UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Aula 02: Introdução a Sistemas Operacionais

Tipos de SO e Arquiteturas

Prof. Rodrigo Campiolo
Prof. Rogério A. Gonçalves[1]

[1]Universidade Tecnológica Federal do Paraná (UTFPR)
Departamento de Computação (DACOM)
Campo Mourão, Paraná, Brasil

Ciência da Computação

BCC34G - Sistemas Operacionais

Sumário

Introdução

Classificação de SO

- Considera características como tamanho, finalidade, recursos, limitações, entre outros.
- Um sistema operacional pode se enquadrar em mais de um tipo.

Tipos de Sistemas Operacionais I

SO Monoprogramados (monotarefas)

- Um processador, um processo e um usuário.
- CPU ociosa durante E/S.
- Simples implementação.
- Exemplo: MS-DOS (antes 4.0)

Tipos de Sistemas Operacionais II

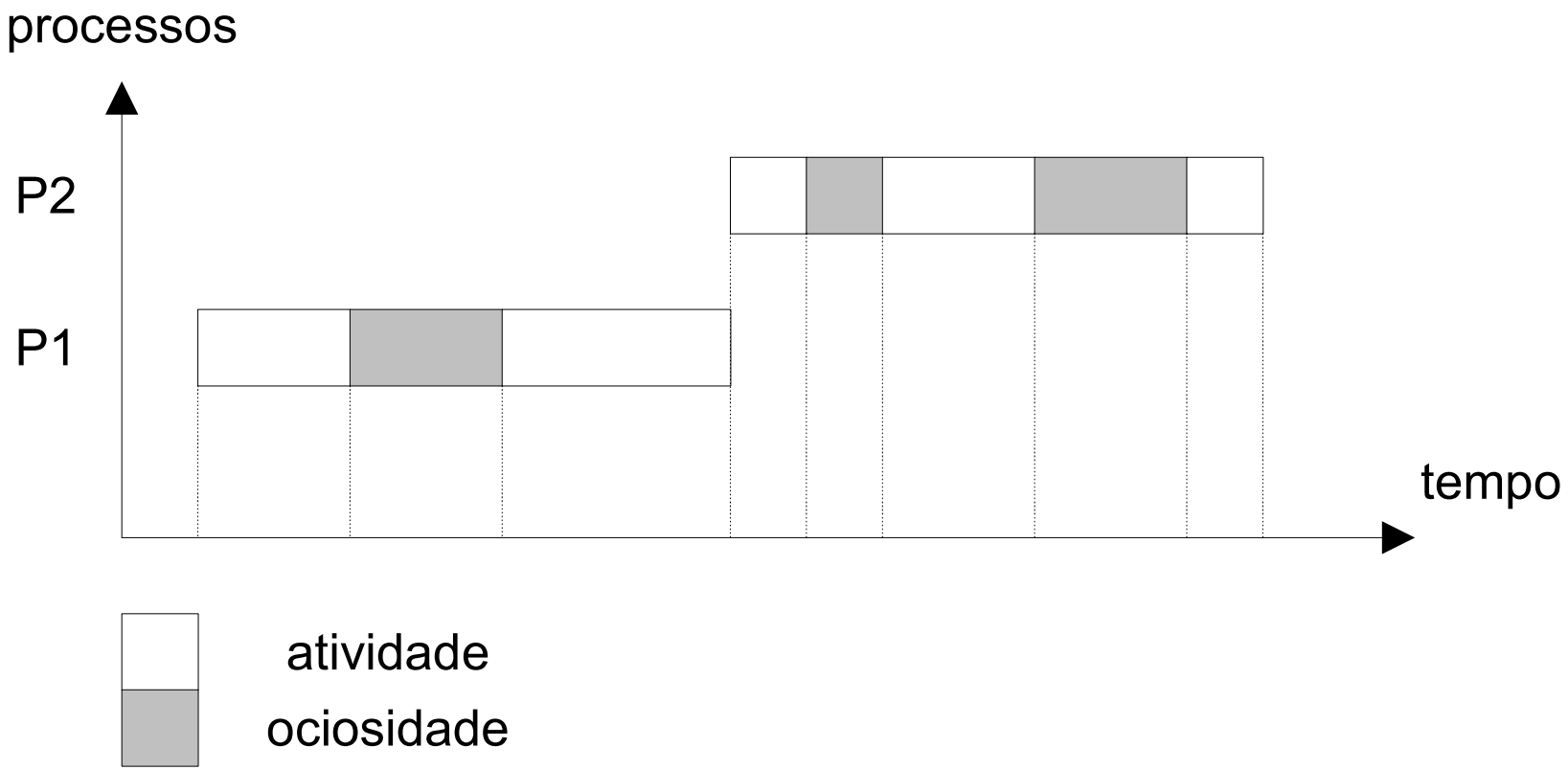

Figura 1: Desvantagem dos SO monotarefas.

Tipos de Sistemas Operacionais III

SO em Lotes (*batch*)

- Programas em fila para execução.
- Não interação com os usuários.
- Alto grau de uso do sistema.
- Exemplo: OS360

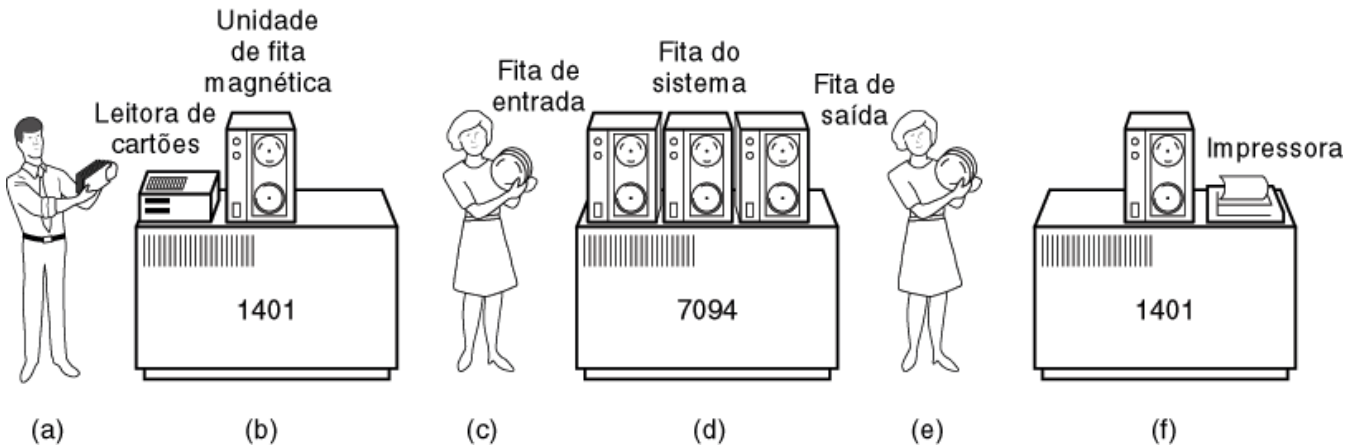

Figura 2: Sistema antigo de lote [2].

Tipos de Sistemas Operacionais IV

SO Multiprogramados (Multitarefas)

- Um ou mais processadores, processos e usuários.
- Não fica ocioso durante E/S.
- Implementação complexa.
- Conceito de compartilhamento de tempo (*time slice*)
- Exemplos: Unix, Linux, Windows NT.

Tipos de Sistemas Operacionais V

SO Multiusuários

- Suporte a vários usuários.
- Controle de acesso a recursos segundo usuários e permissões.
- Exemplos: Unix, Linux, Windows NT.

Tipos de Sistemas Operacionais VI

SO Desktop

- Suporte a usuários domésticos e corporativos.
- Ambiente gráfico para interatividade e facilidade de acesso a rede.
- Exemplos: Windows 7, Linux Desktop, MacOS.

SO Servidores

- Gestão de grande quantidade de recursos e provimento de serviços.
- Exemplos: Windows Server, Solaris, Linux.

Tipos de Sistemas Operacionais VII

SO Embarcados

- Voltado para dispositivos com recursos limitados.
- Usados em dispositivos industriais, domésticos, entre outros.
- Exemplos: LynxOS.

SO Portáteis

- Voltado para dispositivos com recursos limitados e portáteis.
- Usados em smartphones e tablets.
- Exemplos: Android, Symbian, iOS.

Tipos de Sistemas Operacionais VIII

SO Tempo Real

- Tempo de resposta rígido.
- Menos tempo compartilhado e mais prioridade.
- Processos ativados por sensores.
- Aplicações: usinas, refinarias, tráfego aéreo, entre outros.
- Duas classificações: *soft real-time systems* e *hard real-time systems*.
- Exemplos: o QNX, RT-Linux e VxWorks.

Multiprogramação I

Conceitos

- Interrupções e Exceções
- Buffering
- Spooling

Interrupções e Exceções

- **Interrupções de Hardware**: tratam eventos assíncronos externos ao processador.
- **Interrupções de Software (Trap)**: tratam eventos síncronos solicitados por aplicações.
- **Exceções**: tratam eventos síncronos internos ao processador.

Multiprogramação III

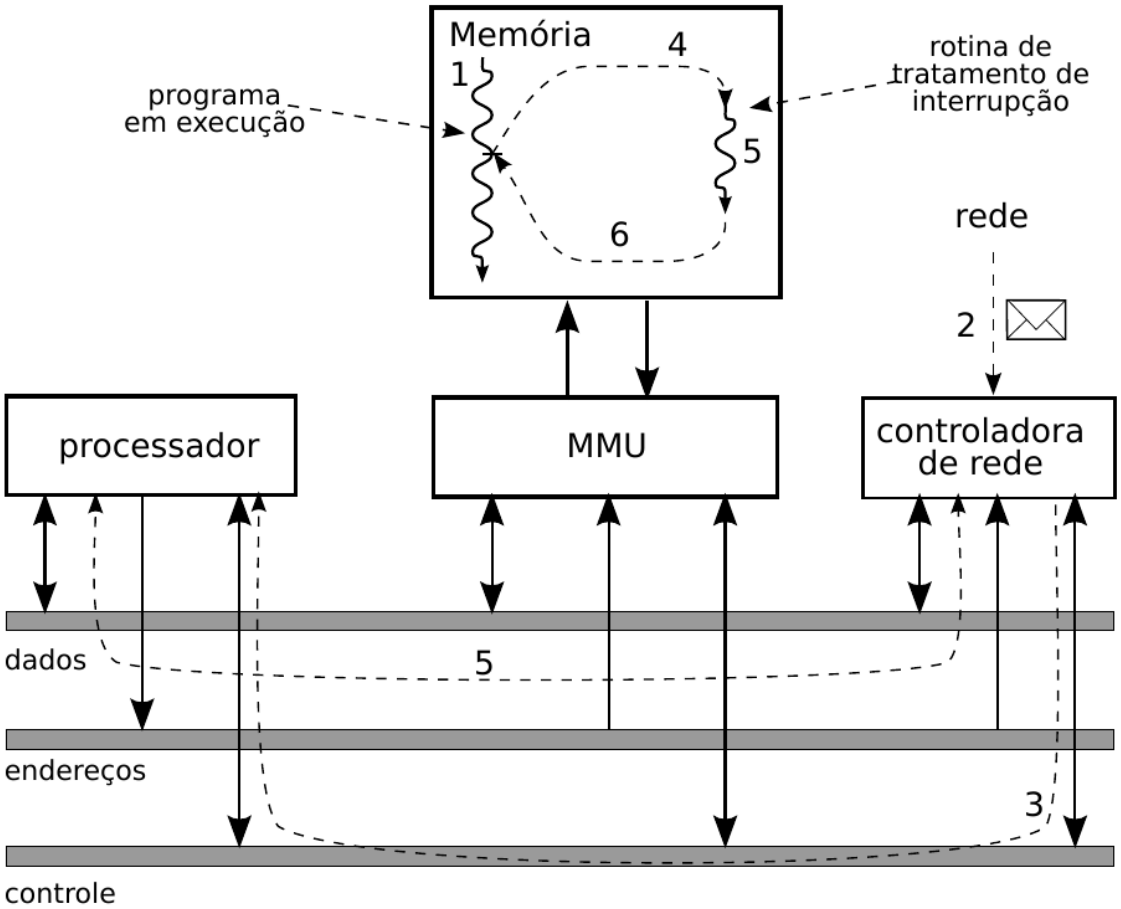

Figura 3: Tratamento de interrupção gerada por hardware [1].

Multiprogramação IV

Tratamento de interrupção de hardware

1. processador executando um programa;
2. pacote recebido pela placa Ethernet;
3. controlador Ethernet envia uma IRQ ao processador;
4. processador executa a rotina de tratamento;
5. rotina transfere os dados para a memória;
6. rotina finalizada e processamento do programa é retomado.

Multiprogramação V

Exemplo: Interrupção do relógio

- O relógio (*clock*) interrompe a CPU em intervalos de tempo fixo.
- Medidos em **ticks**. No Linux é geralmente 10ms.
- A rotina de interrupção de relógio (Clock Interrupt Handler) é executada para tratar uma interrupção de relógio.
- Funcionalidades básicas:
  - reiniciar o relógio;
  - atualizar estatísticas de CPU;
  - funções de escalonamento;
  - manipular sinais;
  - atualizar a hora e temporizadores;
  - alarmes (notificar os processos após um período de tempo).

Multiprogramação VI

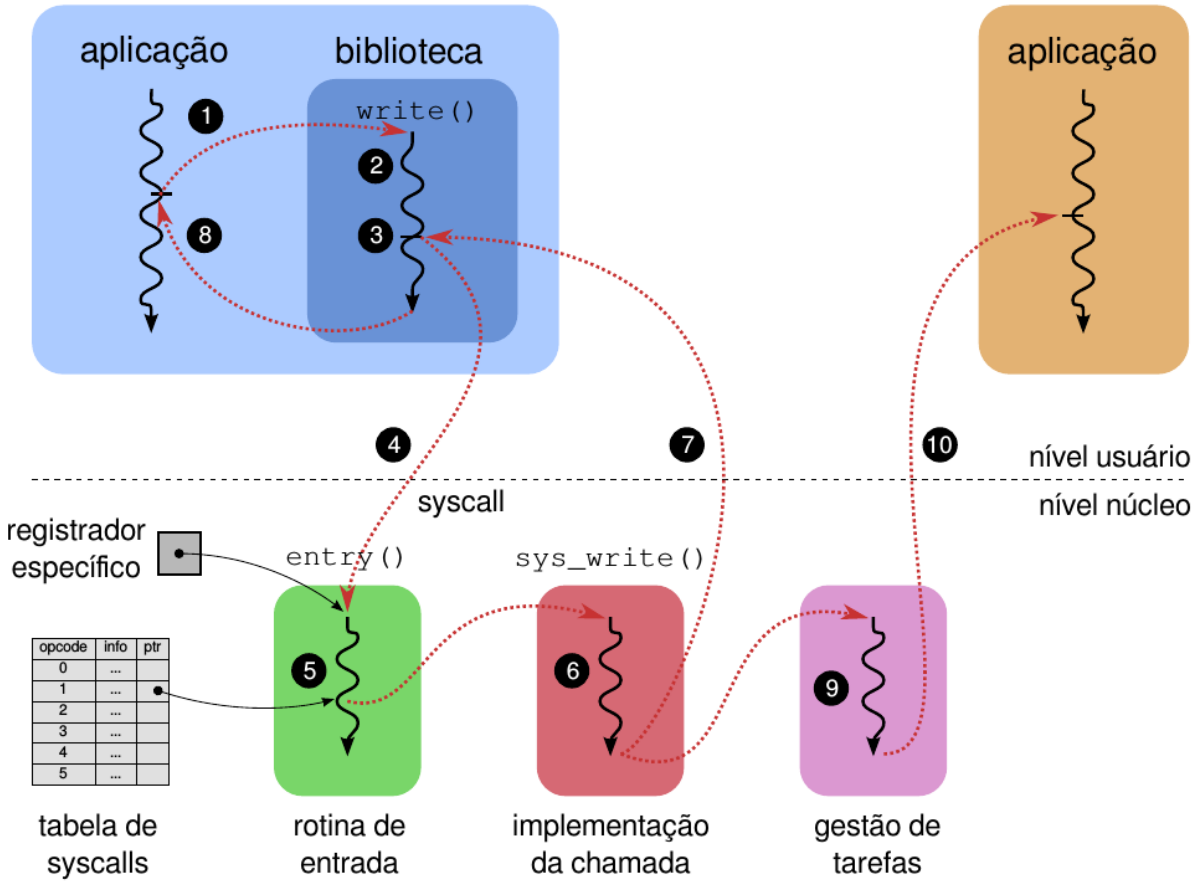

Figura 4: Execução de uma chamada de sistema (interrupção de software) [1].

Multiprogramação VII

Tratamento de interrupção por software

1. aplicação invoca a função `write()` da biblioteca de sistema;
2. a função `write()` preenche registradores CPU com os parâmetros e escreve o *opcode* da *syscall* no registrador AX.
3. função `write()` invoca uma chamada de sistema (*syscall*);
4. processador alterna para modo núcleo e executa a rotina `entry()`;
5. a rotina recebe o *opcode* da operação via registrador AX, consulta a tabela de chamadas de sistema e invoca a operação requisitada;
6. a operação `sys_write()` obtém, valida e processa os parâmetros;
7. ao final, armazena o retorno e devolve o controle para `write()`;
8. a função `write()` devolve controle para aplicação;
9. operação não concluída, gerência de processos é acionada;
10. a gerência de processo bloqueia e escalona outra aplicação.

Multiprogramação VIII

Exemplos: Chamadas de sistema no Linux

- **read**: lê a partir de um descritor de arquivo.
- **write**: escreve para um descritor de arquivo.
- **mmap**: mapeia arquivos e dispositivos na memória.
- **fstat**: devolve informações sobre um arquivo.
- **mprotect**: modifica a proteção de uma região de memória.
- **access**: verifica se o processo pode acessar um arquivo.
- **open**: abre ou cria um arquivo.
- **close**: fecha um descritor de arquivo.
- **brk**: modifica o tamanho do segmento de dados.

* `https://github.com/torvalds/linux/blob/master/arch/x86/entry/syscalls/syscall_64.tbl`

Multiprogramação IX

Exceções

- As exceções são eventos gerados pelo processador devido a condições específicas durante a execução de um fluxo de instruções.
- Exemplos: divisão por zero, código inválido, falta de página, dispositivo não disponível.
- Esses eventos são tratados por uma rotina de tratamento de exceção.
- As exceções e interrupções de software são eventos síncronos e as interrupções de hardware são assíncronos.

Multiprogramação X

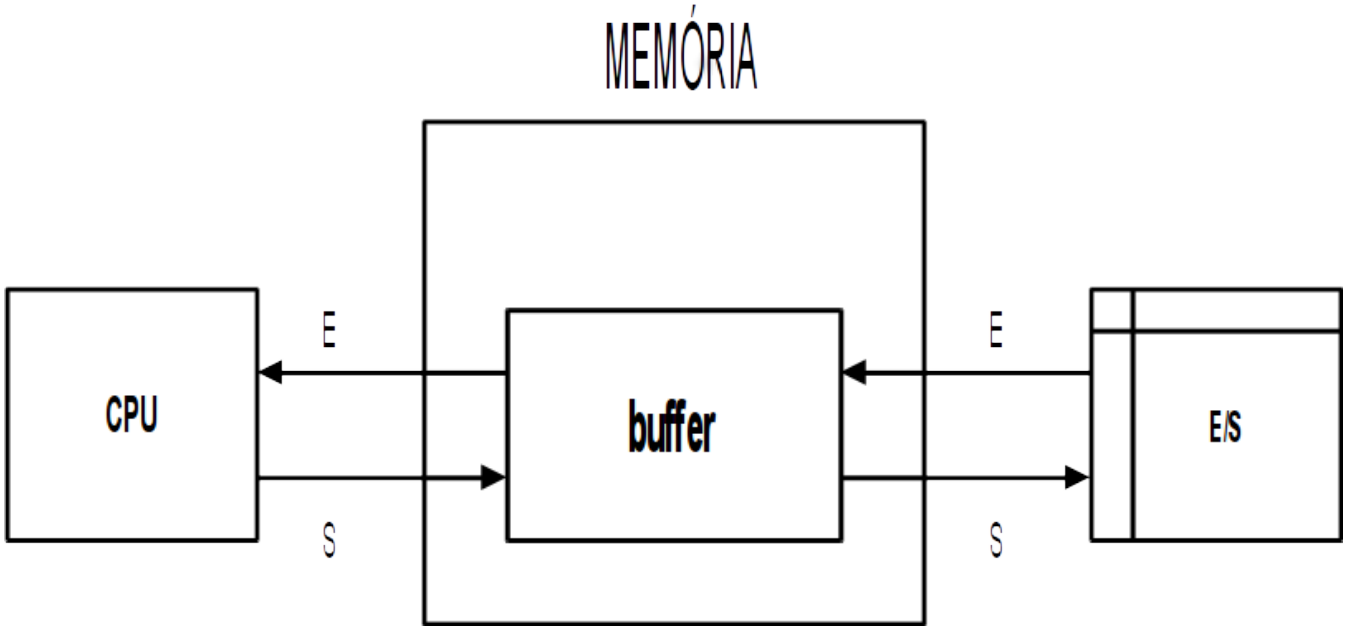

Figura 5: Processo de *buffering*.

Spooling

Simultaneous Peripherical Operating On Line

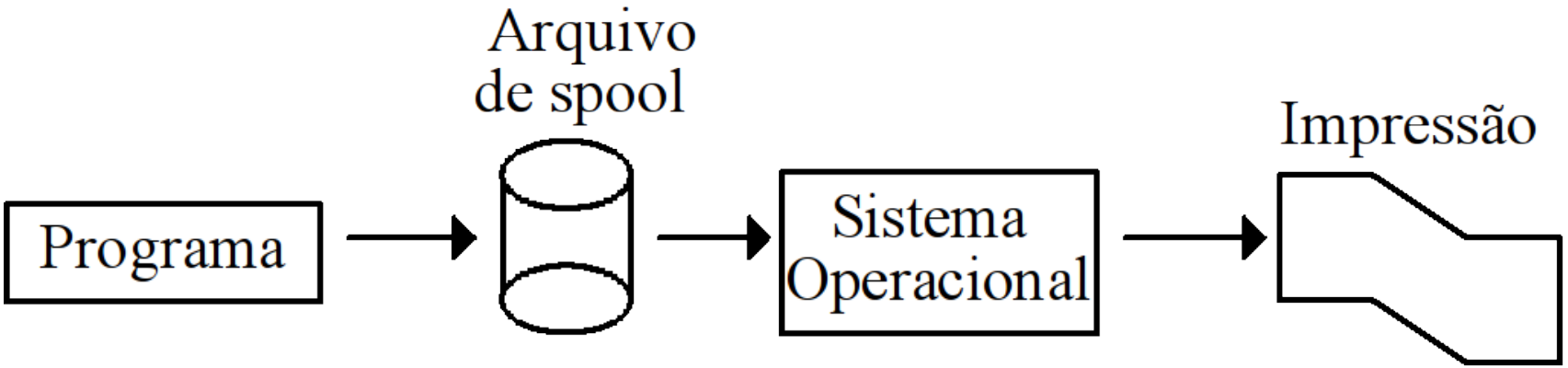

Figura 6: Processo de *spooling*.

Arquiteturas de Sistemas Operacionais I

- Os atuais Sistemas Operacionais tendem a ser complexos e oferecem vários serviços.
- Suportam uma variedade de recursos de hardware e software.
- A arquitetura do sistema operacional ajuda a gerenciar essa complexidade:
  (a) organizando componentes de sistema operacional e
  (b) como esses componentes são executados.

Arquiteturas de Sistemas Operacionais II

Principais arquiteturas

- **Sistemas monolíticos** (ex: FreeBSD, Linux, MS-DOS).
- **Sistemas micronúcleo - microkernel** (ex: Minix, GNU Mach).
- Sistemas em camadas (ex: MULTICS).
- **Sistemas híbridos** (ex: Windows NT, Mac OS).
- **Máquinas virtuais** (ex: Xen, KVM, Virtual Box, VMWare, JVM).
- **Contêineres** (ex: Docker e Kubernetes).
- Sistemas exonúcleo (ex: Nemesis - ideia de LibOS).
- Sistemas uninúcleo (ex: MirageOs, OSv).

Arquiteturas de Sistemas Operacionais III

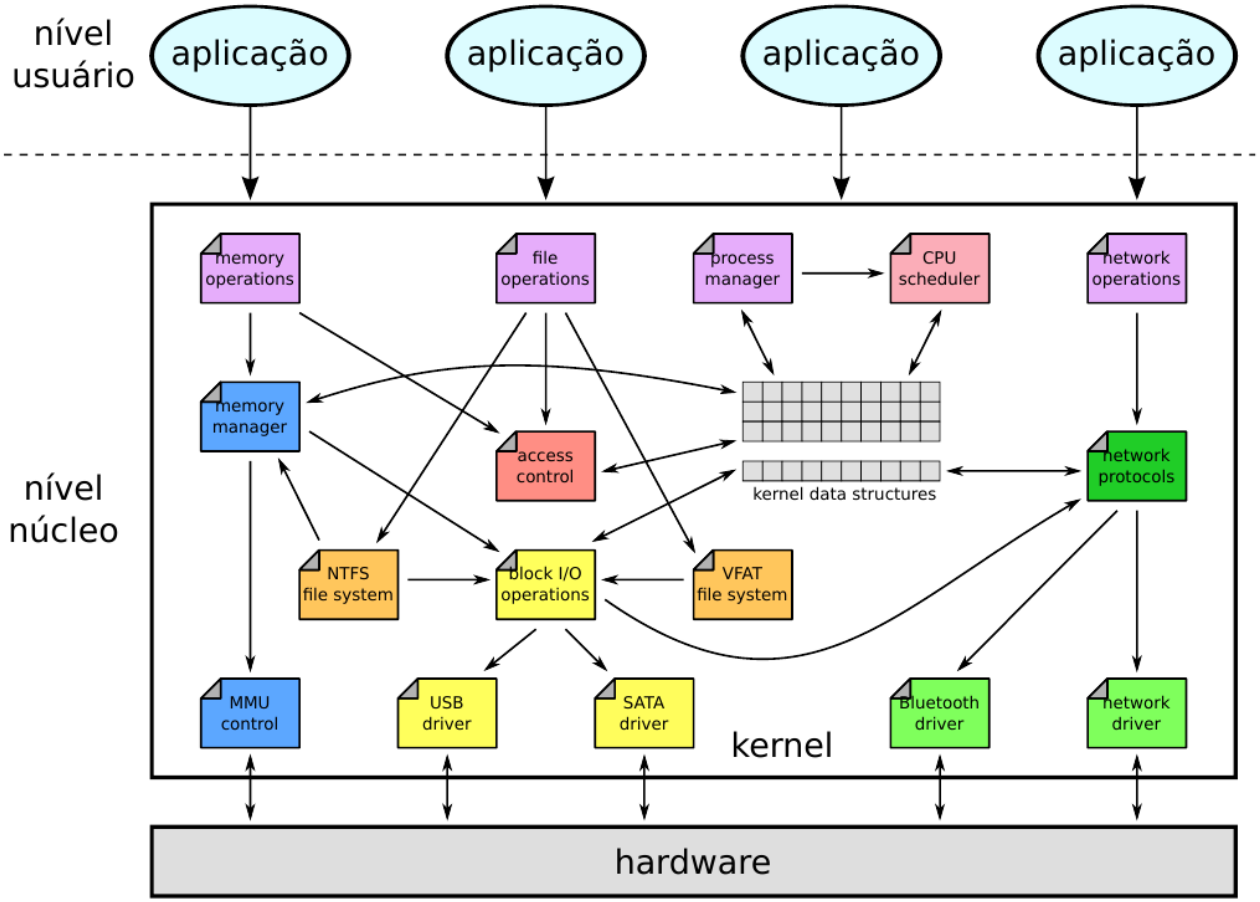

Figura 7: Visão geral da arquitetura monolítica [1].

Arquiteturas de Sistemas Operacionais IV

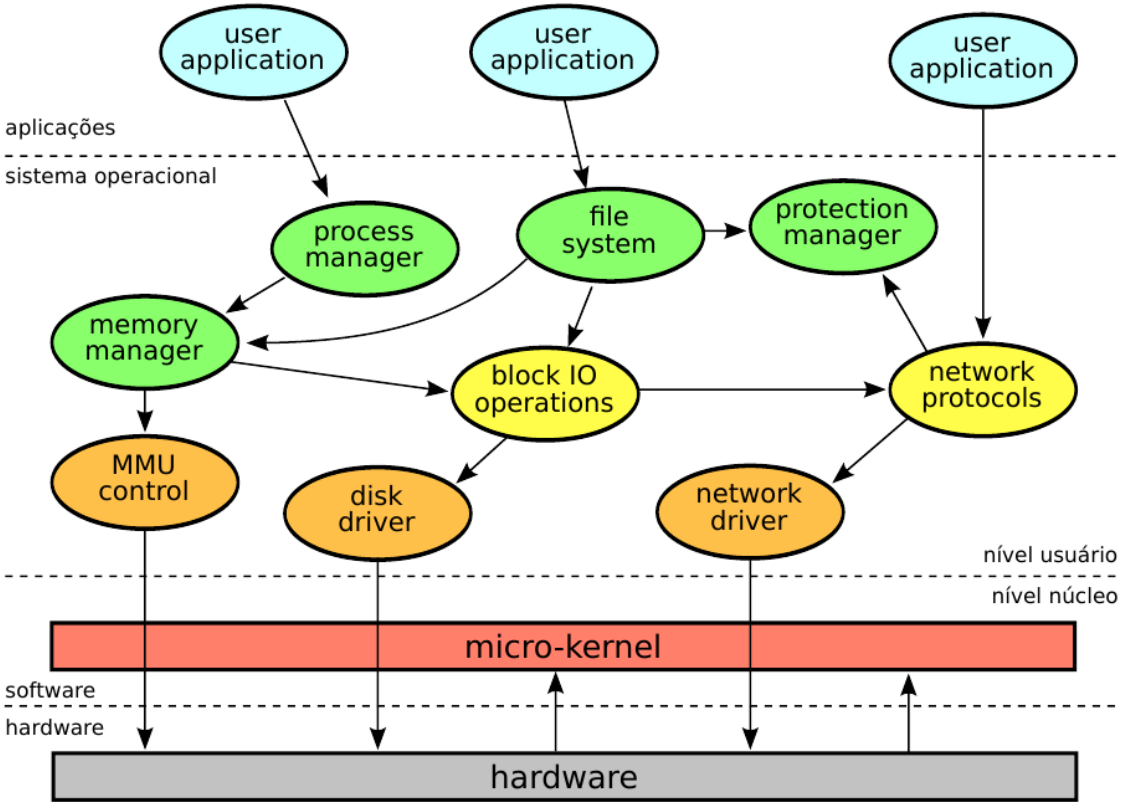

Figura 8: Visão geral da arquitetura micronúcleo [1].

Arquiteturas de Sistemas Operacionais V

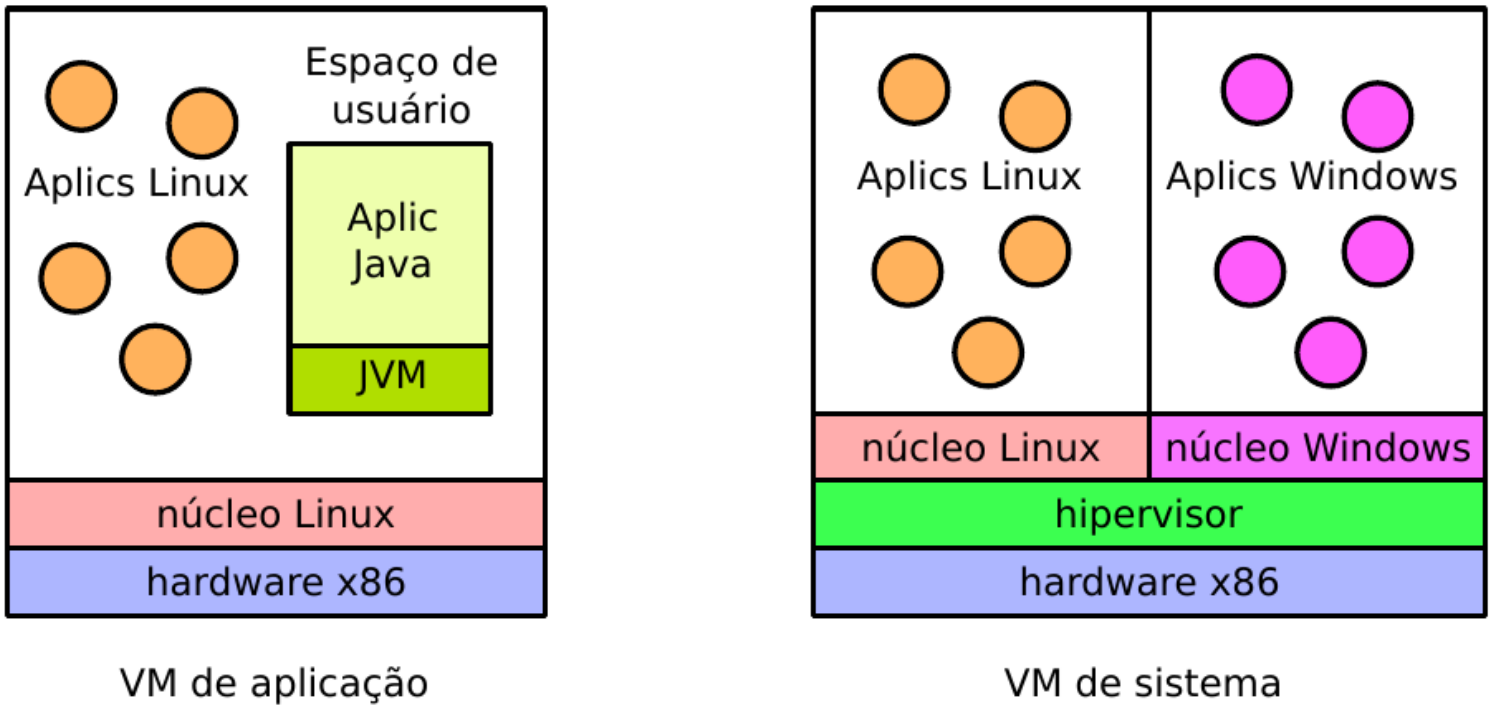

Figura 9: Máquinas virtuais de aplicação e de sistema [1].

Arquiteturas de Sistemas Operacionais VI

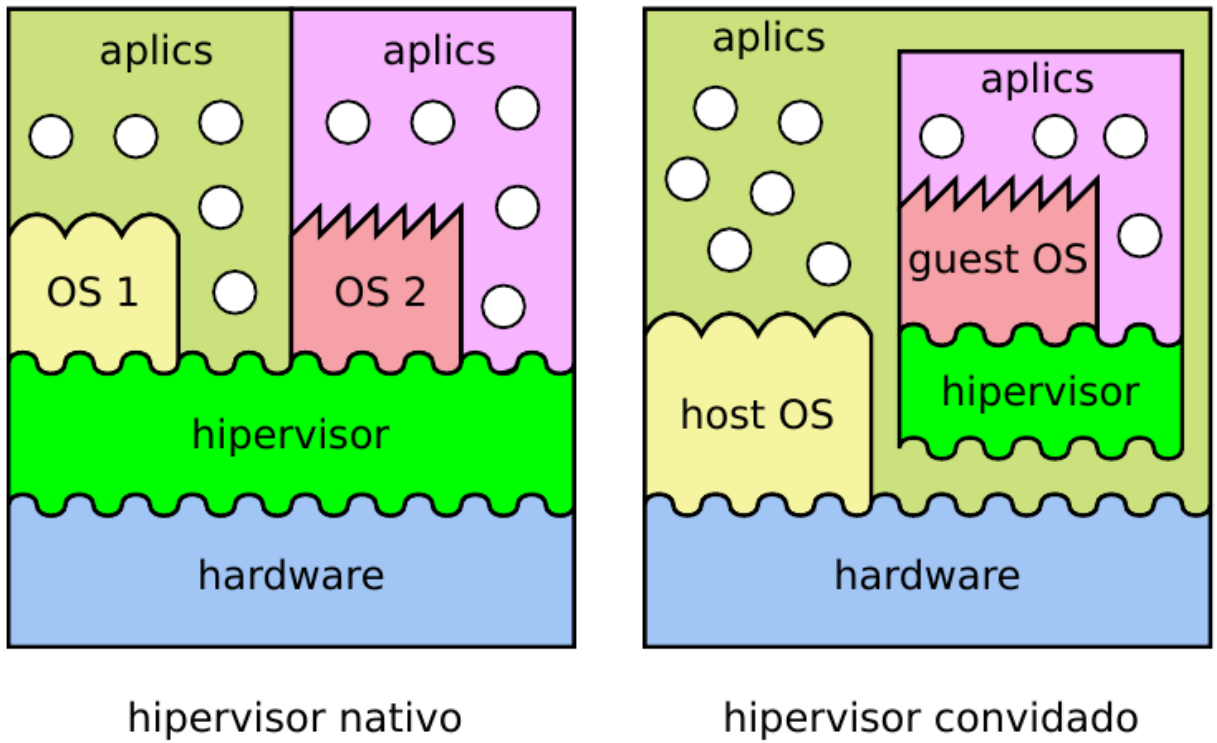

Figura 10: Arquitetura de máquinas virtuais de sistema [1].

Arquiteturas de Sistemas Operacionais VII

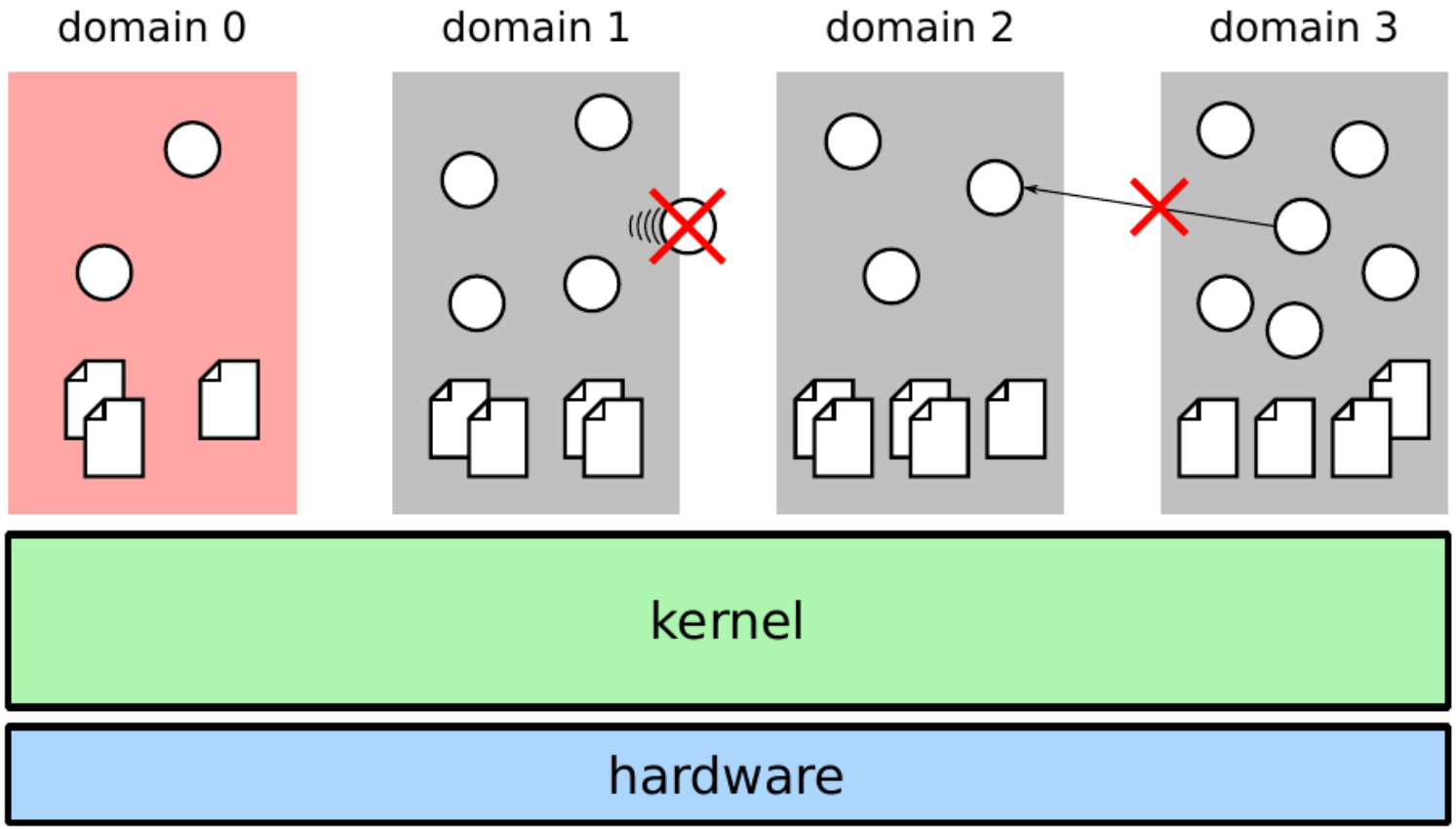

Figura 11: Sistema de contêineres [1].

Arquiteturas de Sistemas Operacionais VIII

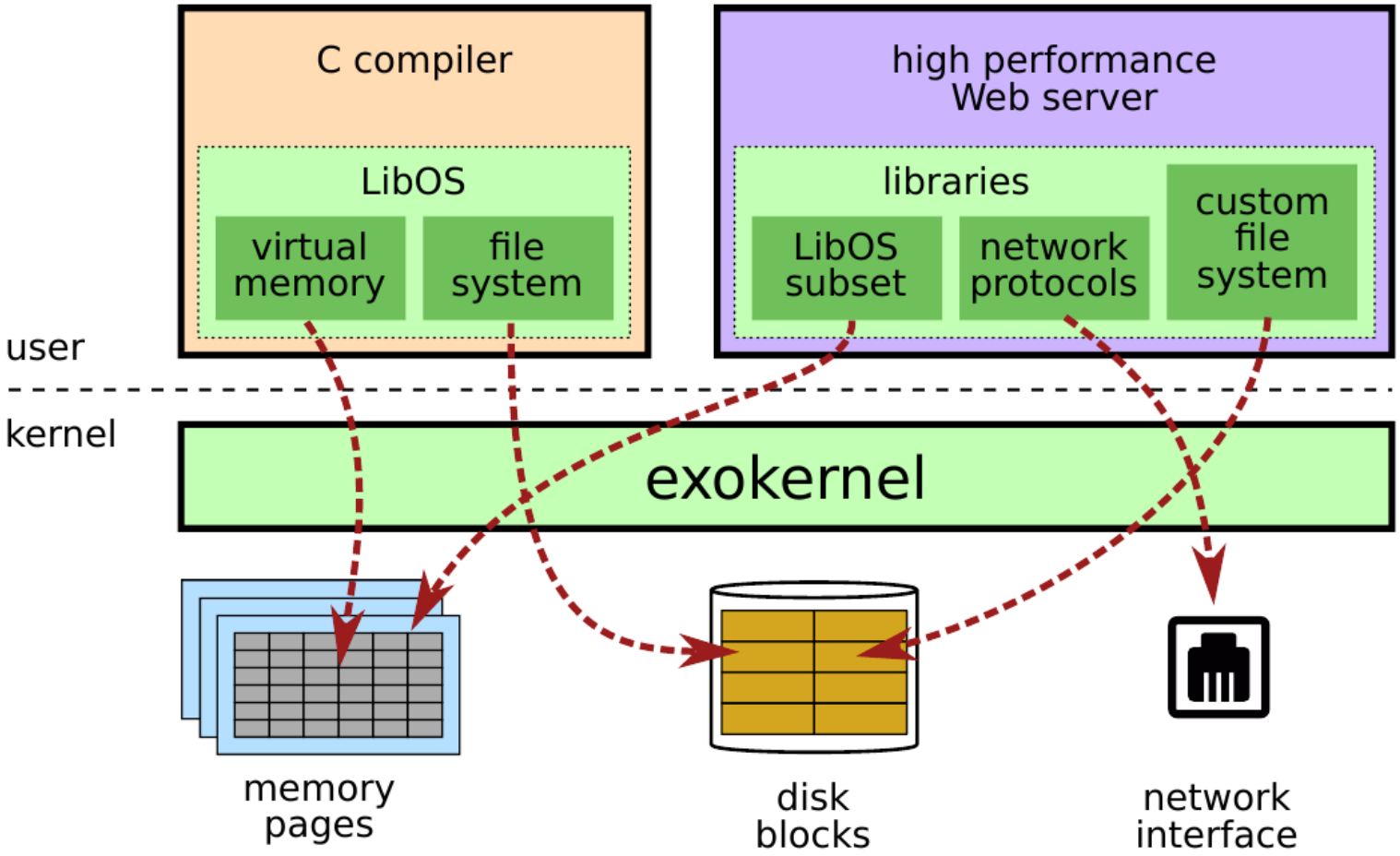

Figura 12: Sistema exonúcleo [1].

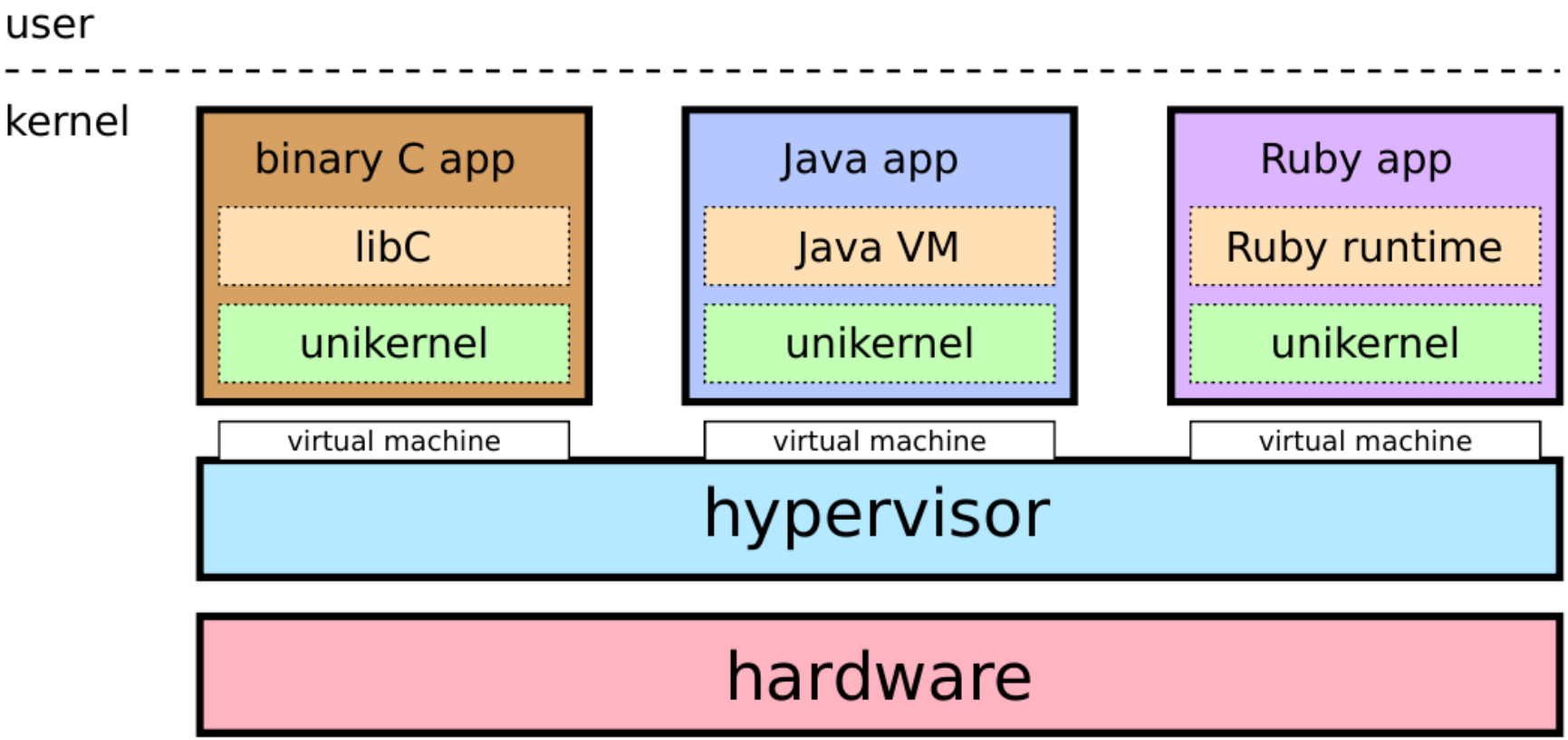

Figura 13: Sistema uninúcleo [1].

Atividades

1. Faça a leitura dos capítulos 1, 2 e 3 do livro do Maziero [1].
   Sugestão: faça anotações para fixar melhor o conteúdo.
2. Resolva a lista de exercícios L01 (Moodle).
3. Execute e analise os exemplos sobre Chamadas de Sistema - Syscalls (Moodle).
4. Compile o núcleo (kernel) do Linux e do FreeBSD (Moodle).

Referências I

[1] Maziero, C. A. (2019). *Sistemas operacionais: conceitos e mecanismos*. online. Disponível em http://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php?media=so:so-livro.pdf.

[2] Tanenbaum, A. S. and Bos, H. (2016). *Sistemas operacionais modernos*. Pearson Education do Brasil, 4 edition.