INSTITUTO FEDERAL
RIO GRANDE DO SUL

Sistemas Operacionais

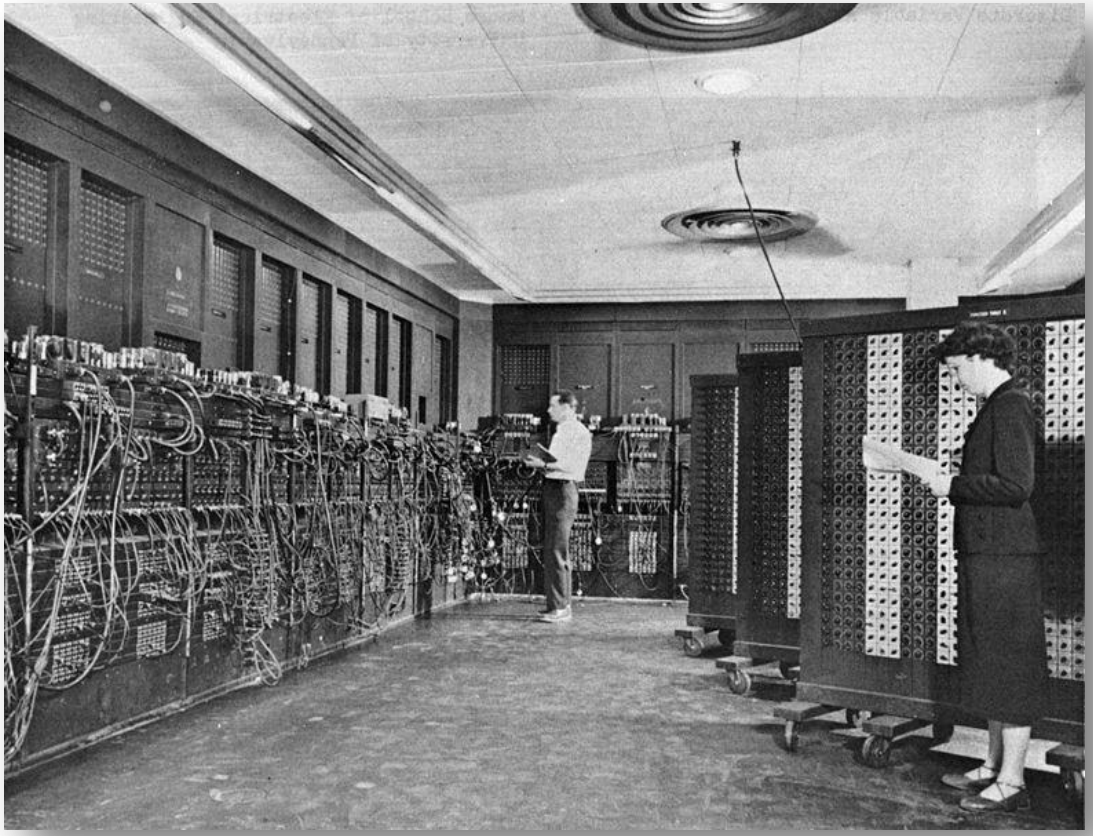
- Tipos de Sistemas Operacionais
- Funcionalidades
- Arquitetura:
    - Monolíticos
    - Em camadas
    - Micro-núcleo
    - Máquinas virtuais

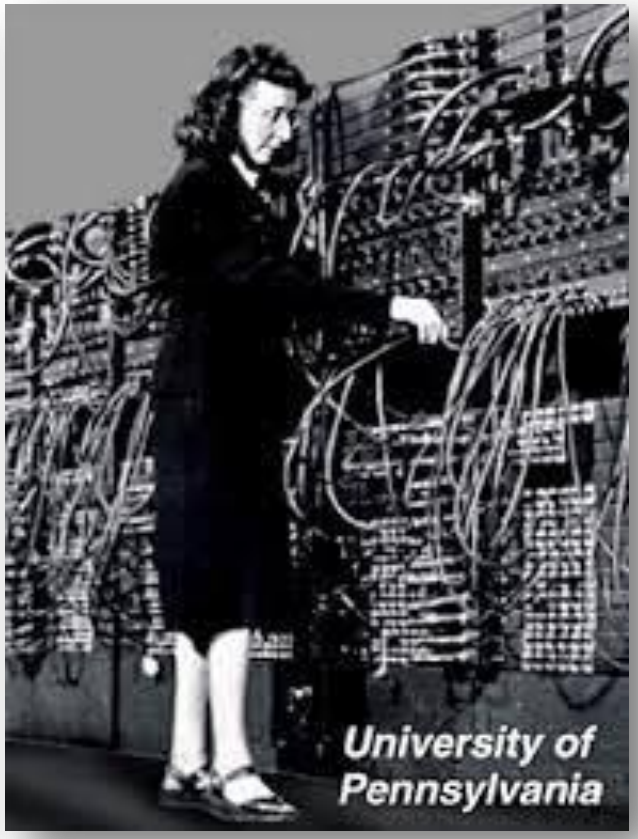
Apresentações Adaptadas do Material do Prof. Marcelo Paravisi

https://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php?media=socm:socm-livro.pdf

Historia do SO

- Os primeiros sistemas de computação, no final dos anos 1940, não possuíam sistema operacional: as aplicações eram executadas diretamente sobre o hardware.





INSTITUTO FEDERAL
RIO GRANDE DO SUL

Historia do SO

- **Anos 40:**
  - cada programa executava sozinho e tinha total controle do computador.
  - A carga do programa em memória, a varredura dos periféricos de entrada para busca de dados, a computação propriamente dita e o envio dos resultados para os periférico de saída, byte a byte, tudo devia ser programado detalhadamente pelo desenvolvedor da aplicação.

- **Anos 50:**
  - Os sistemas de computação fornecem “bibliotecas de sistema” (system libraries) que encapsulam o acesso aos periféricos, para facilitar a programação de aplicações.
  - Algumas vezes um programa “monitor” (system monitor) auxilia a carga e descarga de aplicações e/ou dados entre a memória e periféricos (geralmente leitoras de cartão perfurado, fitas magnéticas e impressoras de caracteres).

INSTITUTO FEDERAL
RIO GRANDE DO SUL

Historia do SO

- 1961:
  - o grupo do pesquisador Fernando Corbató, do MIT, anuncia o desenvolvimento do CTSS – Compatible Time-Sharing System, o primeiro sistema operacional com compartilhamento de tempo.
- 1965:
  - a IBM lança o OS/360, um sistema operacional avançado, com compartilhamento de tempo e excelente suporte a discos.
- 1965:
  - um projeto conjunto entre MIT, GE e Bell Labs define o sistema operacional Multics, cujas ideias inovadoras irão influenciar novos sistemas durante décadas.
- 1969:
  - Ken Thompson e Dennis Ritchie, pesquisadores dos Bell Labs, criam a primeira versão do UNIX.
- 1981:
  - a Microsoft lança o MS-DOS, um sistema operacional comprado da empresa Seattle Computer Products em 1980.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

Historia do SO

- 1984:
  - a Apple lança o sistema operacional Mac OS 1.0 para os computadores da linha Macintosh, o primeiro a ter uma interface gráfica totalmente incorporada ao sistema.
- 1985:
  - primeira tentativa da Microsoft no campo dos sistemas operacionais com interface gráfica, através do MS-Windows 1.0.
- 1987:
  - Andrew Tanenbaum, um professor de computação holandês, desenvolve um sistema operacional didático simplificado, mas respeitando a API do UNIX, que foi batizado como Minix.
- 1987:
  - IBM e Microsoft apresentam a primeira versão do OS/2, um sistema multitarefa destinado a substituir o MS-DOS e o Windows. Mais tarde, as duas empresas rompem a parceria; a IBM continua no OS/2 e a Microsoft investe no ambiente Windows.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

Historia do SO

- 1991:
  - Linus Torvalds, um estudante de graduação finlandês, inicia o desenvolvimento do Linux, lançando na rede Usenet o núcleo 0.01, logo abraçado por centenas de programadores ao redor do mundo.
- 1993:
  - a Microsoft lança o Windows NT, o primeiro sistema 32 bits da empresa, que contava com uma arquitetura interna inovadora.
- 1993:
  - Lançamento dos UNIX de código aberto FreeBSD e NetBSD.
- 1993:
  - A Apple lança o Newton OS, considerado o primeiro sistema operacional móvel, com gestão de energia e suporte para tela de toque.
- 1995:
  - Microsoft lança o Windows 95.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

Historia do SO

- 1999:
  - a empresa VMWare lança um ambiente de virtualização para sistemas operacionais de mercado.
- 2001:
  - a Apple lança o MacOS X, um sistema operacional com arquitetura distinta de suas versões anteriores, derivada da família UNIX BSD.
- 2005:
  - lançado o Minix 3, um sistema operacional micro-núcleo para aplicações embarcadas. O Minix 3 faz parte do firmware dos processadores Intel mais recentes.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

Historia do SO

- 2007:
  - lançamento do iPhone e seu sistema operacional iOS, derivado do sistema operacional Darwin (Darwin é um Kernel de código aberto lançado pela Apple Inc. em 2000).
- 2007:
  - lançamento do Android, um sistema operacional baseado no núcleo Linux para dispositivos móveis. 2010: Windows Phone, SO para celulares pela Microsoft.
- 2015:
  - Microsoft lança o Windows 10 e em 2021 lança o Windows 11.

O que é um Sistemas Operacional?

INSTITUTO FEDERAL
RIO GRANDE DO SUL

Conceito de Sistemas Operacionais

- Os circuitos são complexos, acessados através de interfaces de baixo nível e muitas vezes suas características e seu comportamento dependem da tecnologia usada em sua construção.
  - Exemplo: A forma de acessar um disco rígido IDE de um disco SCSI, ou um SATA.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

Conceito de Sistemas Operacionais

O sistema operacional é uma camada de software que opera entre o hardware e os programas aplicativos voltados ao usuário final.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

Conceito de Sistemas Operacionais

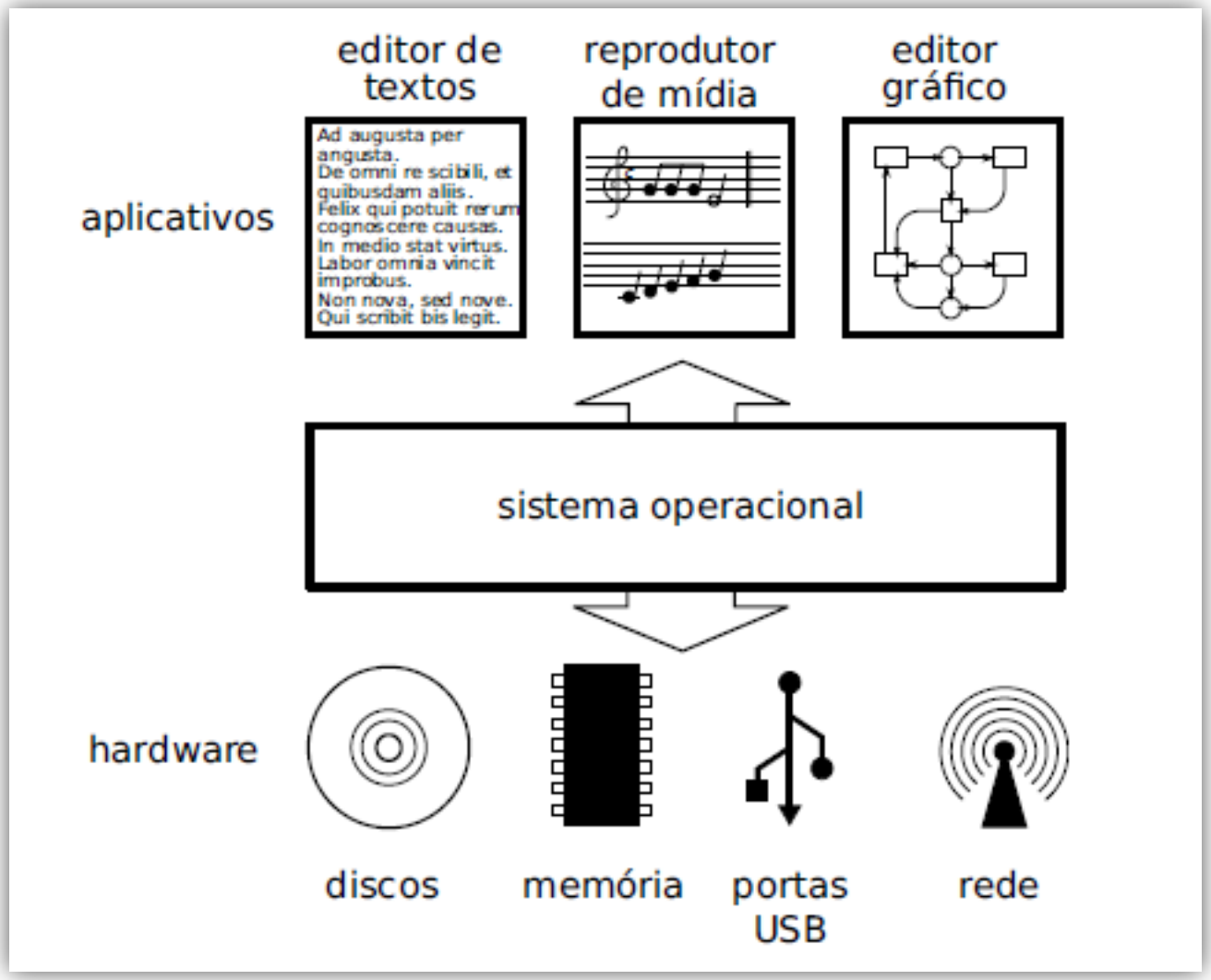
- O sistema operacional é uma estrutura de software ampla, muitas vezes complexa, que incorpora aspectos de baixo nível (como drivers de dispositivos e gerência de memória física) e de alto nível (como programas utilitários e a própria interface gráfica).

- **Oferece uma forma homogênea de acesso aos dispositivos físicos.**

- Exigir que o programador domine como funciona cada dispositivo e programar cada detalhe impossibilitaria a construção de software.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

Conceito de sistemas operacionais



INSTITUTO FEDERAL
RIO GRANDE DO SUL

Como a vida seria sem SO?

ETAPAS PARA ABRIR UM ARQUIVO EM DISQUETE

1. Verificar se os parâmetros informados estão corretos (nome do arquivo, identificador do leitor de disco, buer de leitura, etc.);
2. Verificar se o leitor de disco está disponível;
3. Verificar se o leitor contém um disquete;
4. Ligar o motor do leitor e aguardar atingir a velocidade de rotação correta;
5. Posicionar a cabeça de leitura sobre a trilha onde está a tabela de diretório;
6. Ler a tabela de diretório e localizar o arquivo ou subdiretório desejado;
7. Mover a cabeça de leitura para a posição do bloco inicial do arquivo;
8. Ler o bloco inicial do arquivo e depositá-lo em um buffer de memória.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

ETAPAS PARA ABRIR UM ARQUIVO EM DISQUETE

O PROGRAMADOR DEVE CUIDAR PARA NÃO DESGASTAR O DISQUETE, EQUILIBRANDO O TEMPO QUE O MOTOR FICA LIGADO COM O TEMPO DE LIGAR O MOTOR.

>> OPERAÇÕES READ E WRITE DO DISQUETE EXIGEM 13 PARÂMETROS AGRUPADOS EM 9 BYTES.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

Abstração de recursos

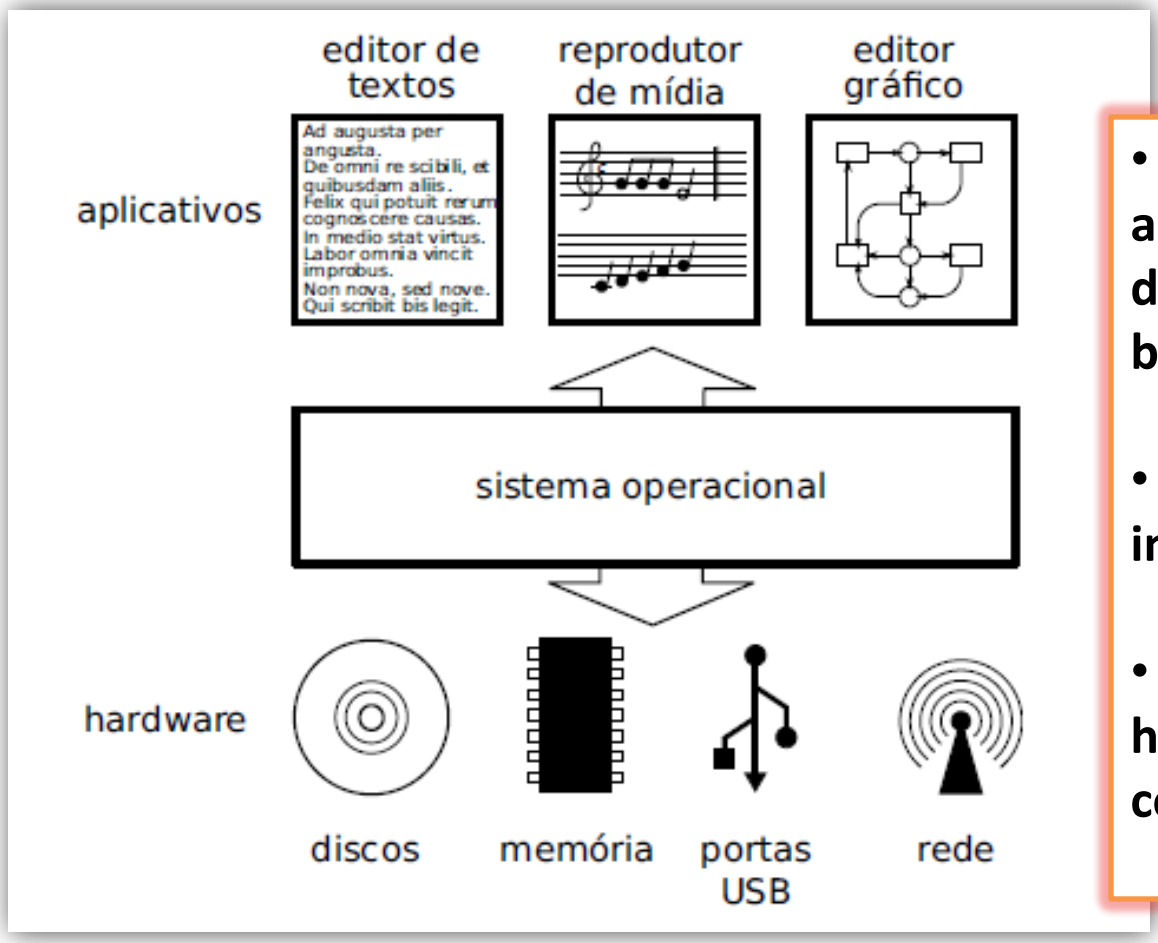
INTERFACES + SIMPLES:

Simplifica a construção de programas aplicativos.
Faz-se uso do conceito de arquivo que implementa uma visão abstrata do disco rígido, acessível através de operações como open, read e close.

Caso tivesse de acessar o disco diretamente, teria de manipular portas de entrada/saída e registradores com comandos para o controlador de disco (sem falar na dificuldade de localizar os dados desejados dentro do disco).

Abstração de recursos

Tarefa do S.O. definir interfaces abstratas para atender os seguintes objetivos:



- **Prover interfaces de acesso aos dispositivos, mais simples de usar que as interfaces de baixo nível.**
- **Tornar os aplicativos independentes do hardware.**
- **Definir interfaces de acesso homogêneas para dispositivos com tecnologias distintas**

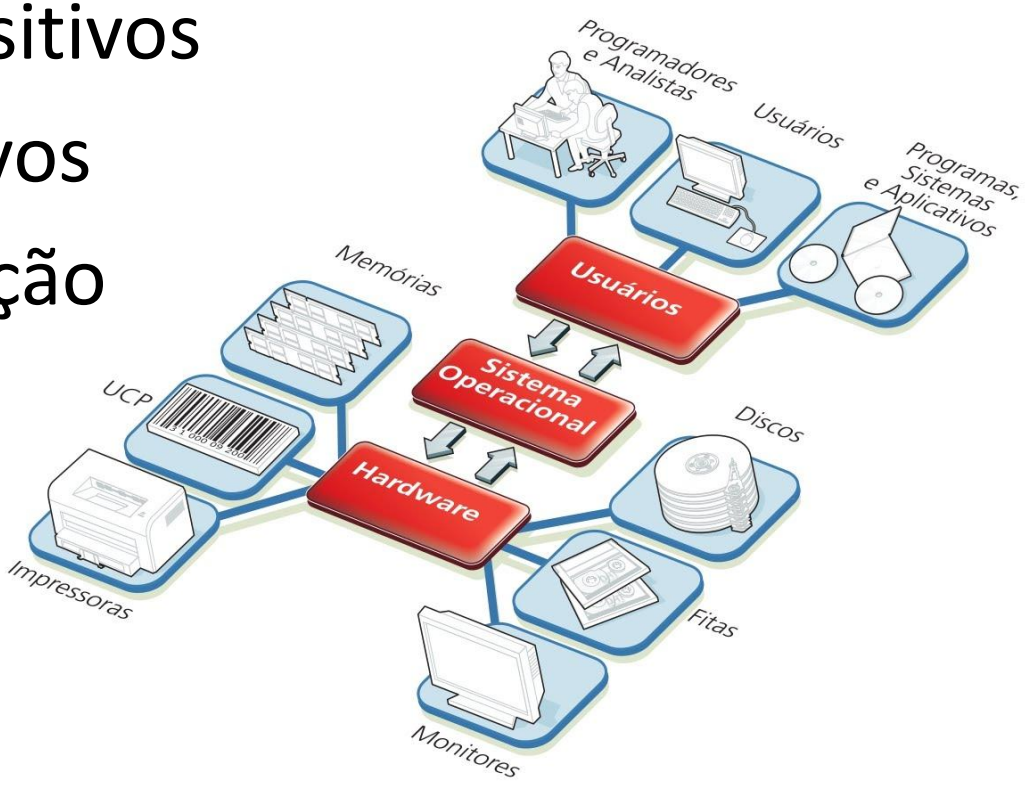
INSTITUTO FEDERAL
RIO GRANDE DO SUL

SO = Gerenciador de Recursos

- **Tarefa do S.O.:**
  - Controlar e ordenar a alocação dos recursos do computador (memória, dispositivos de E/S, processadores).
  - **Exemplo de controle:**
    - 03 programas competindo por um impressora.
    - Proteger a memória de acessos proibidos.

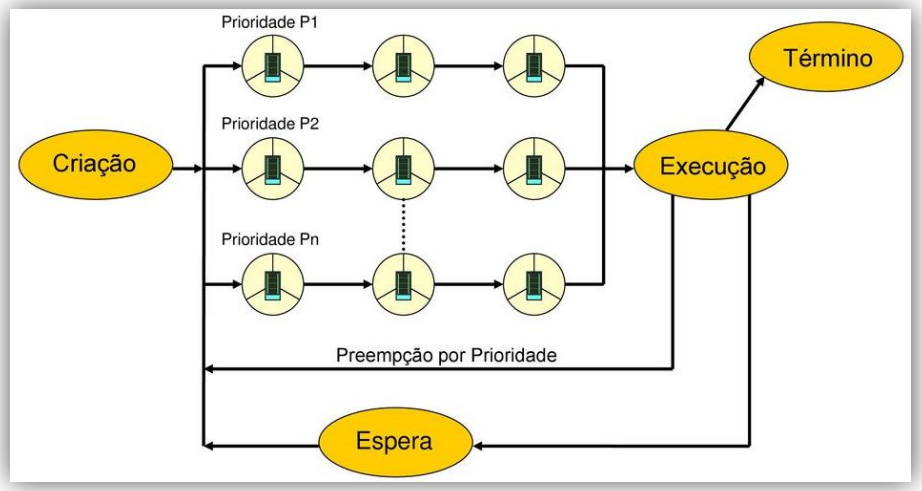
SO = Gerenciador de Recursos

- Gerência de processador
- Gerência de memória
- Gerência de dispositivos
- Gerência de arquivos
- Gerência de proteção



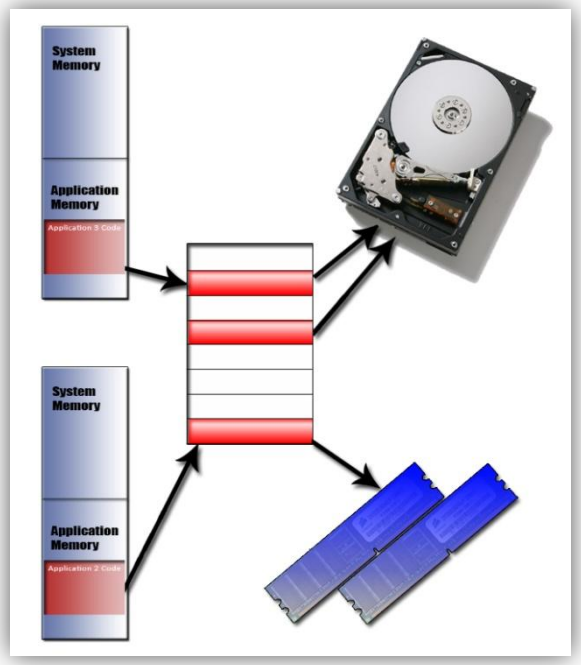
SO = Gerenciador de Recursos

- **Gerência de processador:**
  - Também conhecida como gerência de processos ou de atividades, esta funcionalidade visa distribuir a capacidade de processamento de forma justa entre as aplicações, evitando que uma aplicação monopolize esse recurso e respeitando as prioridades dos usuários.



SO = Gerenciador de Recursos

- **Gerência de memória:**
  - Tem como objetivo fornecer a cada aplicação uma área de memória própria, independente e isolada das demais aplicações e inclusive do núcleo do sistema.



  - O isolamento das áreas de memória das aplicações melhora a estabilidade e segurança do sistema como um todo, pois impede aplicações com erros (ou aplicações maliciosas) de interferir no funcionamento das demais aplicações.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

SO = Gerenciador de Recursos

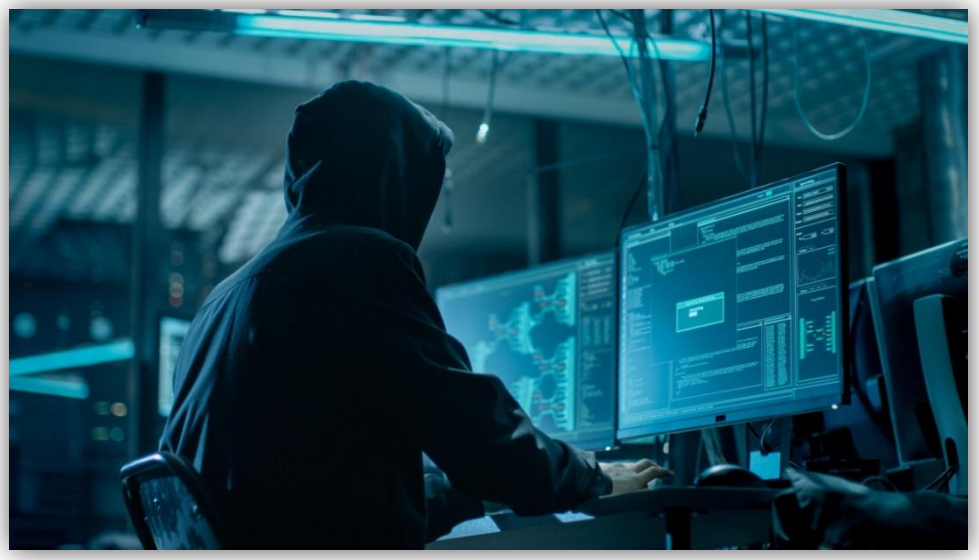
- **Gerência de Dispositivos**
  - Cada periférico do computador possui suas peculiaridades; assim, o procedimento de interação com uma placa de rede é completamente diferente
  - Da interação com um disco rígido SCSI. A função da gerência de dispositivos (também conhecida como gerência de entrada/saída) é implementar a interação com cada dispositivo por meio de **drivers** e criar modelos abstratos que permitam agrupar vários dispositivos distintos sob a mesma interface de acesso.

SO = Gerenciador de Recursos

- **Gerência de arquivos**
  - Esta funcionalidade é construída sobre a gerência de dispositivos e visa criar arquivos e diretórios, definindo sua interface de acesso e as regras para seu uso.
  - É importante observar que os conceitos abstratos de arquivo e diretório são tão importantes e difundidos que muitos sistemas operacionais os usam para permitir o acesso a recursos que nada tem a ver com armazenamento.

SO = Gerenciador de Recursos

- Gerência de proteção
  - Com computadores conectados em rede e compartilhados por vários usuários, é importante definir claramente os recursos que cada usuário pode acessar, as formas de acesso permitidas (leitura, escrita, etc.) e garantir que essas definições sejam cumpridas.



INSTITUTO FEDERAL
RIO GRANDE DO SUL

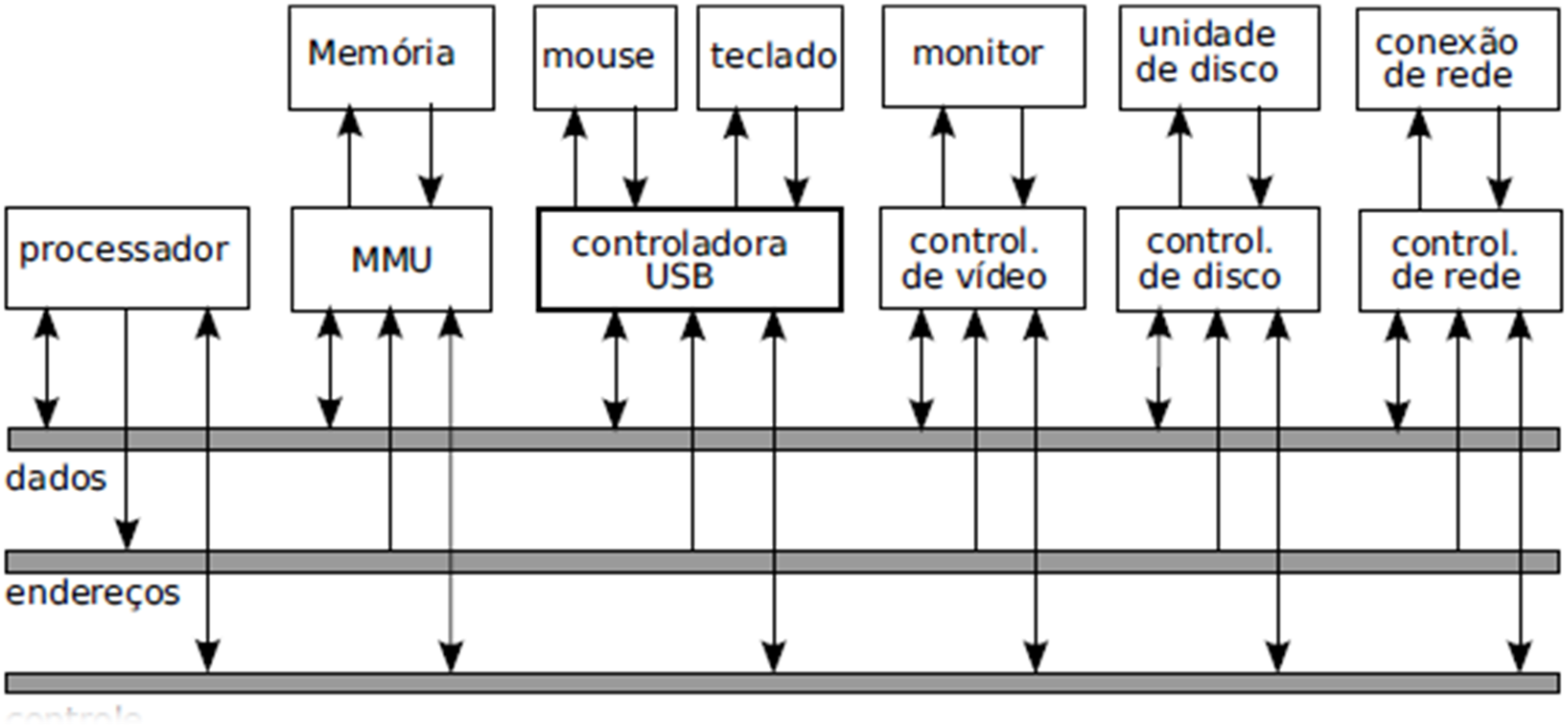
HARDWARE

- Para o processador, cada dispositivo é representado por seu respectivo controlador.

- Os controladores podem ser acessados através de portas de entrada/saída endereçáveis: a cada controlador é atribuída uma faixa de endereços de portas de entrada/saída.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

HARDWARE



dispositivo	endereços de acesso
teclado	0060h-006Fh
barramento IDE primário	0170h-0177h
barramento IDE secundário	01F0h-01F7h
porta serial COM1	02F8h-02FFh
porta serial COM2	03F8h-03FFh

INSTITUTO FEDERAL
RIO GRANDE DO SUL

INTERRUPÇÕES

Controlador querendo comunicar com o processador.

1ª opção: aguarda a consulta do processador

2ª opção: Requisição de interrupção

INSTITUTO FEDERAL
RIO GRANDE DO SUL

INTERRUPÇÕES

- Aguardar até que o processador o consulte, o que poderá ser demorado caso o processador esteja ocupado com outras tarefas (o que geralmente ocorre);

- Notificar o processador através do barramento de controle, enviando a ele uma requisição de interrupção **(IRQ – Interrupt ReQuest).**
  1. Processador suspende a execução
  2. Consulta o vetor de Interrupção (tabela).
  3. Desvia para o endereço da Vetor de Interrupção (tabela).
  4. Executa a rotina de tratamento de interrupção
  5. Retorna a execução anterior.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

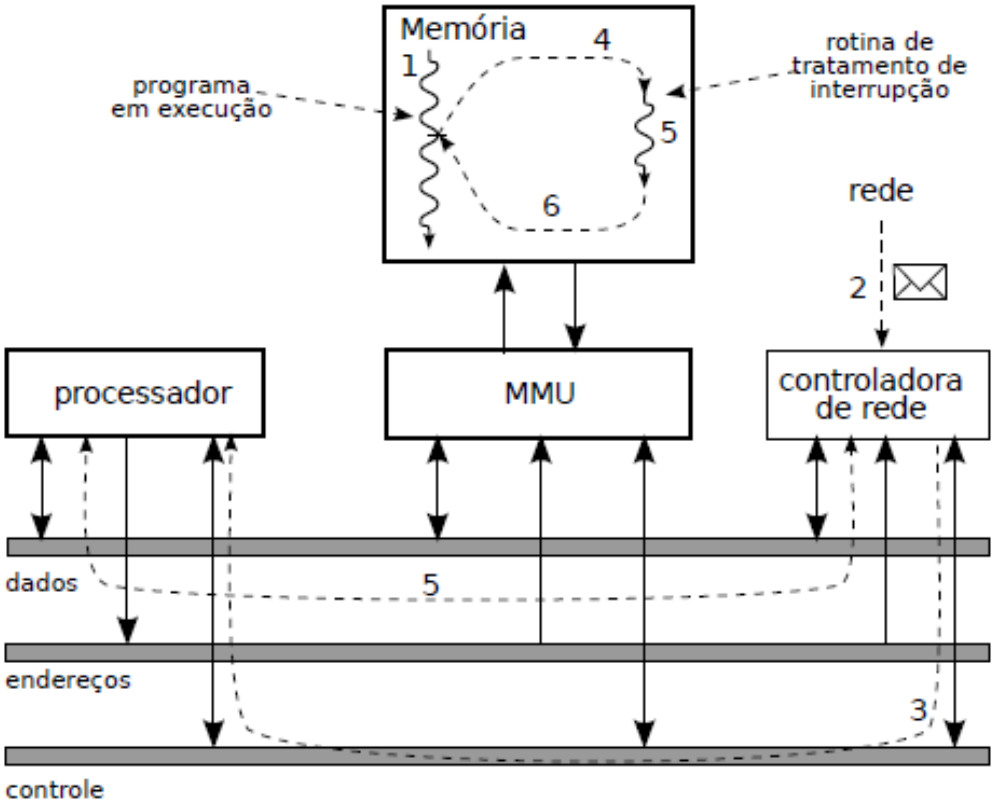
INTERRUPÇÕES

Exemplo:

1. O processador está executando um programa qualquer (em outras palavras, um fluxo de execução);
2. Um pacote vindo da rede é recebido pela placa Ethernet;
3. A placa envia uma solicitação de interrupção (IRQ) ao processador;
4. O processamento é desviado do programa em execução para a rotina de tratamento da interrupção;
5. A rotina de tratamento é executada para receber as informações da placa de rede (via barramentos de dados e de endereços) e atualizar as estruturas de dados do sistema operacional;
6. A rotina de tratamento da interrupção é finalizada e o processador retorna à execução do programa que havia sido interrompido.

INTERRUPÇÕES

Controlador querendo comunicar com o processador.



INSTITUTO FEDERAL
RIO GRANDE DO SUL

Proteção do Núcleo

O Sistema Operacional deve gerenciar o hardware:

- Apenas o SO deve acessar diretamente o hardware
- Aplicações pedem ao SO, o qual pedirá ao hardware

Aplicações com acesso pleno ao hardware tornariam inúteis os mecanismos de segurança e controle de acesso aos recursos (tais como arquivos, diretórios e áreas de memória).

INSTITUTO FEDERAL
RIO GRANDE DO SUL

Proteção do núcleo

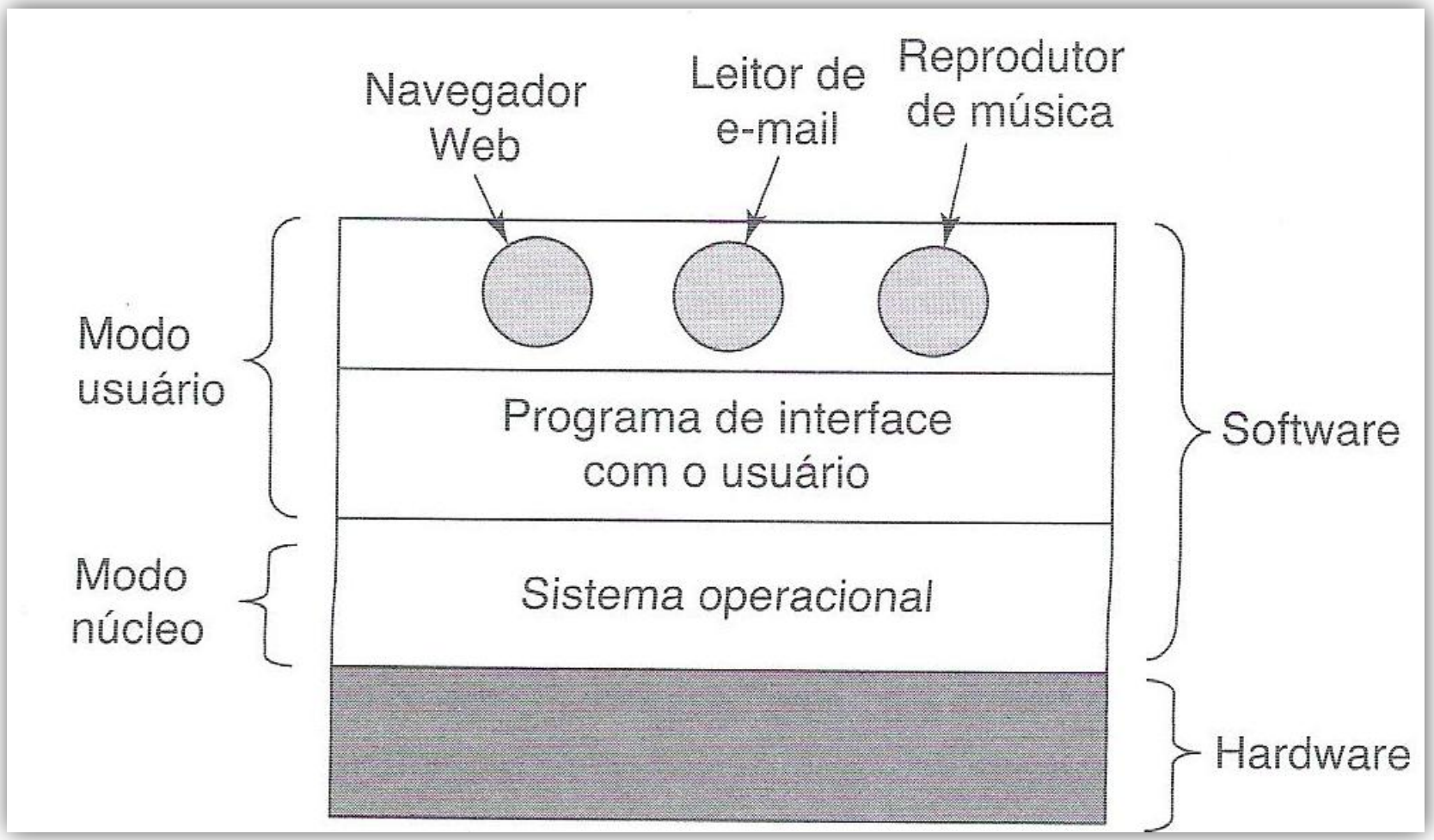
Como se consegue isso?

CPUS tem níveis de privilégio de execução

Flags na CPU que definem o nível de acesso ao hardware.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

MODOS DE OPERAÇÃO



INSTITUTO FEDERAL
RIO GRANDE DO SUL

MODOS DE OPERAÇÃO

NÍVEL NÚCLEO :

- também denominado nível supervisor, sistema, monitor ou ainda kernel space.

- Para um código executando nesse nível, todo o processador está acessível: todos os recursos internos do processador (registradores e portas de entrada/saída) e áreas de memória podem ser acessados.

- Além disso, todas as instruções do processador podem ser executadas.

- Ao ser ligado, o processador entra em operação neste nível.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

MODOS DE OPERAÇÃO

NÍVEL USUÁRIO (ou userspace):

- Neste nível, somente um sub-conjunto das instruções do processador, registradores e portas de entrada/saída estão disponíveis.
- Instruções “perigosas” como HALT (parar o processador) e RESET (reiniciar o processador) são proibidas para todo código executando neste nível.
- Além disso, o hardware restringe o uso da memória, permitindo o acesso somente a áreas previamente definidas.
- Caso o código em execução tente executar uma instrução proibida ou acessar uma área de memória inacessível, o hardware irá gerar uma exceção, desviando a execução para uma rotina de tratamento dentro do núcleo, que provavelmente irá abortar o programa em execução (e também gerar a famosa frase “este programa executou uma instrução ilegal e será finalizado”, no caso do Windows).

INSTITUTO FEDERAL
RIO GRANDE DO SUL

Chamadas de Sistemas

Mas como um programa invoca o SO para acessar um recurso?

INSTITUTO FEDERAL
RIO GRANDE DO SUL

Chamadas de sistemas

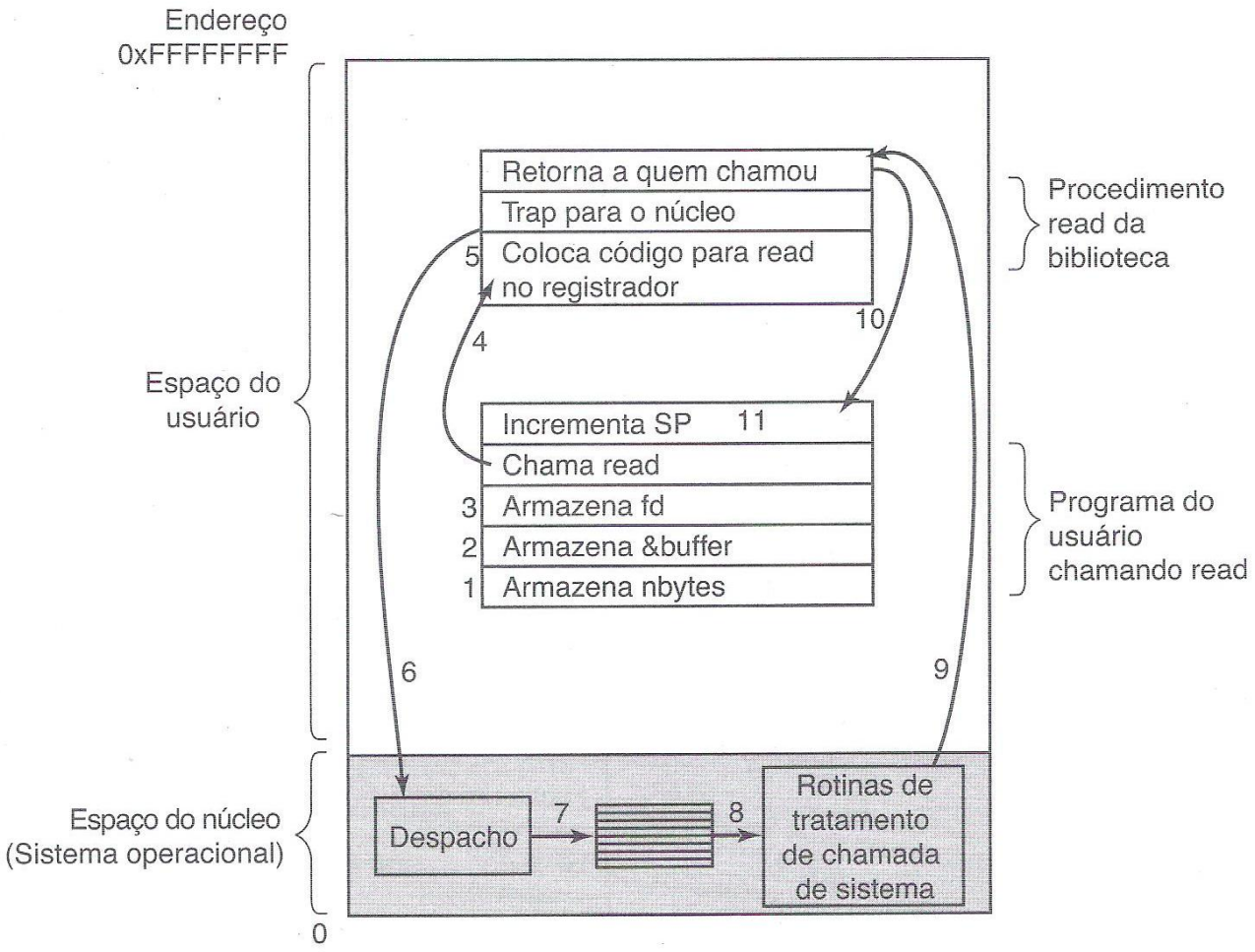
Uma chamada do sistema é uma solicitação feita por um programa ao SO. Permite uma aplicação acessar recursos do sistema operacional.

As chamadas do sistema realizam operações no nível do sistema, como comunicação com Hardware dispositivos e leitura e escrita arquivos.

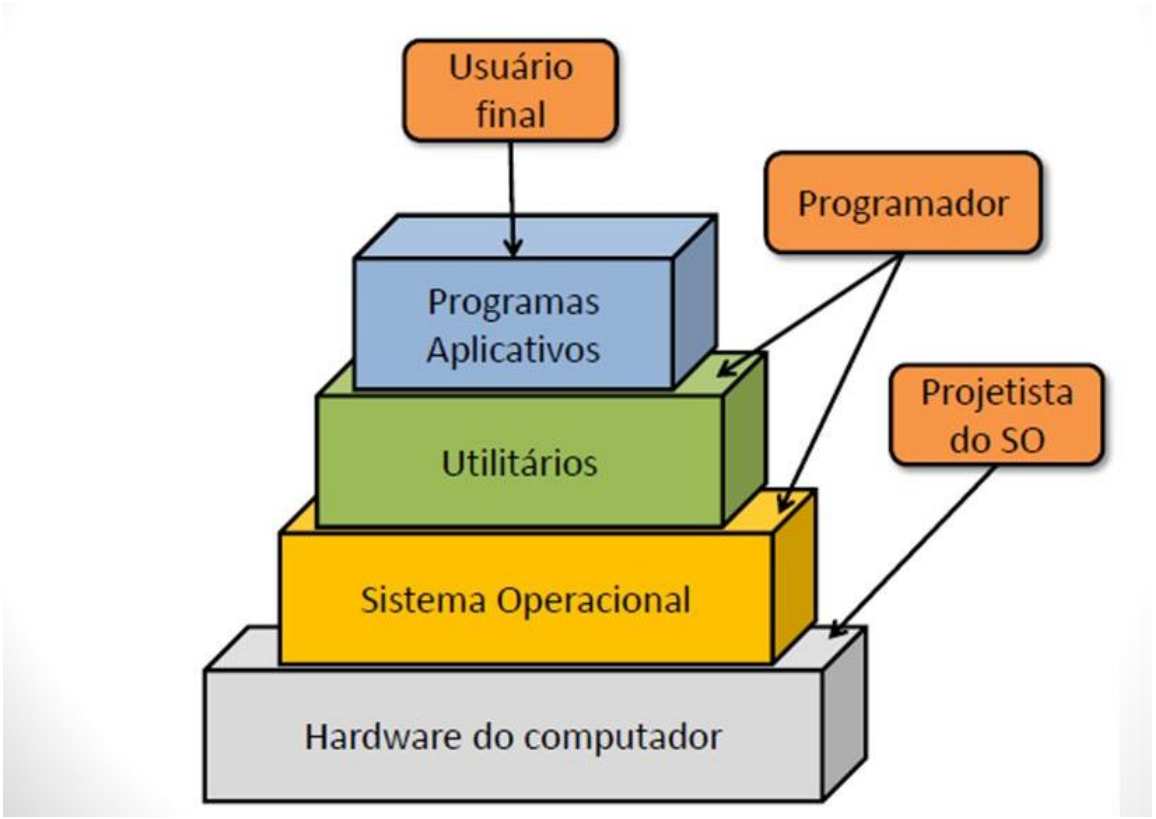
Ao fazer chamadas do sistema, os desenvolvedores podem usar funções pré-escritas suportadas pelo sistema operacional (SO) em vez de escrevê-las do zero.

Isso simplifica o desenvolvimento, melhora a estabilidade do aplicativo e torna os aplicativos mais "portáteis" entre diferentes versões de um sistema operacional.

Chamadas de sistemas



ARQUITETURAS DE SISTEMAS OPERACIONAIS



INSTITUTO FEDERAL
RIO GRANDE DO SUL

ARQUITETURAS DE SIST. OPERAC.

Um SO pode ter uma arquitetura do tipo:

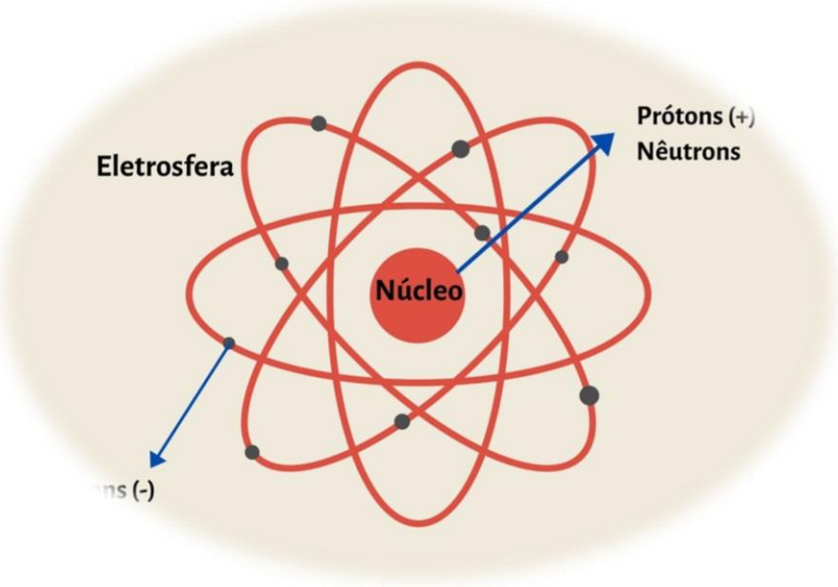
01) MONOLÍTICOS (ou Monocúcleos)

02) DE CAMADAS

03) MICRONÚCLEO

04) SISTEMAS HIBRIDOS

- *MÁQUINAS VIRTUAIS
- *EXONÚCLEO
- * UNINÚCLEO
- * CONTAINERS



01 - SISTEMAS MONILÍTICOS
(MONONÚCLEOS)

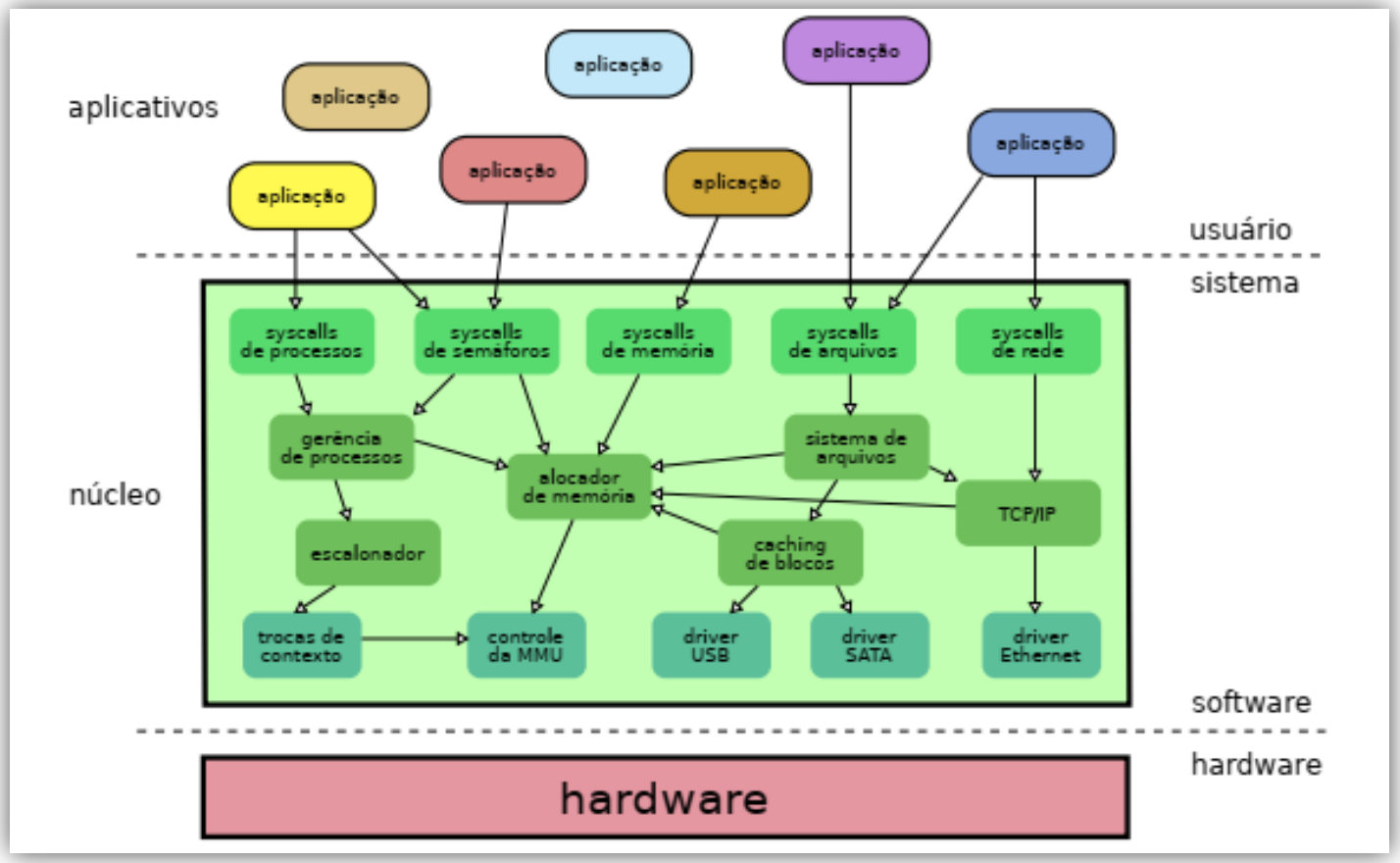
- Todo o sistema operacional é executado no núcleo
- Opera em modo núcleo, com acesso a todos os recursos do hardware e sem restrições de acesso à memória.
- Grande vantagem da arquitetura monolítica é seu desempenho: qualquer componente do núcleo pode acessar os demais componentes
- Considerando que todos os componentes do SO têm acesso privilegiado ao hardware, caso um componente perca o controle,pode levar o sistema ao colapso.

Rotina principal: grande programa binário executável. Liga-se as outras rotinas. Expandindo as funcionalidades do S.O.
Rotinas de serviços e utilitários: são ligados ao programa binário executável.

Estrutura básica:
1) Um programa principal que invoca a rotina do serviço requisitado.
2) Um conjunto de rotinas de serviço que executam as chamadas de sistema
3) Um conjunto de rotinas utilitárias que auxiliam as rotinas de serviço

01 - SISTEMAS MONILÍTICOS
(MONONÚCLEOS)

A arquitetura monolítica foi a primeira forma de organizar os sistemas operacionais; sistemas UNIX antigos e o MS-DOS seguiam esse modelo.



INSTITUTO FEDERAL
RIO GRANDE DO SUL

02 - SISTEMAS DE CAMADAS

- Hierarquia de camadas (Divisão).

- A camada mais baixa realiza a interface com o hardware, enquanto as camadas intermediárias proveem níveis de abstração e gerência cada vez mais sofisticados. Por fim, a camada superior define a interface do núcleo para as aplicações (as chamadas de sistema).

- As camadas têm níveis de privilégio decrescentes: a camada inferior tem acesso total ao hardware, enquanto a superior tem acesso bem mais restrito.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

02 - SISTEMAS DE CAMADAS

A estruturação em camadas é apenas parcialmente adotada hoje em dia. Como exemplo de sistema fortemente estruturado em camadas pode ser citado o MULTICS e o THE - Technische Hogeschool Eindhoven (1968)

CAMADA	FUNÇÃO
5	O operador
4	Programas de usuário
3	Gerenciamento de entrada/saída
2	Comunicação operador-operador
1	Memória e gerenciamento de tambor
0	Alocação do processador e multiprogramação

INSTITUTO FEDERAL
RIO GRANDE DO SUL

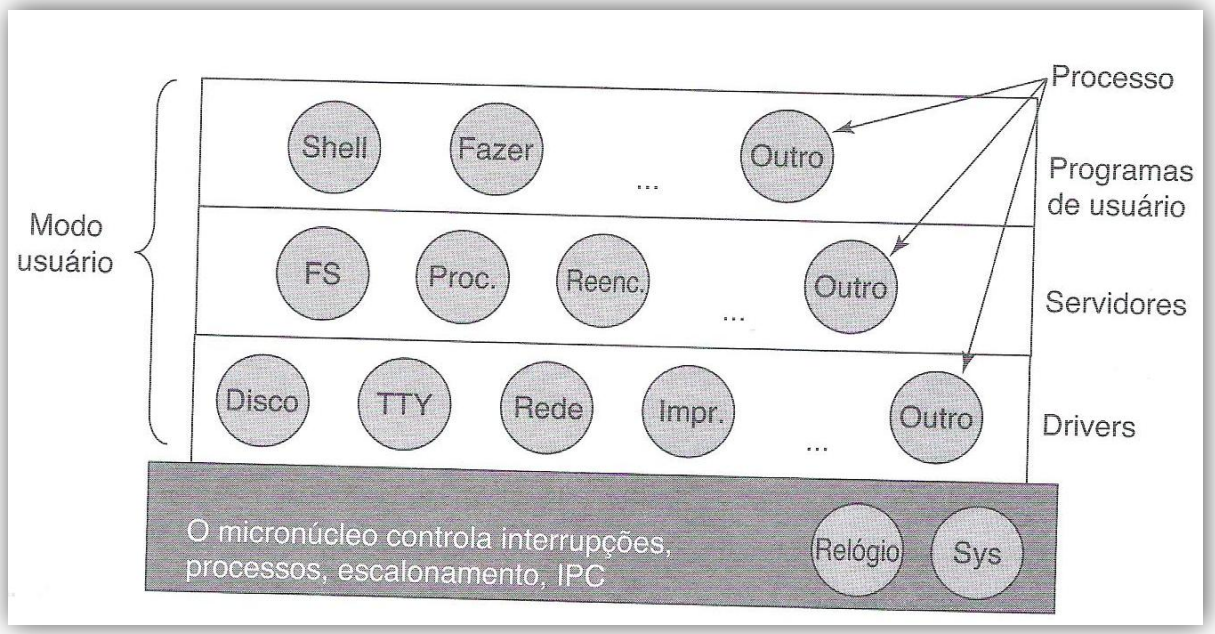
03) SISTEMAS MICRONÚCLEO

- Consistes em dividir o núcleo em pequenos módulos.
- O Objetivo é a diminuição do número de erros no núcleo.
- Consiste em retirar do núcleo todo o código de alto nível, deixando no núcleo somente o código de baixo nível. O restante do código seria transferido para programas separados no espaço de usuário, denominados *serviços*.
- Por fazer o núcleo de sistema ficar menor, essa abordagem foi denominada *micronúcleo* (ou µ-*kernel*).

Exemplo: Symbian, Minix 3

03) SISTEMAS MICRONÚCLEO

- O micronúcleos vem sendo investigados desde os anos 1980. Os melhores exemplos de micronúcleos bem sucedidos são provavelmente o Minix 3 e o QNX. Vários sistemas operacionais atuais adotam parcialmente essa estruturação, adotando núcleos híbridos, como por exemplo o MacOS X da Apple, o Digital UNIX e o Windows NT.



Exemplo: se a unidade de audio tiver erro, para-se o audio ou apenas distorce-se o audio. No monolítico, o sistema inteiro irá parar.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

04) SISTEMAS HÍBRIDOS

- Devido ao baixo desempenho dos MICRONÚCLEOS, uma solução encontrada para esse problema consiste em trazer de volta ao núcleo os componentes mais críticos, para obter melhor desempenho.

- Essa abordagem intermediária entre o núcleo monolítico e micronúcleo é denominada *núcleo híbrido*. (também teve a influência da arquitetura em camadas).

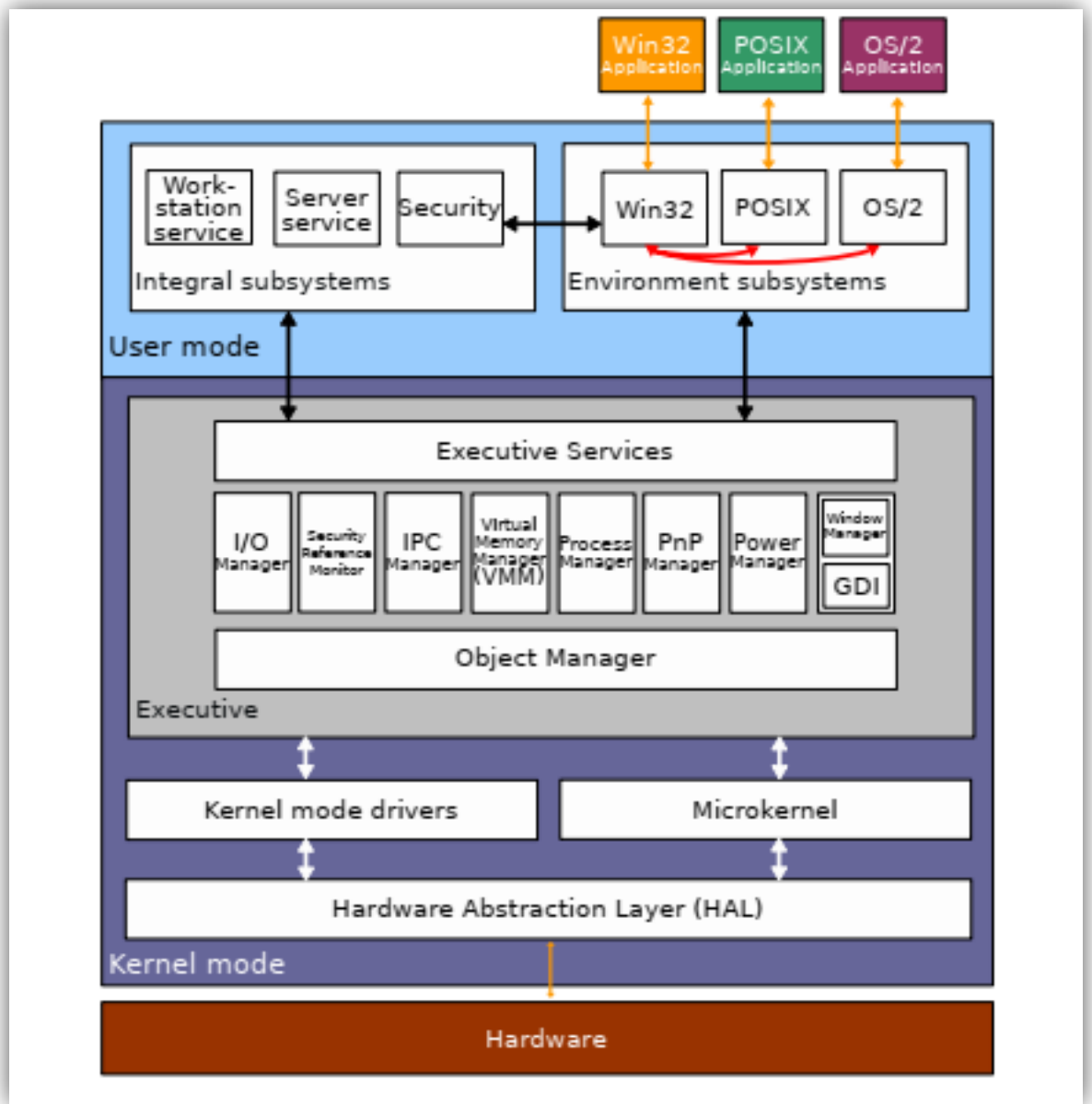
- Vários sistemas operacionais atuais empregam núcleos híbridos, entre eles o Microsoft Windows, a partir do Windows NT.

- As primeiras versões do Windows NT podiam ser consideradas micronúcleo, mas a partir da versão 4 vários subsistemas foram movidos para dentro do núcleo para melhorar seu desempenho, transformando-o em um núcleo híbrido.

- Os sistemas MacOS e iOS da Apple também adotam um núcleo híbrido chamado XNU (de *X is Not Unix*), construído a partir dos núcleos Mach (micronúcleo) e FreeBSD (monolítico)

INSTITUTO FEDERAL
RIO GRANDE DO SUL

04) SISTEMAS HÍBRIDOS



Arquiteturas Avançadas

SISTEMAS MÁQUINA VIRTUAIS

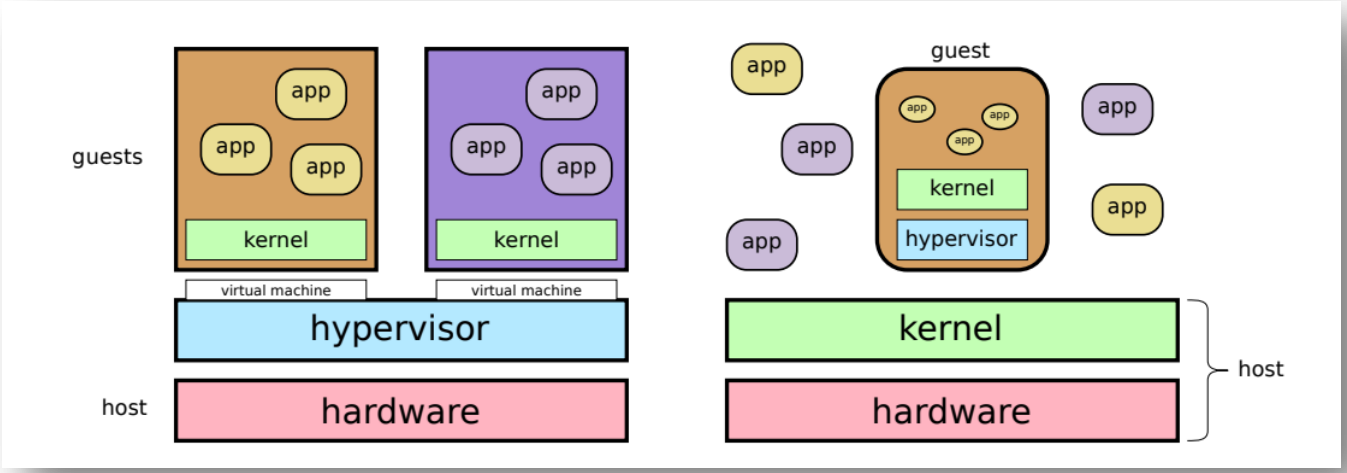
Utilizado em grandes servidores. Ex. IBM z13

Separa os seguintes elementos:

- Multiprogramação
- Estende o hardware com uma interface melhor

Monitor da MV fornece várias máquinas virtuais.

Cada MV pode rodar um S.O. diferente.



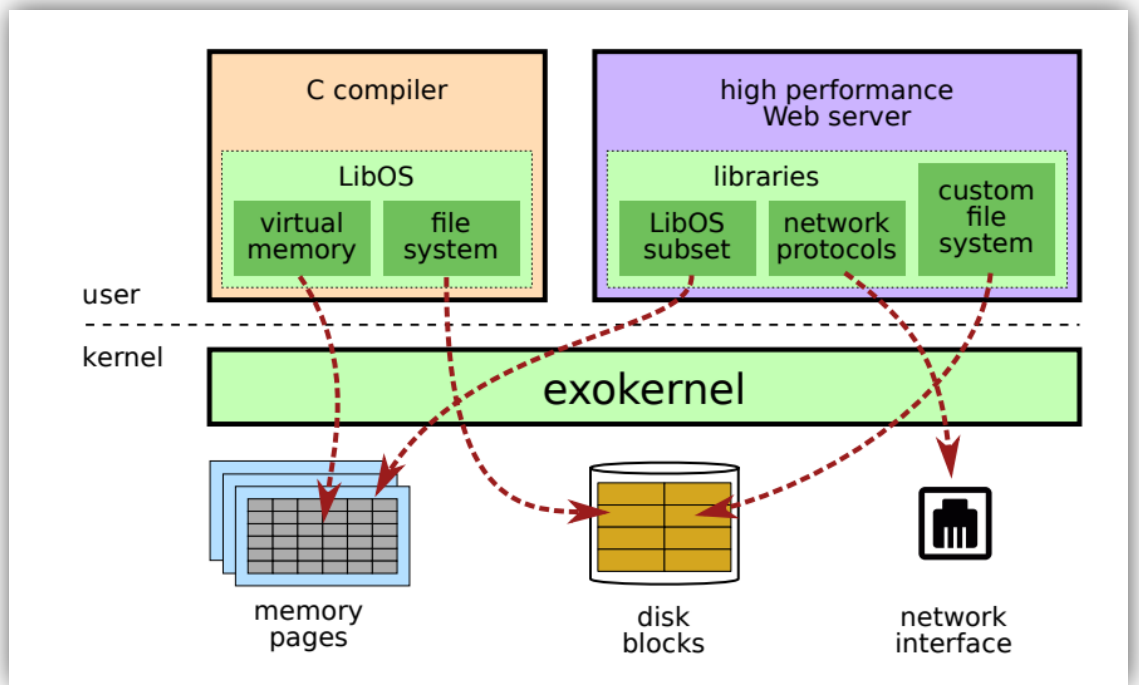
INSTITUTO FEDERAL
RIO GRANDE DO SUL

SISTEMAS EXONÚCLEO

- Os exonúcleos (*exokernels*) o núcleo do sistema apenas proporciona acesso controlado aos recursos do hardware, mas não implementa nenhuma abstração.
- Todas as abstrações e funcionalidades de gerência que uma aplicação precisa terão de ser implementadas pela própria aplicação, em seu espaço de memória
- Subdivide a máquina real: Um pouco de cada recurso para cada usuário.
- Não precisa camada de mapeamento.

SISTEMAS EXONÚCLEO

- É importante observar que um exonúcleo é diferente de um micronúcleo, pois neste último as abstrações são implementada por um conjunto de processos servidores independentes e isolados entre si, enquanto no exonúcleo cada aplicação implementa suas próprias abstrações (ou as incorpora de bibliotecas compartilhadas).
- Até o momento não existem exonúcleos em produtos comerciais



INSTITUTO FEDERAL
RIO GRANDE DO SUL

SISTEMAS UNINÚCLEO

- Nos sistemas uninúcleo (*unikernel*) as bibliotecas e uma aplicação são compilados e ligados entre si,formando um bloco monolítico de código.

- O custo da transição aplicação / núcleo nas chamadas de sistema diminui muito, gerando um forte ganho de desempenho.

- Adequada para computadores que executam uma única aplicação, como servidores de rede (HTTP, DNS, DHCP).

- Sistemas embarcados e nas chamadas *appliances* de rede (roteadores, modems, etc), é usual compilar o núcleo, bibliotecas e aplicação em um único arquivo binário que é em seguida executado diretamente sobre o hardware, em modo privilegiado.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

SISTEMAS CONTAINERS

- O espaço de usuário do sistema operacional é dividido em áreas isoladas denominadas *domínios* ou *contêineres*.
- A cada domínio é alocada uma parcela dos recursos do sistema operacional, como memória, tempo de processador e espaço em disco.
- Para garantir o isolamento entre os domínios, cada domínio tem sua própria interface de rede virtual e, portanto, seu próprio endereço de rede (IP).
- Processos não podem trocar de domínio nem acessar recursos de outros domínios. Os domínios são vistos como sistemas distintos, acessíveis somente através de seus endereços de rede.

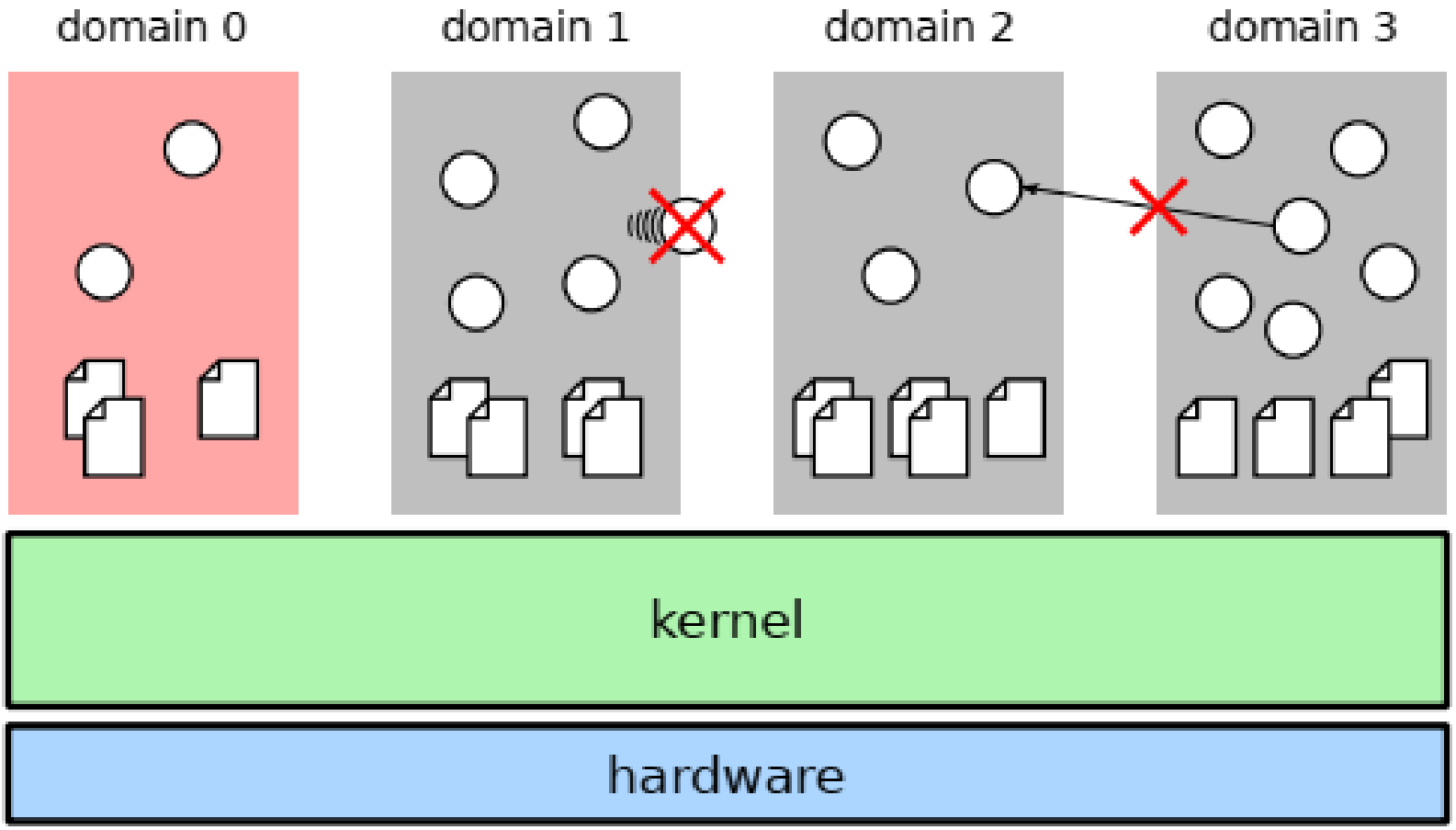
INSTITUTO FEDERAL
RIO GRANDE DO SUL

SISTEMAS CONTAINERS

- Além disso, na maioria das implementações cada domínio tem seu próprio espaço de nomes para os identificadores de usuários, processos e primitivas de comunicação.
- O núcleo do sistema operacional é o mesmo para todos os domínios.
- Processos em um mesmo domínio podem interagir entre si normalmente, mas processos em domínios distintos não podem ver ou interagir entre si.
- Processos não podem trocar de domínio nem acessar recursos de outros domínios. Os domínios são vistos como sistemas distintos, acessíveis somente através de seus endereços de rede.
- O núcleo Linux oferece diversos mecanismos para o isolamento de espaços de recursos, que são usados por plataformas de gerenciamento de contêineres como *Linux Containers* (LXC), *Docker* e *Kubernetes*.

INSTITUTO FEDERAL
RIO GRANDE DO SUL

SISTEMAS CONTAINERS



INSTITUTO FEDERAL
RIO GRANDE DO SUL