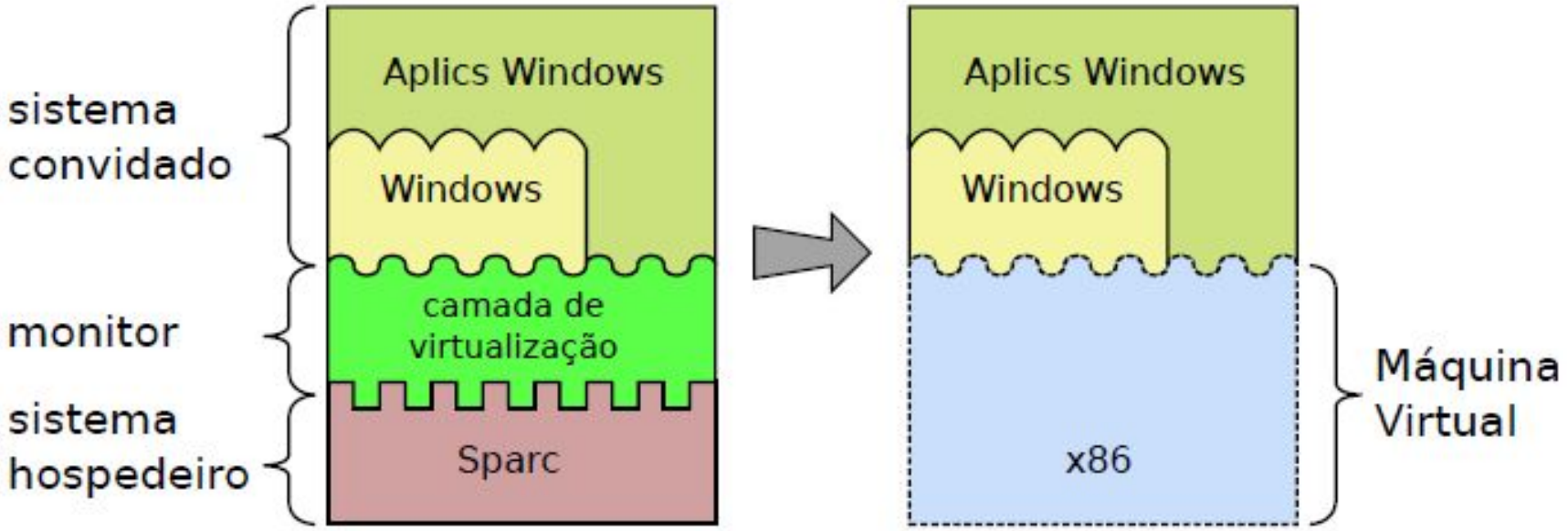
Sistema de Camadas

- Essa estruturação fez muito sucesso
  - Isolar as funções do sistema operacional
    - Manutenção e depuração
  - Redes de computadores
    - Modelo de Referência OSI – (*Open Systems Interconnction*)
- Inconvenientes
  - Empilhamento de varias camadas de software
    - Pedido de uma camada superior leva muito tempo
    - Camadas são inter-dependentes

Máquinas Virtuais



Máquinas Virtuais



Uma máquina virtual

- Sistema Real ou hospedeiro
  - Recursos reais
- O sistema virtual (convidado)
  - Executa sobre virtualizado
- Camada de virtualização
  - Monitor

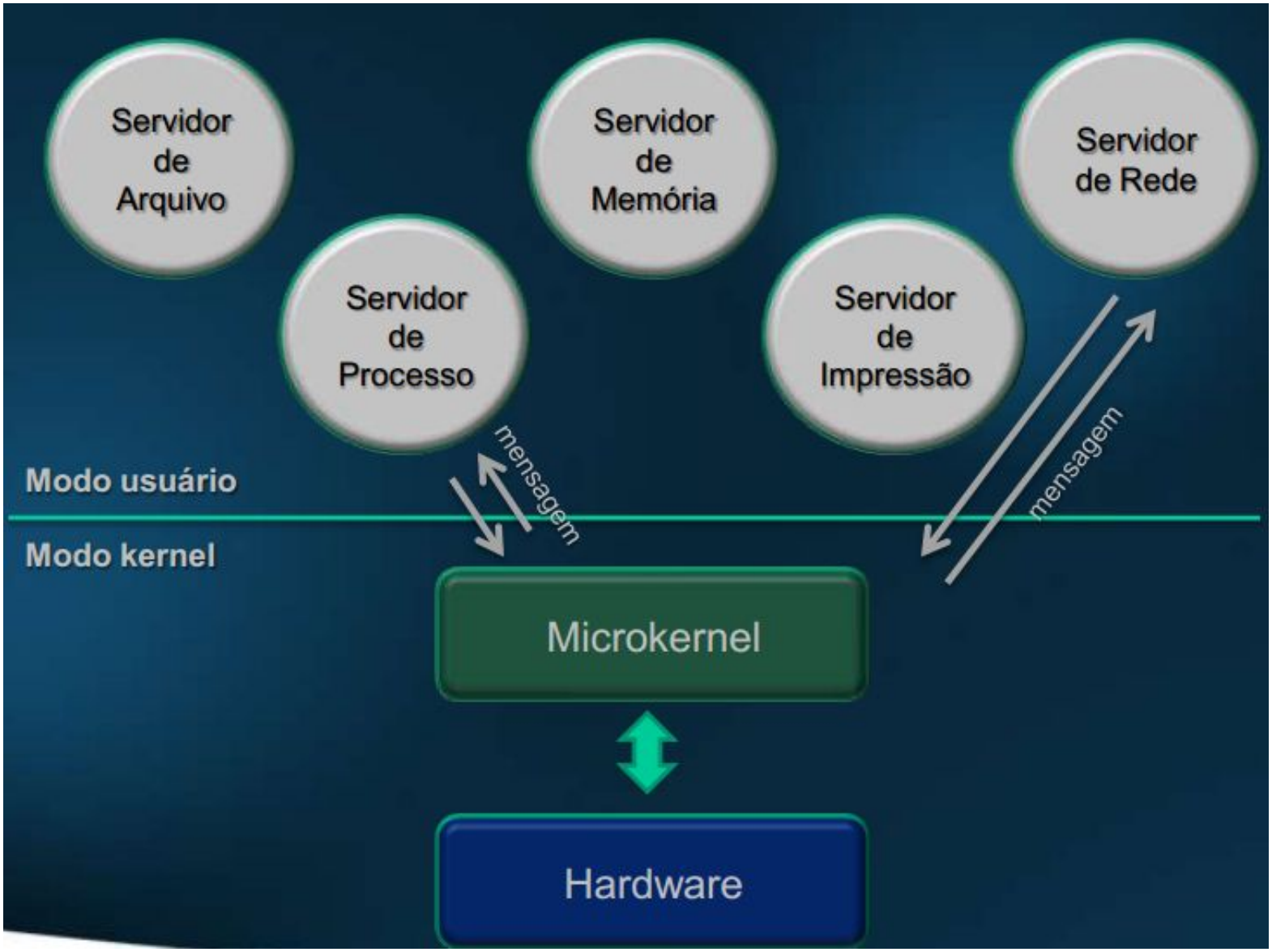
Exonúcleos

- Tarefa
  - Alocar recursos às maquinas virtuais
  - Verificar as tentativas de usar esse recursos
  - Assegurar que nenhuma esteja tentando usar recursos das outras
- Máquina virtual
  - Pensa que tem todos os recursos só para ela

MicroKernel

- É fruto do resultado da tendência de um núcleo de S.O. se tornar o **menor possível**;
- Os serviços são disponibilizados em **processos**;
- Cada processo é responsável em gerenciar um conjunto específico de funções como gerência de memória, gerência de arquivos, gerência de processos etc.;
- Dois tipos: **processo cliente** e **processo servidor**;
- A principal função do núcleo é **gerenciar a comunicação** entre esses processos.

MicroKernel



MicroKernel

- Prós
  - **Maior proteção do núcleo**: todos os processos são executados em modo usuário;
  - **Alta disponibilidade**: se um servidor falhar, o sistema não ficará altamente comprometido;
  - **Maior eficiência**: a comunicação entre serviços poderá ser realizada entre vários processadores ou até mesmo várias máquinas distribuídas;
  - **Melhor confiabilidade e escalabilidade**;
- Contras
  - Grande complexidade para sua implementação
  - Menor desempenho devido à necessidade de mudança de modo de acesso

Referências

- Andrew S. Tanenbaum. “Sistemas Operacionais Modernos”. 3ª Edição, Prentice Hall, 2010.

- Francis B. Machado e Luiz P. Maia. “Arquitetura de Sistemas Operacionais”. 3ª. Edição. LTC, 2004.

UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS DE RUSSAS

Rus0013 - Sistemas Operacionais

Aula 02: Chamadas ao Sistema & Estruturas de Sistemas Operacionais

Professor Pablo Soares

2022.2