

Sistemas Operacionais

Professor:
Cristiano Bonato Both



JESUÍTAS BRASIL



Somos infinitas possibilidades

Sumário

- Princípios básicos de hardware
 - Arquitetura de computadores
- Dispositivos de entrada e saída
- Mapeamento de memória
- Técnicas de realização de E/S
- Evolução de arquiteturas de E/S
- Gerência de entrada e saída
- Referências



JESUÍTAS BRASIL



Somos infinitas possibilidades

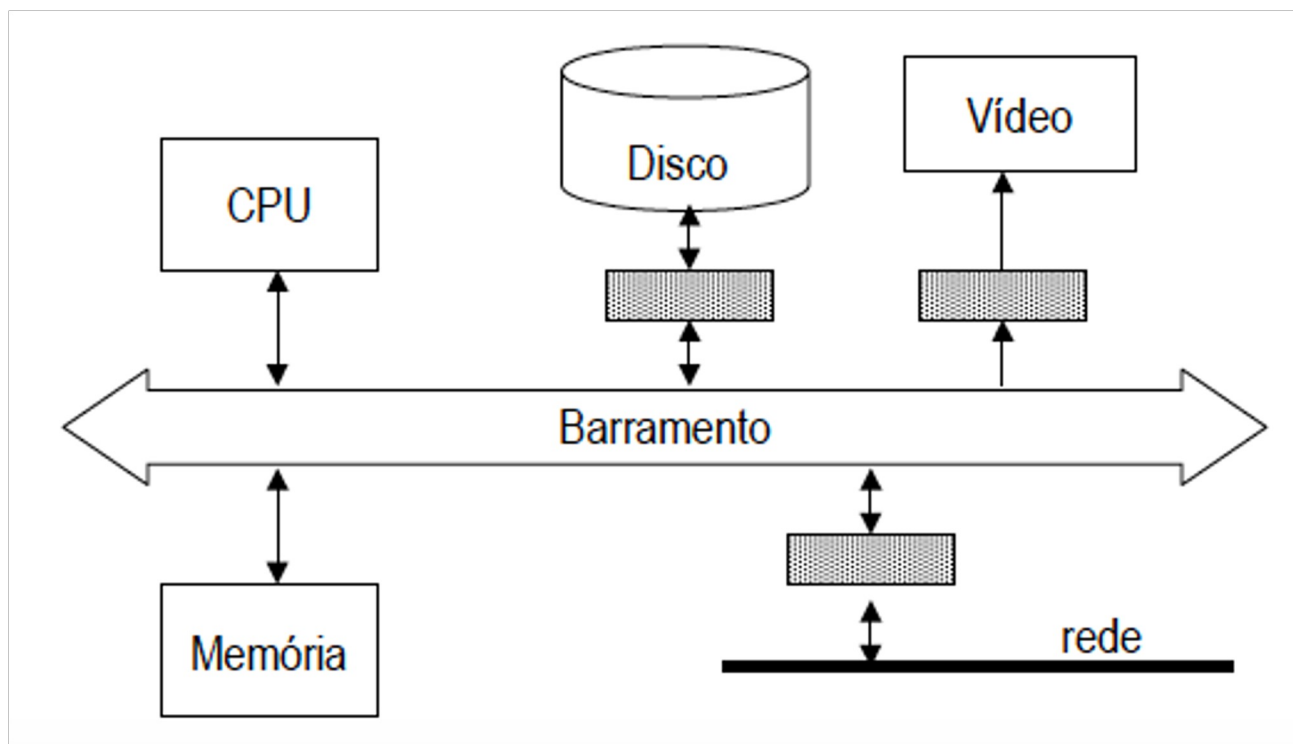
Princípios básicos de hardware

- Periférico: dispositivo que possibilita a interação com o mundo externo
- Periféricos são conectados ao computador através de um componente de hardware denominado de interface
- Interfaces são interconectadas aos barramentos internos de um computador
 - Elemento chave na coordenação da transferência de dados
 - Se utilizam de um processador dedicado à realização e controle das operações de entrada e saída
 - Controladoras



Arquitetura de entrada e saída

- Dispositivo de entrada e saída possui uma parte mecânica e outra eletrônica



Dispositivos de entrada e saída

- Classificados como:
 - Orientado a caractere
 - e.g., teclado, interface serial
 - Orientado a bloco
 - e.g., disco
 - Alguns dispositivos não se enquadram nestas situações
 - e.g., relógio, memória de vídeo mapeada em espaço de E/S



Dispositivos de entrada e saída

- Dispositivos de E/S podem ser classificados de acordo com o tipo de entidade que interagem:
 - Com usuário
 - e.g., vídeo, teclado, mouse, impressora, etc.
 - Com dispositivos eletrônicos
 - e.g., discos, pen drives, etc.
 - Com dispositivos remotos
 - Modem, placas de rede, etc.



Dispositivos de entrada e saída

- Apresentam características próprias:
 - Taxa de transferência de dados
 - Complexidade de controle
 - Unidade de transferência
 - Caractere, bloco ou *stream*
 - Representação de dados
 - Esquemas de codificação
 - Tratamento de erros
 - Depende do tipo de dispositivo



Tipos de conexão e transferência de dados

- Em função da interconexão física das interfaces com os periféricos, podem ser classificadas em dois tipos
 - Interface serial
 - Apenas uma linha para transferência de dados (bit-a-bit)
 - Interface paralela
 - Mais de uma linha para transferência de dados
 - e.g., $n \times 8$ bits



Como controladoras e sistemas operacionais interagem?

- Controladora é programada via registradores de configuração
 - Recebem ordens do processador
 - Fornecem estados de operação
 - Leitura e escrita de dados do periférico
- Registradores são “vistos” como posições de memória
 - E/S mapeada em espaço de E/S
 - E/S mapeada em espaço de memória



Mapeamento em espaço de memória e em espaço de E/S

- Espaço de endereçamento:
 - Conjunto de endereços de memória que o processador consegue endereçar
 - Definido no projeto de processador
 - **Pode haver um único espaço de endereçamento**
 - **Pode haver um espaço de endereçamento dedicado a E/S**
 - Instruções específicas para acessar um ou outro espaço de endereçamento
 - *e.g., mov, in, out*



Mapeamento em espaço de memória

- Um único espaço de endereçamento
- No projeto do computador se reserva uma parte de sua área de endereçamento para acesso a periféricos (controladoras)
- Instruções de acesso a memória do tipo "*mov end, dado*" podem tanto referenciar uma posição real de memória como um registrador associado a um periférico de E/S
- e.g., processadores da família Motorola



JESUÍTAS BRASIL



Somos infinitas possibilidades

Mapeamento em espaço de E/S

- O processador possui duas áreas distintas de endereçamento
 - Espaço de memória: acessado via instruções de acesso de memória (*mov*)
 - Espaço de E/S: acessado via instruções de acesso específica (*in, out*)
- Neste tipo de processador é possível utilizar os dois mapeamentos para acesso a periféricos de E/S
- e.g., processadores da família Intel



Técnicas para realização de E/S

- Três técnicas normalmente utilizadas:
 - E/S programada
 - E/S orientada a interrupções
 - Acesso direto à memória



JESUÍTAS BRASIL



Somos infinitas possibilidades

E/S programada

- Interação entre o processador e controlador é de responsabilidade do **programador**
- Ciclo de funcionamento
 - Envio de comando à controladora
 - Espera pela realização do comando
- Módulo (controladora) de E/S atualiza bits de estado da operação
- Processador espera o término da operação (*busy waiting*)



Desvantagem E/S programada

- Desperdício do tempo do processador para verificar continuamente o estado de uma operação de E/S
- **Solução é inserir operações entre sucessivas consultas sobre o estado de uma operação de E/S**
 - *Polling*
- Problema: determinar a frequência para a realização do *polling*



E/S orientada à interrupção

- Método utilizado para evitar o desperdício de tempo do método de *polling*
- Processador é interrompido quando o módulo de E/S está pronto
- Enquanto a interrupção não ocorre, o processador está liberado para executar outras tarefas – **Troca de Contexto**
- Processador é responsável por iniciar uma operação de E/S
- Interrupção solicita atenção do processador para executar uma rotina específica ao final da operação de E/S



E/S orientada à interrupção

- Desvantagens:
 - Processador atua como um intermediário na transferência de dados
 - Cada palavra lida/escrita passa pelo processador



Acesso direto à memória

- DMA (*Direct Memory Access*)
- Transfere diretamente um bloco de dados entre a memória e o módulo de E/S
- Mecanismo de interrupção é utilizado para sinalizar final de tarefa
- Processador é envolvido com a tarefa de E/S apenas no começo e no final da transferência (controle)



Evolução de arquitetura de E/S

- Processador controla diretamente o periférico
- Controlador ou módulo de E/S é adicionado
- Controlador ou módulo de E/S baseado em interrupções
- Transferência de dados em DMA
- Módulo de E/S possui um processador separado
- Processador de E/S



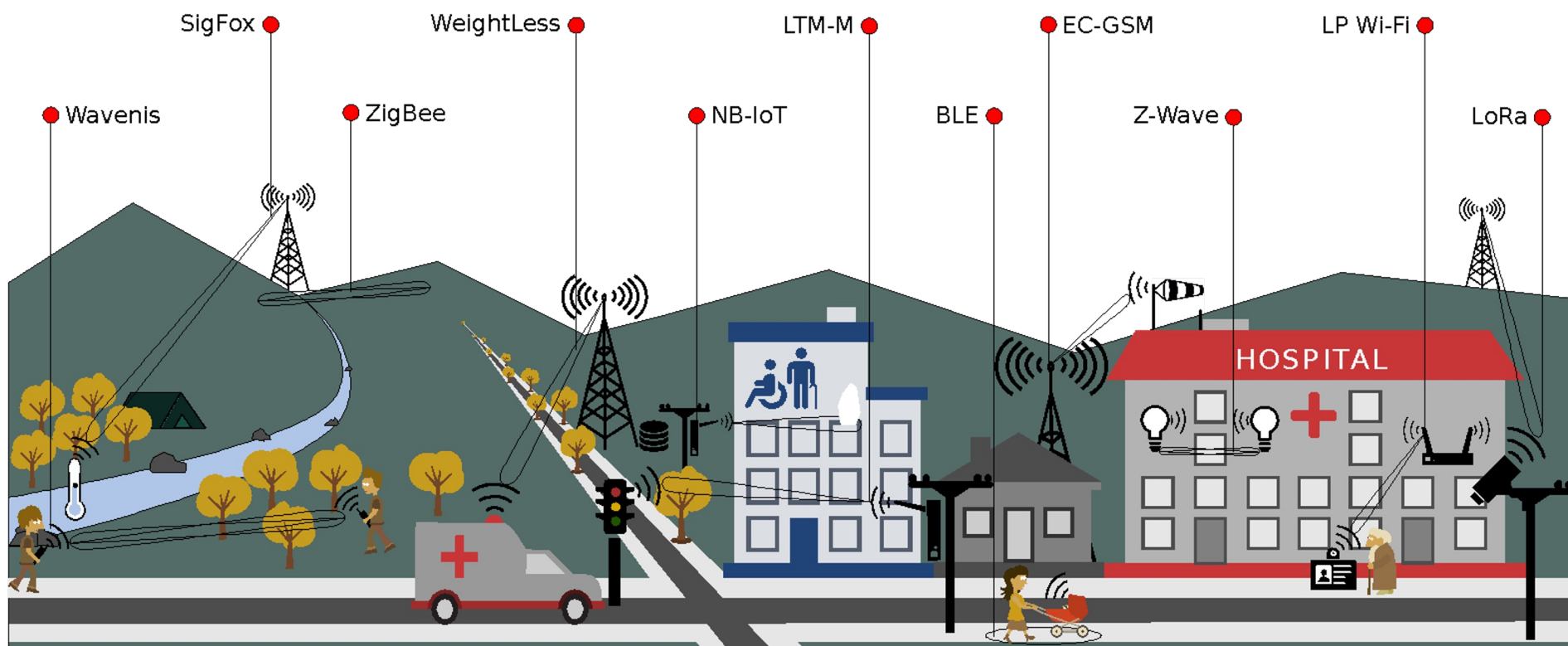
JESUÍTAS BRASIL



Somos infinitas possibilidades

O problema da gerência de E/S

- Como gerenciar a futura geração de tecnologias IoT?



JESUÍTAS BRASIL



Somos infinitas possibilidades

O problema da gerência de E/S

- Entrada e saída: extremamente lenta se comparada com a velocidade de processamento e de acesso a memória principal
- Multiprogramação: possibilita que processos executem, enquanto outros esperam por operações de E/S
- Procedimento de *swapping* é uma entrada e saída
- Eficiência no tratamento de entrada e saída é importante



JESUÍTAS BRASIL



Somos infinitas possibilidades

Objetivos da gerência de E/S

- Eficiência
- Generalidade
- Esconder os detalhes do serviço de E/S em camadas de mais baixo nível
- Fornecer ao alto nível abstrações genéricas como *read*, *write*, *open* e *close*
- Envolve aspectos de hardware e de software



JESUÍTAS BRASIL



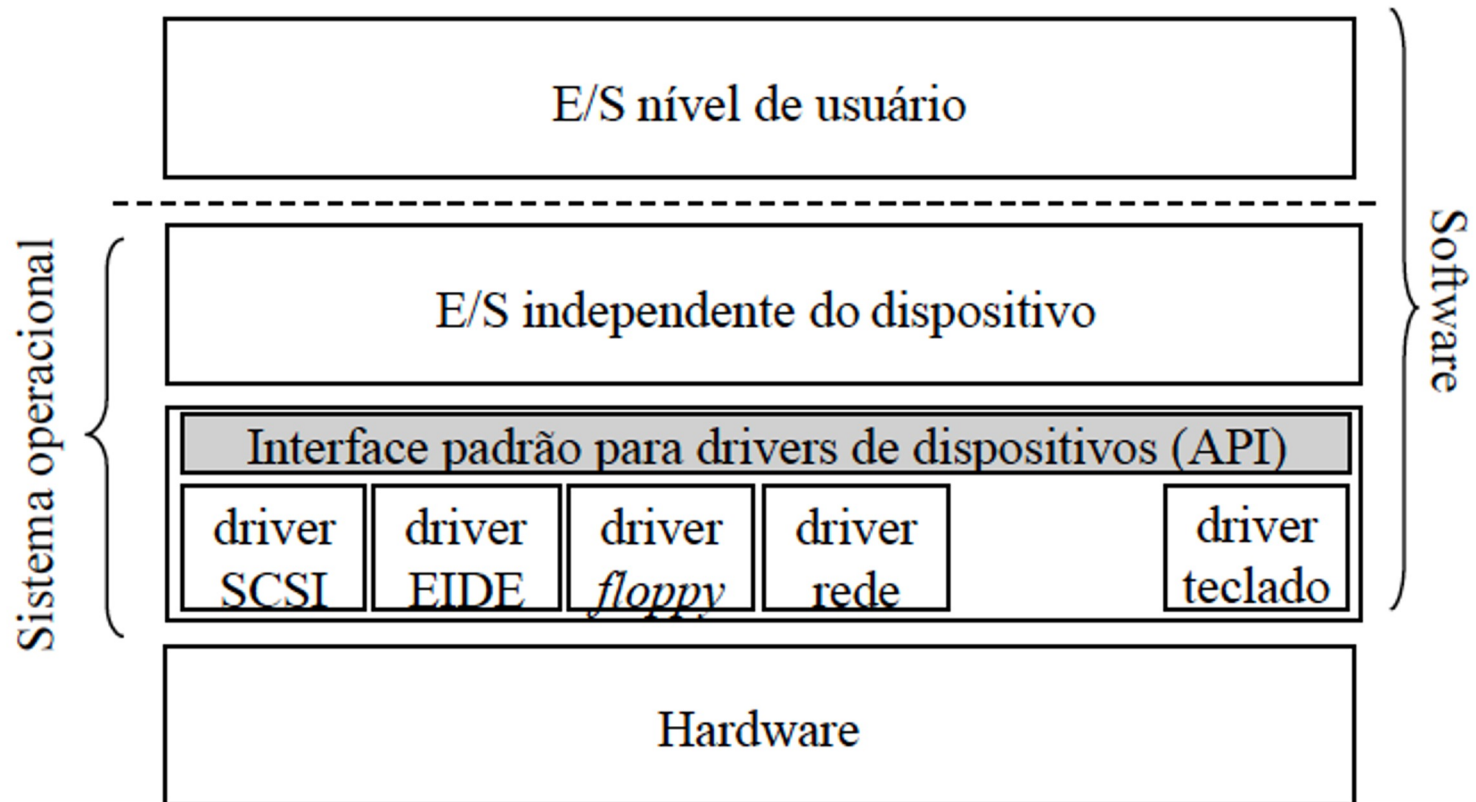
Somos infinitas possibilidades

Princípios básicos de software de entrada e saída

- Subsistema de entrada e saída é bastante complexo, devido a diversidade de periféricos
- Objetivo: padronizar as rotinas de acesso aos periféricos de E/S de forma a reduzir o número de rotinas
 - Permite inclusão de novos dispositivos sem alterar a “visão” do usuário (interface de utilização)
- Para atingir esse objetivo o subsistema de E/S é organizado em camadas



Estrutura em camadas do subsistema de entrada e saída



Organização lógica do software de entrada e saída

- Segue uma filosofia modular (três módulos base)
 - Dispositivo lógico de Entrada e Saída
 - Dispositivo físico de Entrada e Saída
 - Escalonamento e controle
- Organização não é única
 - Depende do tipo do dispositivo e de sua aplicação



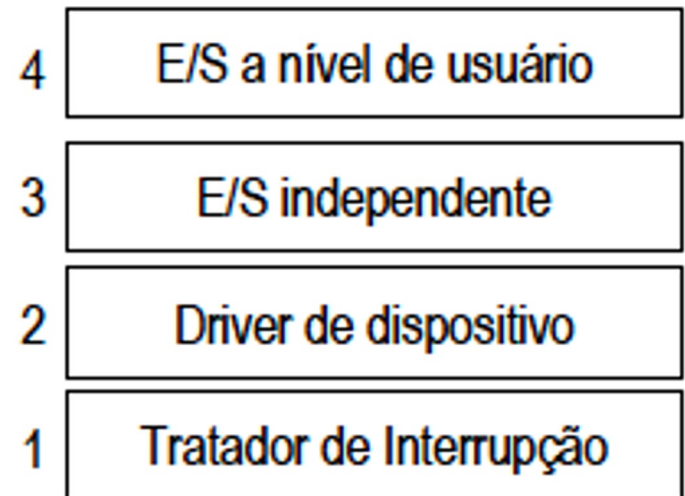
JESUÍTAS BRASIL



Somos infinitas possibilidades

Visão geral do software de E/S

1. Tratador de interrupção
 - Aciona driver ao final da operação de transferência
2. Driver de dispositivo
 - Configuração controladora e status
 - Recebimento de requisições
3. E/S independente do dispositivo
 - Nomeação e proteção
 - Bufferização
4. E/S em nível de usuário
 - Chamadas de E/S
 - Formatação de E/S



Device driver

- Conjunto de estruturas de dados e funções que controlam um ou mais dispositivos interagindo com o núcleo via uma interface bem definida
- Fornecido pelo fabricante do periférico
- Vantagens:
 - Isolar código específico a um dispositivo em um módulo a parte
 - Facilidade de adicionar novos drivers
 - Fabricantes não necessitam “mexer” no núcleo
 - Núcleo tem uma visão uniforme de todos os dispositivos através de uma mesma interface



Funcionamento básico

- Núcleo aciona driver para:
 - Configurar dispositivo
 - Realizar acessos de leitura e escrita
 - Controlar e obter informações de status
 - Tratamento de interrupções
- Driver aciona núcleo para:
 - Efetuar gerenciamento de buffers
 - Escalonamento



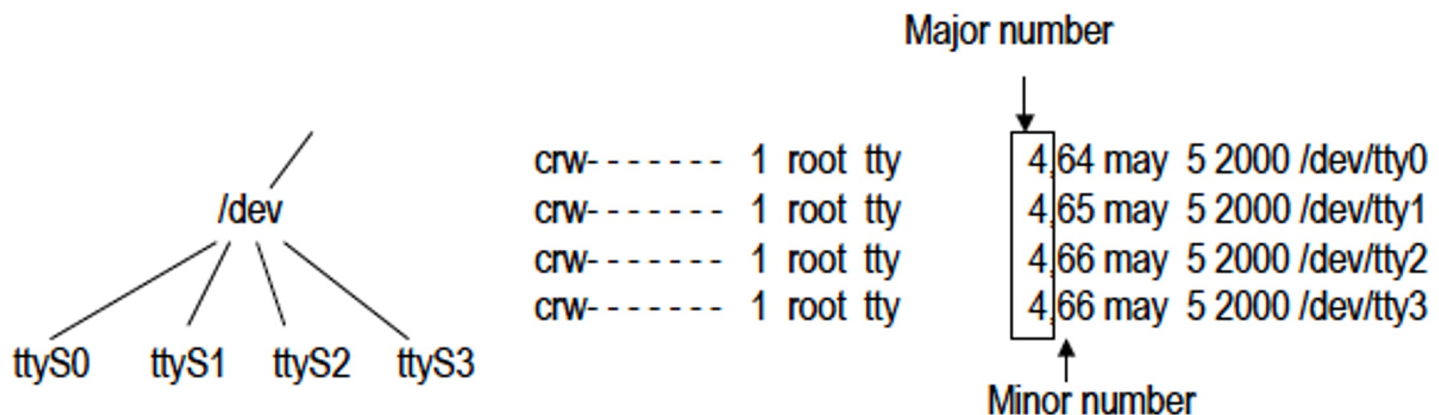
Subsistema de entrada e saída

- Porção do núcleo que controla a parte independente e interage com o driver de dispositivo para manipular/tratar a parte dependente
- Responsável por:
 - Distribuição uniforme dos nomes (nomeação, espaço de nomes)
 - Proteção
 - Fornecer uma interface aos processos usuários (aplicativos)
 - Gerenciamento e desempenho do sistema de E/S



Exemplo: subsistema de entrada e saída do UNIX

- Corresponde a visão lógica do dispositivo
- Atribuição uniforme do nome independente do dispositivo
- UNIX é um exemplo clássico:
 - Nome do dispositivo é um string
 - Discos (dispositivos) fazem parte do sistema de arquivos



Funcionalidades básicas do subsistema de E/S

- Escalonamento de Entrada e Saída
 - Determinar a melhor ordem para o atendimento de requisições de entrada e saída
 - Dividir de forma justa o acesso a periféricos
- Bufferização
 - Área de armazenamento temporário de dados
- Cache
 - Permitir o acesso rápido aos dados
- Spooling
 - Controlar acesso a dispositivos que atendem apenas uma requisição por vez
- Reserva de dispositivos
 - Controlar acesso exclusivo a dispositivos (alocação, desalocação, deadlock)



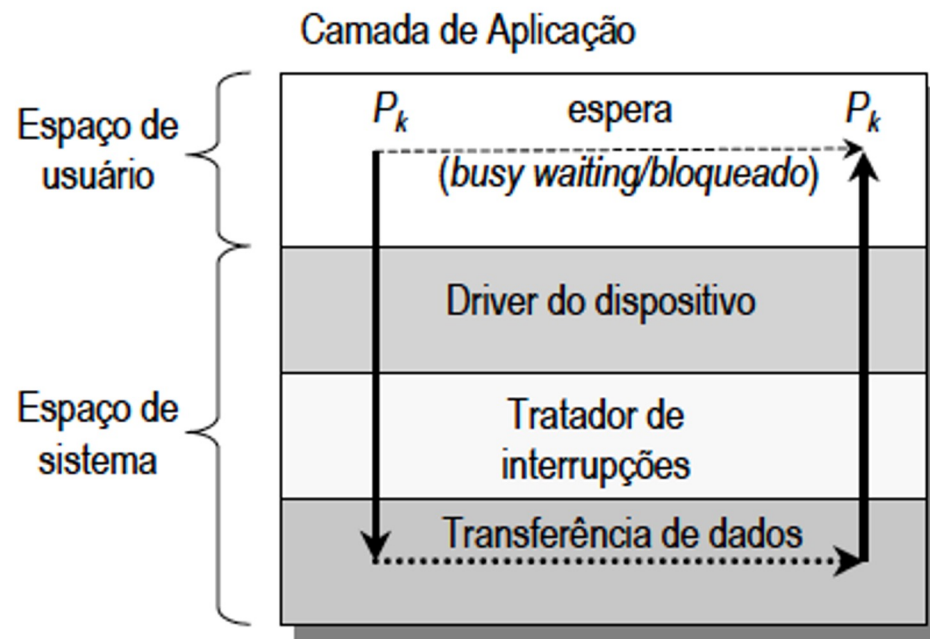
Interface de programação

- Bloqueante
 - Processo é suspenso até a conclusão da operação
 - Fácil programação e compreensão
- Não bloqueante
 - Retorna imediatamente com os dados disponíveis no momento
 - Baseado em buffer
- Assíncrona
 - Processo continua sua execução, enquanto a E/S é realizada
 - Difícil programação
 - Necessário determinar o término da operação (*signal versus polling*)



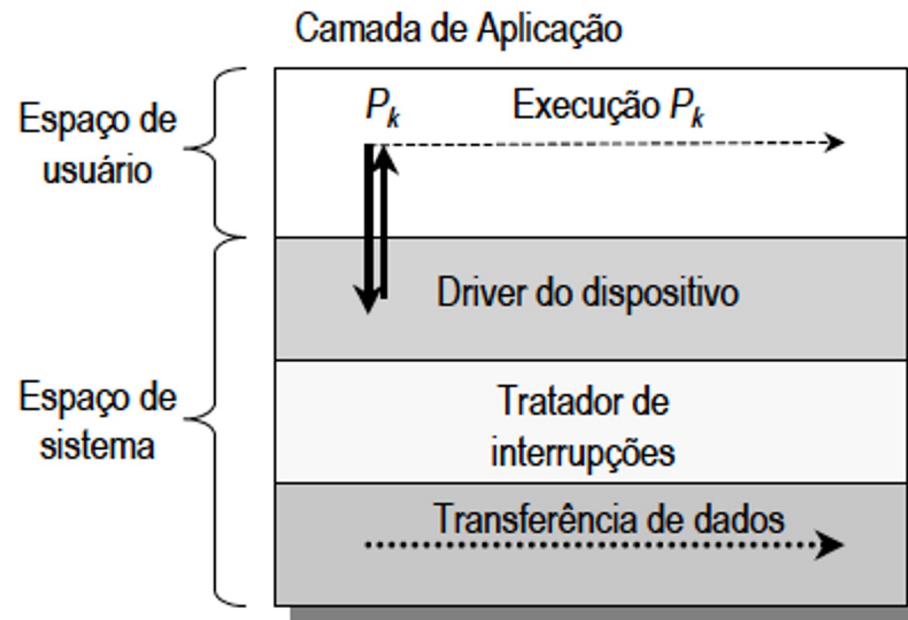
Bloqueante

- O “controle” retorna a camada de aplicação somente após a conclusão da operação de entrada e saída



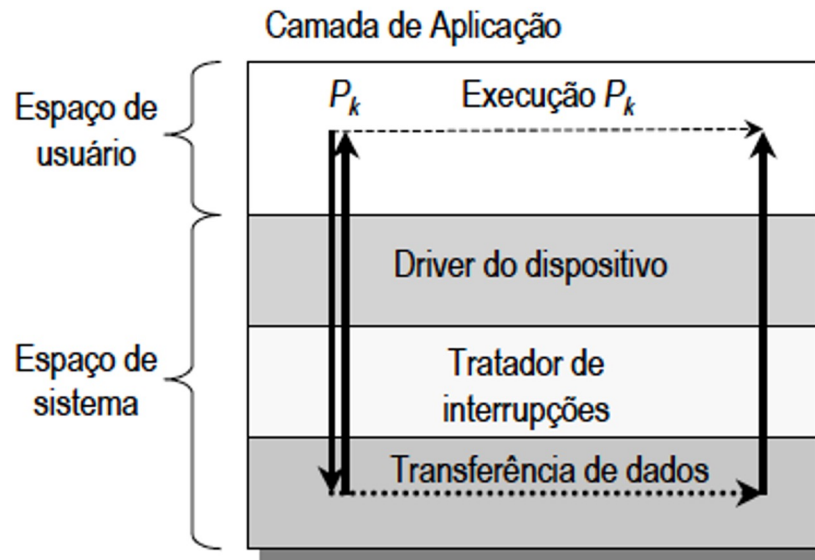
Não bloqueante

- O “controle” retorna a camada de aplicação com os dados disponíveis no momento da operação de entrada e saída



Assíncrona

- O “controle” retorna a camada de aplicação sem que a operação de entrada e saída tenha sido concluída
 - Programação (agendamento) de uma operação de E/S



Referências Bibliográficas

- SILBERSCHATZ, A.; GALVIN, Peter; GAGNE Greg, Operating System Concepts Essentials. John Wiley & Sons, Inc. 2th edition, 2013.
- TANENBAUM, Andrew S. Sistemas operacionais modernos. 3a. ed. São Paulo: Pearson, 2009-2013. p. 653.
- OLIVEIRA, Rômulo; CARÍSSIMI, Alexandre; TOSCANI, Simão. Sistemas Operacionais. Porto Alegre: Bookman, 4a. ed. 2010.



JESUÍTAS BRASIL



Somos infinitas possibilidades