

Ministério da Educação
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca
UNED Nova Friburgo
Bacharelado em Sistemas da Informação

Conceitos de Sistemas Operacionais

Sistemas Operacionais



Prof. Bruno Policarpo Toledo Freitas
bruno.freitas@cefet-rj.br



Objetivos

- **Compreender as principais funções do Sistemas Operacionais**
- **Diferenciar as principais classes de Sistemas Operacionais**
- **Compreender o funcionamento das Chamadas de Sistemas**
- **Introduzir os principais conceitos de sistemas operacionais**
- **Apresentar e diferenciar as principais arquiteturas de Sistemas Operacionais**
- **Apresentar o modelo de execução e criação de programas em C**

Visão geral de um computador



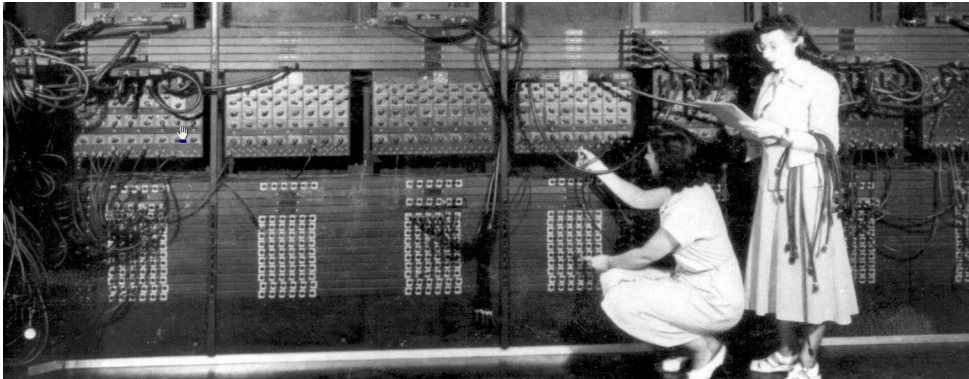
Funções do Sistema Operacional

- **É uma máquina estendida**
 - Esconde os detalhes daquilo das camadas mais baixas
 - Fornece uma máquina virtual, mais fácil de se usar
- **É um gerenciador de recursos**
 - Cada programa utiliza um recurso por um tempo
 - Cada programa possui espaço no recurso

História dos Sistemas Operacionais

- **Primeira geração: 1945 - 1955**

- Válvulas
- Programação em baixo-nível
- Ausência de sistema operacional



ENIAC (1944)

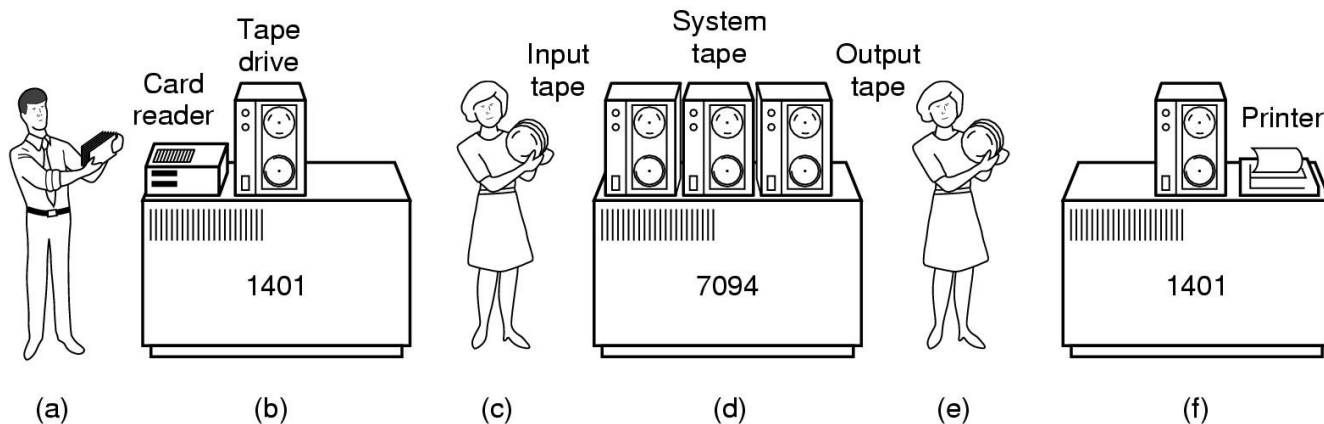


Mark I

História dos Sistemas Operacionais

- **Segunda geração: 1955 - 1965**

- Transístores
- Tarefas são executadas em múltiplos passos
- Perda de tempo entre passos leva a criação de sistemas de lote (*batch*)



História dos Sistemas Operacionais

- **Terceira geração: 1965 - 1980**

- IBM 360 introduz computadores feitos com **circuitos integrados**
- IBM lança famílias de computadores com compatibilidade de software
 - Aplicações científicas x comerciais
 - Gerou S.O. grandes
 - Introduz a **multiprogramação** (*multitasking*)
- CTSS introduz o **compartilhamento de tempo** (*timesharing*)

História dos Sistemas Operacionais

- **Quarta geração: 1980 - 2005**

- (V)LSI : (Very) Large Scale Integration
 - Arquitetura aberta do IBM-PC: Computadores são criados por componentes de diversos fabricantes
- Primeiros Sistemas Operacionais para computadores pessoais
 - CP/M → QDOS → MS-DOS → Windows
 - UNIX → Minix → GNU/Linux
 - Apple: Lisa → Macintosh (clássico) → MacOS
 - Introduziram a **Interface Gráfica** (Graphical User Interface, GUI)

Lisa e Macintosh (1981 - 1985)

- **Após a visita ao Xerox Parc, a Apple investe em computadores com Interface Gráfica (GUI)**
- **Primeiro modelo: Lisa - 1983**
 - CPU Motorola 68000 @ 5 Mhz
 - Preço: USD 9.995,00
 - Desenvolvido pelos “Corporate Shirts”
- **Segundo modelo: Macintosh - 19**
 - CPU: Motorola 68000 @ 8 MHz
 - Preço: USD 2.495,00
 - Desenvolvido pelos “Job’s pirates”
 - Monitor monocromático 9”
- **Em 1985:**
 - Macintosh + LaserWriter+PageMaker = Desktop Publishing



MacOS X

- **Em 1999, lançado o Mac OS 9**
 - Ainda Multi-tarefa cooperativo
 - O último dos “Classic” Mac OS
 - Desenvolvido somente até 2002
 - Substituído pelo Mac OS X, com kernel Mach (Unix)
- **Mac OS X**
 - Desenvolvido desde 1999, lançado em 2001
 - Kernel Mach
 - 10.0 – Cheetah (2001)
 - 10.1 – Puma (2001-2002)
 - 10.2 – Jaguar (2002-2003)
 - 10.3 – Panther (2003-2005)
 - 10.4 – Tiger (2005-2007)
 - 10.5 – Leopard (2007-2009)
 - 10.6 – Snow Leopard (2009)



Windows 3.x

- **Windows 3.0 - 22 maio 1990**

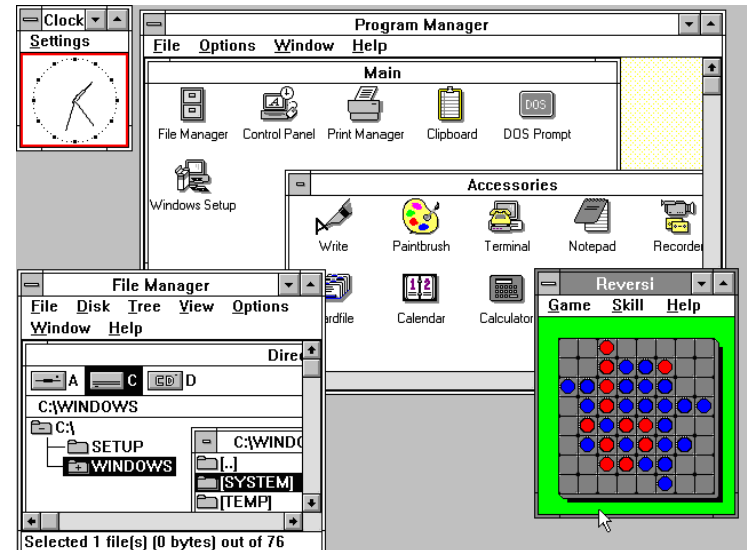
- Suporte ao processador 386
- Modos de operação
 - Real (8086, usa 640 KB de memória)
 - Standard (80286, usa 16 MB de memória XMS)
 - 386 Enhanced (80386, com memória virtual)
- Program Manager (executáveis)
- File Manager (arquivos de dados)

- **Windows 3.1**

- Suporte a fonts True Type

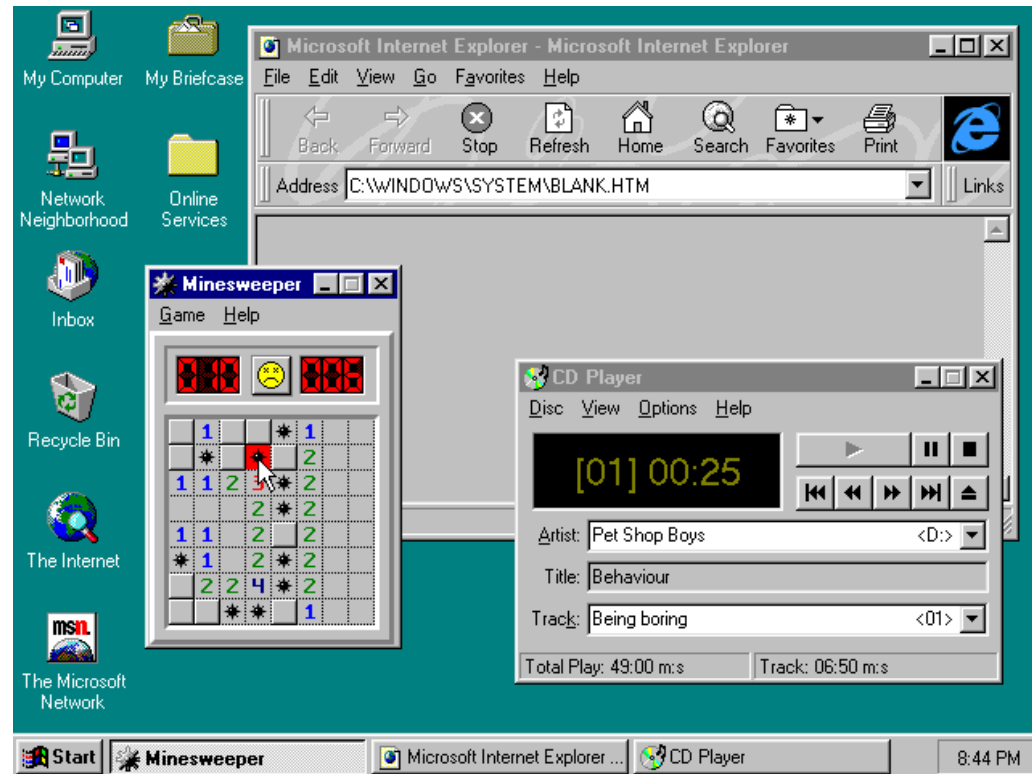
- **Windows 3.11**

- “Service Pack” para 3.1



Windows 95

- **Windows 95 - 24 agosto 1995 (16 bits) - Chicago**
- **O início da “Era moderna” e o grande sucesso da Microsoft**
- **Nomes longos**
- **FAT 32**
- **GUI atual**
- **Sucessores:**
 - Windows 98
 - Windows Me



Vídeo: comercial do Windows 95

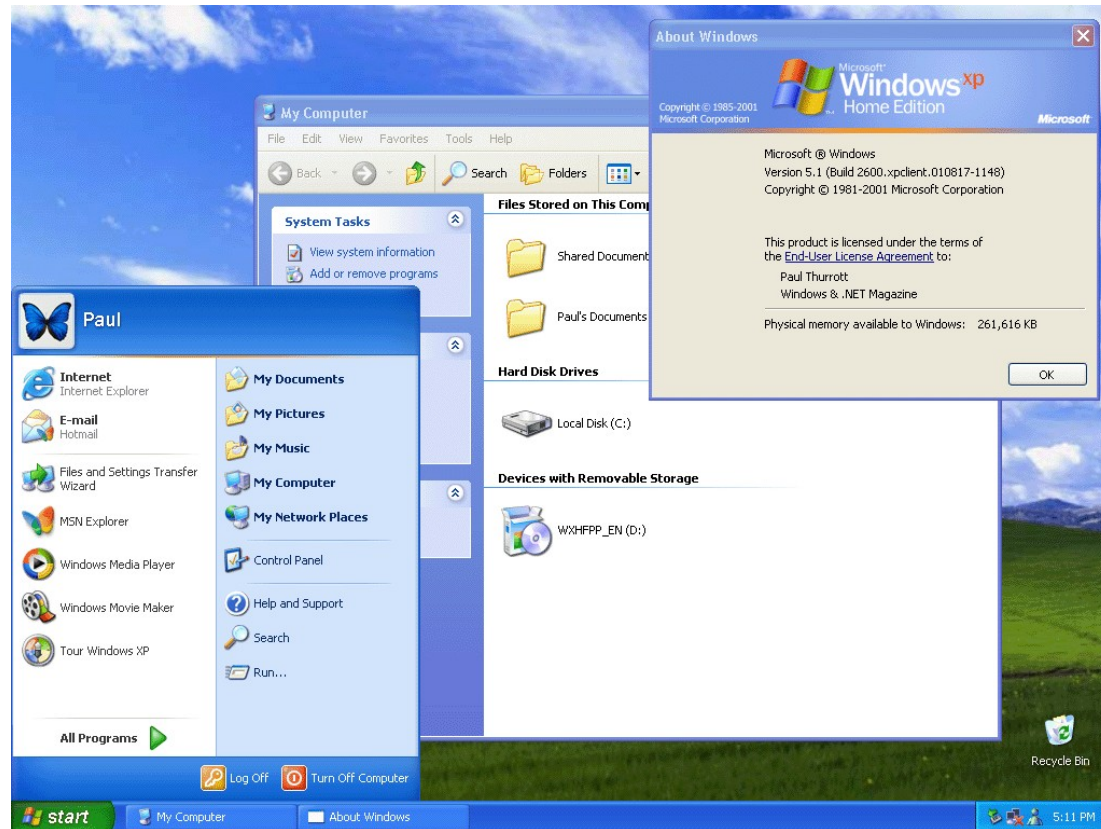
Vídeo: apresentação do Windows 98

Windows NT

- **Windows NT 3.1 - 27 julho 1993 (32 bits) - NT OS/2**
- **Desenvolvimento “portável”**
 - Plataformas:
 - I860 (CPU Intel, nunca lançada)
 - X86
 - MIPS
 - Alpha (DEC)
- **Sistema operacional de 32 bits**
- **Suporte a aplicações de 16 bits: WoW (Windows on Windows)**
- **Windows NT 3.5 - setembro 1994**
- **Windows NT 4.0 - agosto 1996**

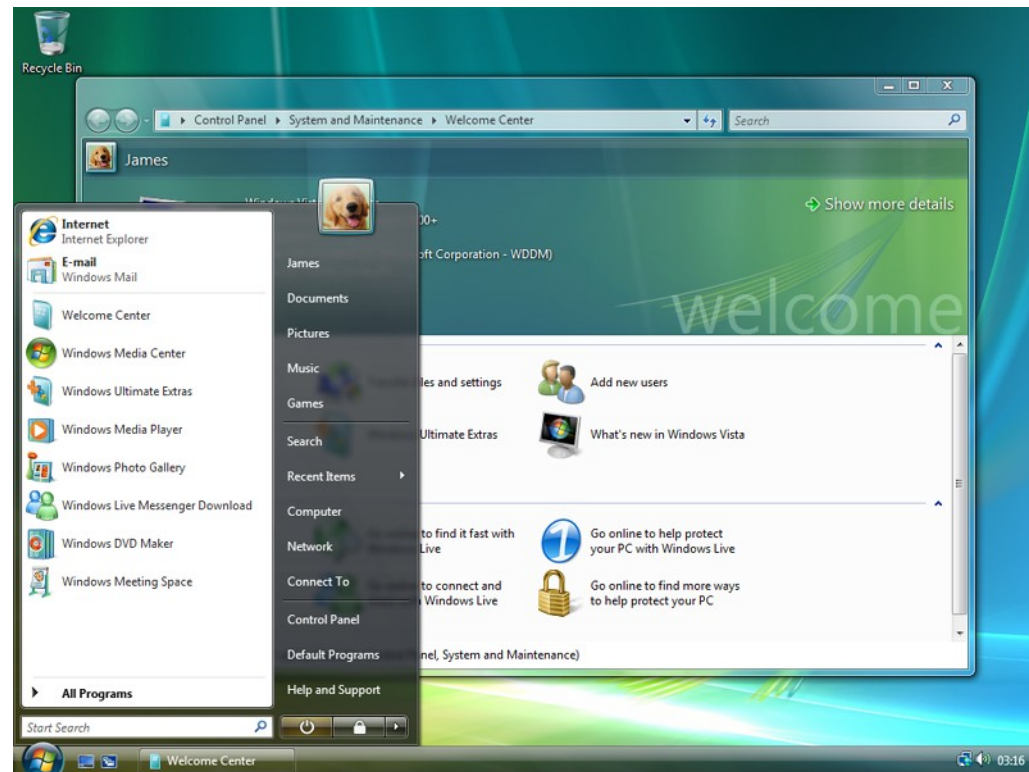
Windows XP

- **Lançamento: 25 outubro 2001 (32 bits) - Whistler**
- **Windows “eXPerience”**
- **Fusão da linha 9x e da linha NT**
- **Características:**
 - Service Packs
 - Ativação
 - Efeitos visuais
 - NTFS



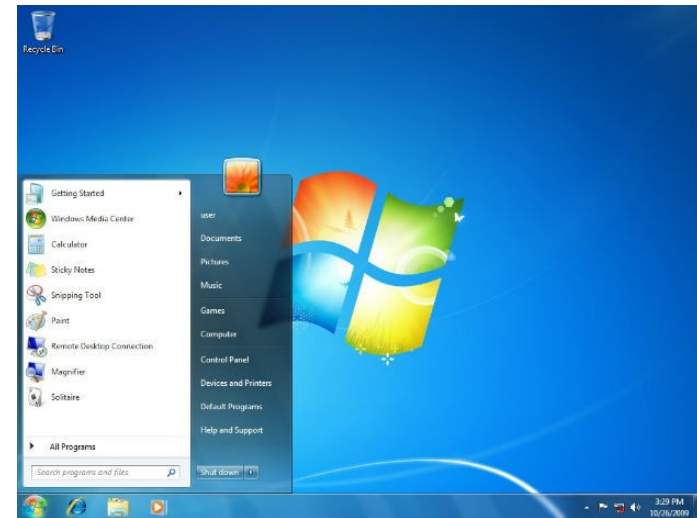
Windows Vista

- **Lançamento: 30 novembro 2006 (32/64 bits) - Longhorn**
- **Versões em 32/64 bits**
- **GUI “Aero”**
- **Ênfase em segurança**
 - UAC (User Account Control)
 - BitLocker
- **Muito “pesado” com as características “premium”**
- **Problemas de performance e compatibilidade de hardware**



Windows 7

- **Lançamento: 22 de julho de 2009**
 - 32b/64b
- **Versão refinada do Windows Vista**
 - Aumento de desempenho
- **Nova barra de tarefas (pin apps)**
- **Bibliotecas**
- **Windows Power Shell**
- **2020: 25% dos PCs no mundo ainda o utilizam**



Windows 8

- **Características**

- Interface Metro para Tablets
- Search global
- Windows Store
- Rework do Task manager
- USB 3.0
- Processadores ARM
- Pouco sucesso comercial



Windows 10

- **Características**

- Interface volta as origens
- Navegador Edge
- Cortana
- Windows Subsystem for Linux (WSL)
- Biometria
- “Spyware”



<https://arstechnica.com/gadgets/2015/07/review-windows-10-is-the-best-version-yet-once-the-bugs-get-fixed/>

Windows 11

- **Obrigatório login por conta Microsoft**
- **Limitação rigorosa de hardware**
- **Microsoft Teams integrado**
- **PowerShell descontinuado**
 - “MegaTerminal” Windows
- **Pesado**



<https://arstechnica.com/gadgets/2021/10/windows-11-the-ars-technica-review/>

História dos Sistemas Operacionais

- **Quinta geração: 2005-2011**
 - Smartphones
 - Sistemas operacionais são específicos
 - iOS → Mac OS X “leve”
 - Android → Linux “leve”

Os eletrônicos portáteis (2007 - presente)

- **iPhone**

- Lançado em 2007
- Modelo 3G lançado em 2008
- Modelo 3Gs lançado em 2009

- **iPod Touch**

- Lançado em 2007
- Processador: ARM 11 620 MHz

- **iPad**

- Lançado em 2010
- Sistema operacional iPhone OS 3.2
- Processador: Apple A4 @ 1 GHz

- **Diversos wearables**

- iWatch



<https://www.youtube.com/watch?v=x7qPAY9JqE4>

História dos Sistemas Operacionais

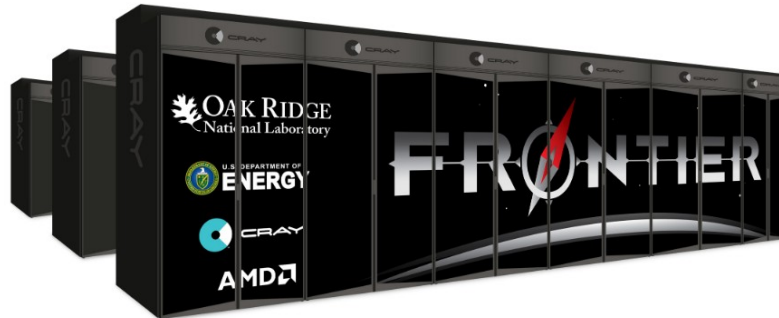
- **Sexta geração: 2011-presente**
 - “Internet das coisas”
 - Sistemas computacionais são utilizados em aplicações não-computacionais em larga escala
 - Sistemas operacionais embarcados: são específicos para suas aplicações

Classificações de Sistemas Operacionais

Mainframes

- Antigos computadores clássicos
- Hoje, supercomputadores

- <https://www.top500.org/>



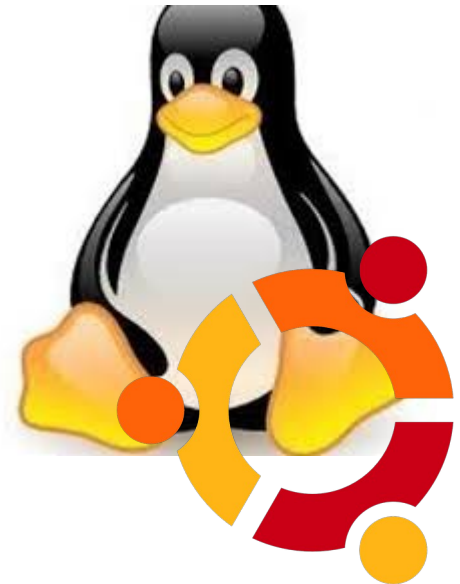
TensorFlow
BigDL for Apache Spark
PyTorch
MXNet
Keras
MLib
scikit-learn
OpenCV

https://www.olcf.ornl.gov/wp-content/uploads/2019/05/frontier_specsheets.pdf

Classificações de Sistemas Operacionais

Sistemas de propósito geral

- **Sistemas operacionais pessoais clássicos**
- Quando usados para trabalho, geralmente são chamados de *Workstations*



Classificações de Sistemas Operacionais

Sistemas de servidores

- **“Servem” funcionalidades computacionais a usuários finais**
 - Web, e-mail, sistemas de arquivos remotos, etc.
 - Consequentemente, não possuem várias aplicações de propósito geral
- **Sistemas na nuvem utilizam geralmente virtualização e barramentos específicos**
- **Segurança é um fator muito importante**



Classificações de Sistemas Operacionais

Sistemas operacionais embarcados

- Contém apenas funcionalidades necessárias para funcionamento de uma aplicação específica não-computacional
- Geralmente distribuições GNU/Linux customizadas criadas com auxílio de ferramentas específicas
 - Buildroot
 - Yocto



Classificações de Sistemas Operacionais

Sistemas operacionais de *smartphones*

- Subclasse de sistemas embarcados
- Hoje se assemelham a computadores pessoais minituarizados (que também fazem ligações!)

ios



Classificações de Sistemas Operacionais

Sistemas de tempo real

- **Sistemas com tarefas que devem cumprir prazos**
- **Kernel deve ser preemptável**
- **GNU/Linux: necessário compilar extensões de Kernel**
 - RTAI, Xenomais, VxWorks, RTEMS etc.
 - Versões atuais do kernel Linux contém PREEMPT_RT

<https://www.rtems.org/>



<https://arstechnica.com/features/2020/10/the-space-operating-systems-booting-up-where-no-one-has-gone-before/>

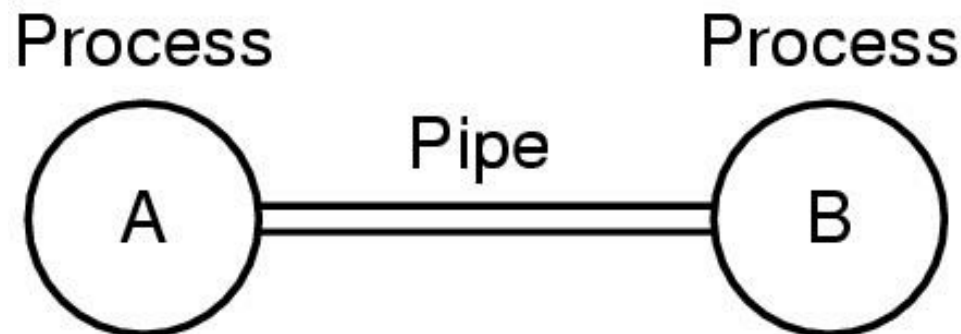
Processos

- Cada programa executado é um processo que compete pelo processador
- Constituem a unidade mínima de abstração do sistema operacional

Processos

Comunicação entre processos

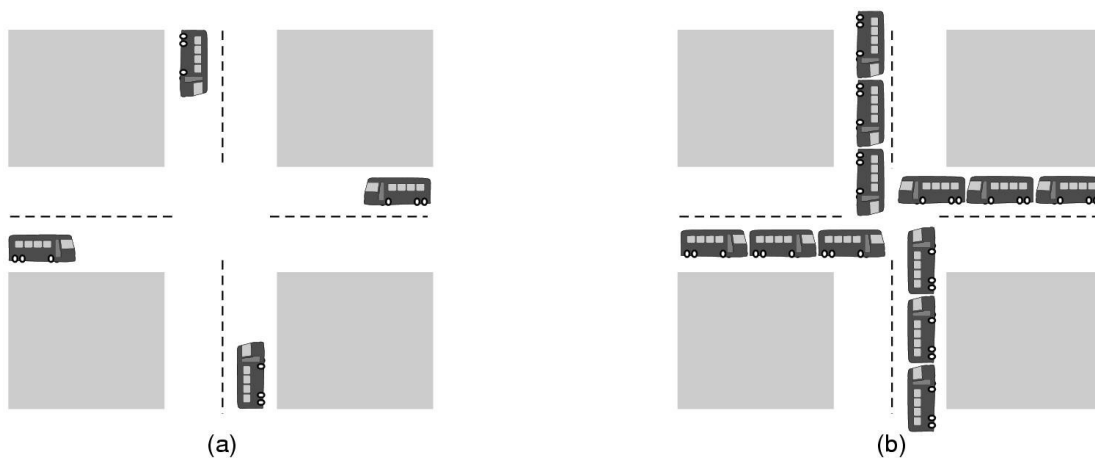
- **Processos podem comunicar-se entre si, enviando, ou simplesmente compartilhando dados de outros processos**
 - Comunicação entre Processos (*Inter Process Communication*, ou IPC)
- **Existe vários métodos para realizar IPC**
 - Memória compartilhada, mensagens, sockets, RMI, RPC ...
- **Pipes são uma forma de comunicação entre processos em UNIX**



Processos

Condições de corrida e Deadlocks

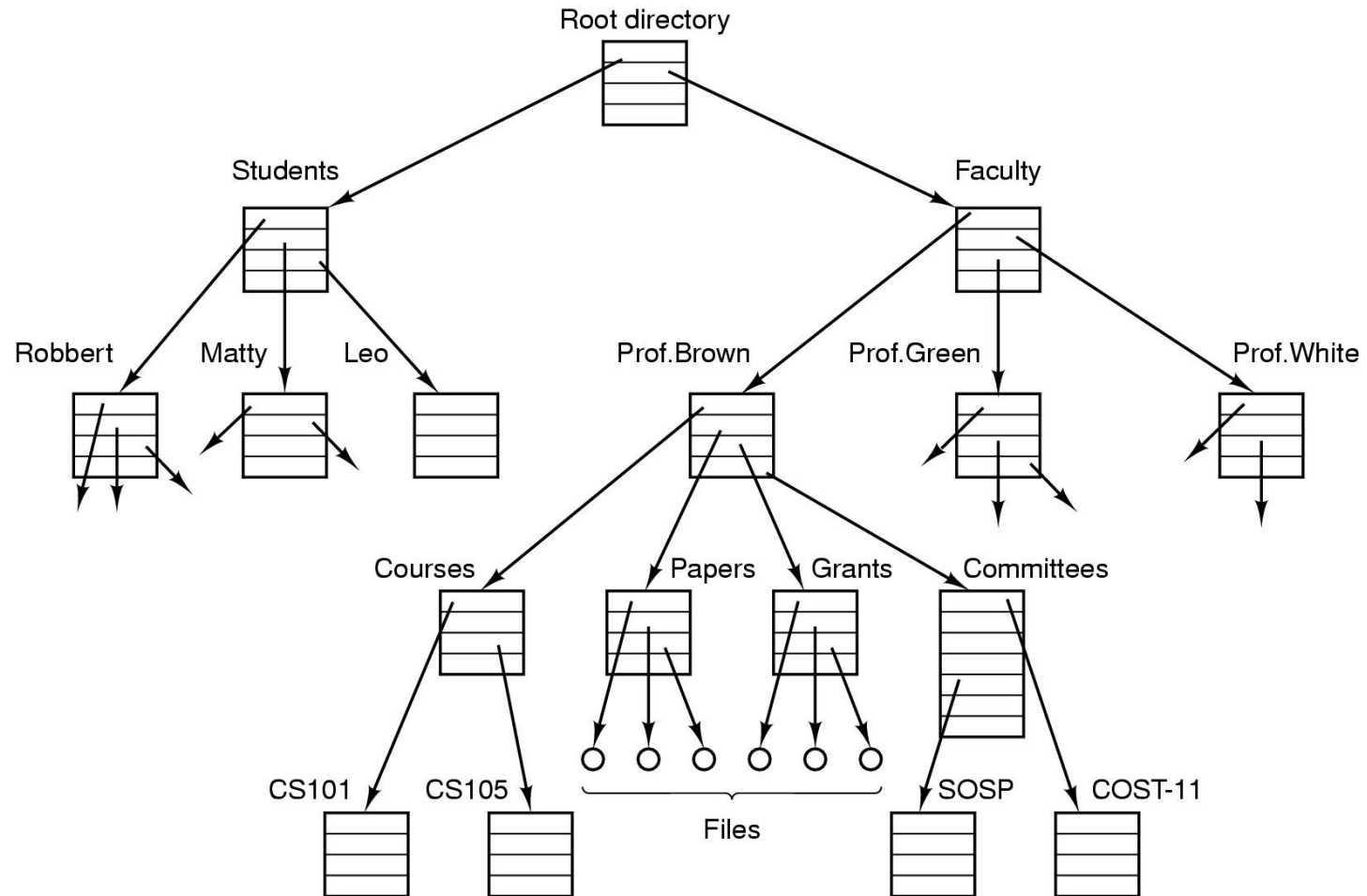
- Quando mal programada, a comunicação entre processos pode gerar erros muito difíceis de se corrigir:
 - Condições de corrida
 - Deadlocks
- Foco de estudo em Programação Concorrente



Sistemas de arquivos

- **Dispositivos de armazenamento armazenam dados em *blocos***
 - Hoje, geralmente 4096 Bytes (4 KiB)
- **Arquivos são compostos por 3 partes principais:**
 - Blocos
 - Índices
 - Metadados

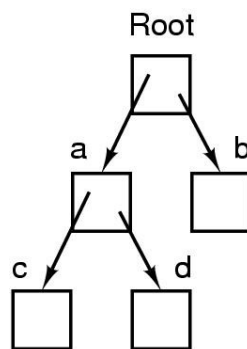
Sistemas de arquivos



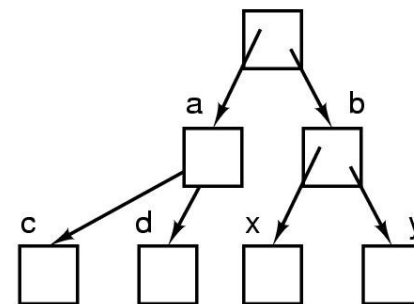
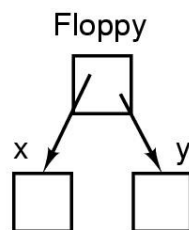
Sistemas de arquivos

Montagem

- **Conceito de sistemas *NIX**
- **A montagem visa unificar a visão de sistemas de arquivos diferentes**
- **Feita em cima do sistema de arquivos raiz (/)**



(a)



(b)

Chamadas de Sistema

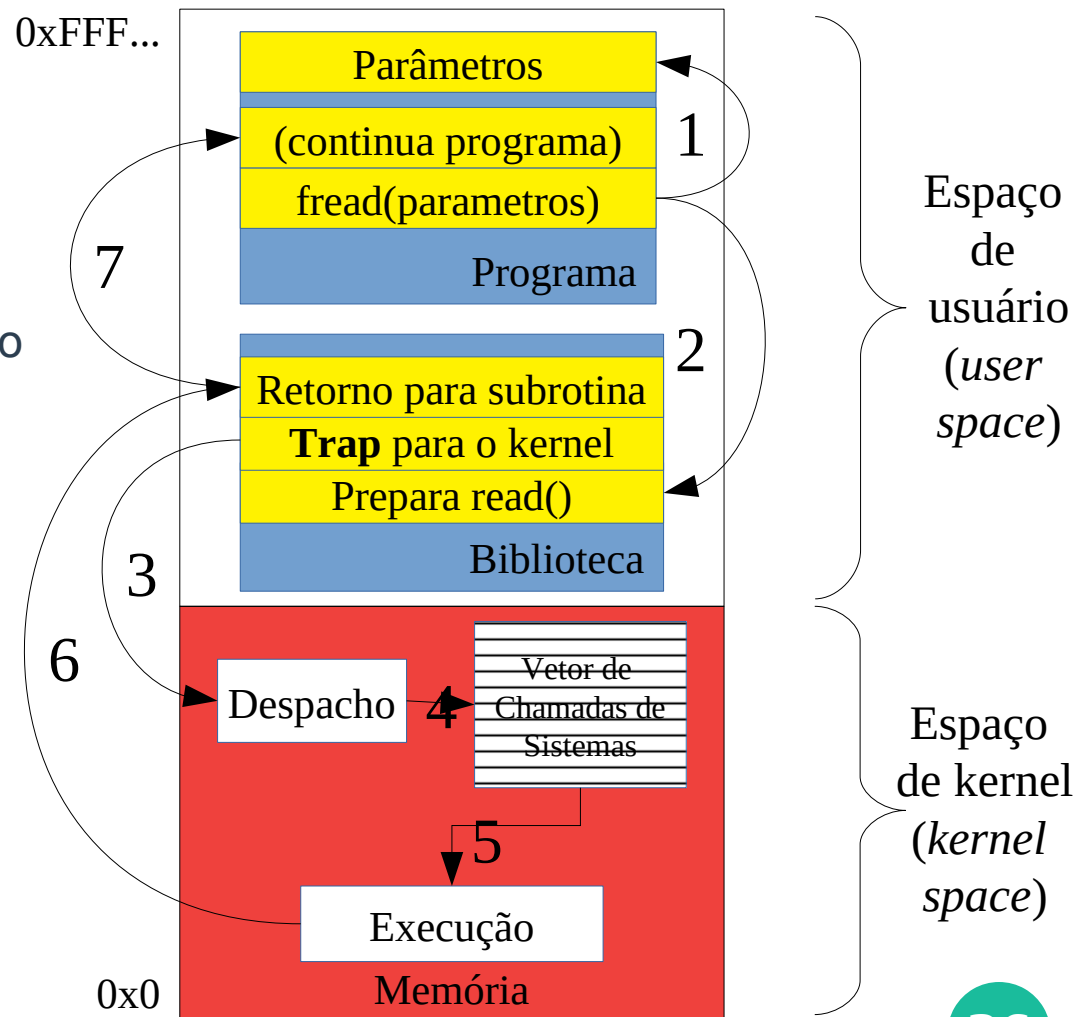
- Para executar um serviço do sistema operacional, os processos precisam passar do modo de usuário para o modo de kernel
- A interface de programação (API) para passar de um modo ao outro constituem as chamadas de sistema
- Chamadas de sistema são padronizadas a fim de permitir a compatibilidade binária entre programas

Chamadas de Sistema

Funcionamento

- Chamada de sistemas **read()**

- Salva parâmetros de *fread()* na pilha do processo
- Desvia para dentro da função *fread()* e prepara chamada de sistema *read()*
- Trap para o kernel
- Despacho da chamada da sistema
- Execução da chamada de sistema
- Retorna para o espaço de usuário, dentro da biblioteca
- Retorna para o processo



Chamadas de sistema POSIX (UNIX)

Gerência de processos

Chamada	Descrição
<code>pid = fork()</code>	Cria um processo-filho igual ao processo-pai
<code>Pid = waitpid(pid, &status, options)</code>	Espera um processo filho terminar
<code>s = execve(name, argv, environment)</code>	Substitui a imagem de um processo
<code>exit(status)</code>	Termina um processo com o código <i>status</i>

Chamadas de sistema POSIX (UNIX)

Sistema 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 de um arquivo em <i>buffer</i>
<code>N = write(fd, buffer, nbytes)</code>	Escreve <code>nbytes</code> bytes de <code>buffer</code> no arquivo e incrementa o ponteiro de acordo
<code>position = lseek (fd, offset, whence)</code>	Movimenta o ponteiro do arquivo
<code>S = stat(name, &buf)</code>	Obtém informações de status

Chamadas de sistema POSIX (UNIX)

Diretórios

Chamada	Descrição
S = mkdir (nome, modo)	Cria um novo diretório
S = rmdir(nome)	Remove um diretório vazio
S = link (nome1, nome2)	Cria uma nova entrada, nome2, apontado o nome1
S = unlink (nome)	Remove uma entrada de diretório
S = mount(dispositivo, nome, flags)	Monta um sistema de arquivos no sistema de arquivos raiz
S = umoun(dispositivo)	Desmonta o sistema de arquivo

Chamadas de sistema POSIX (UNIX)

Outros

Chamada	Descrição
S = chdir (diretório)	Muda o diretório de trabalho
S = chmod (arquivo, modo)	Muda as permissões de arquivos
S = kill (pid)	Envia um sinal de controle do sistema operacional para o processo com PID especificado
Seconds = time (& segundos)	Retorna o tempo transcorrido desde 1º de janeiro de 1970

Chamadas de sistema UNIX vs Windows

UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time

Alguma chamadas da API Win32
Sistemas Operacionais

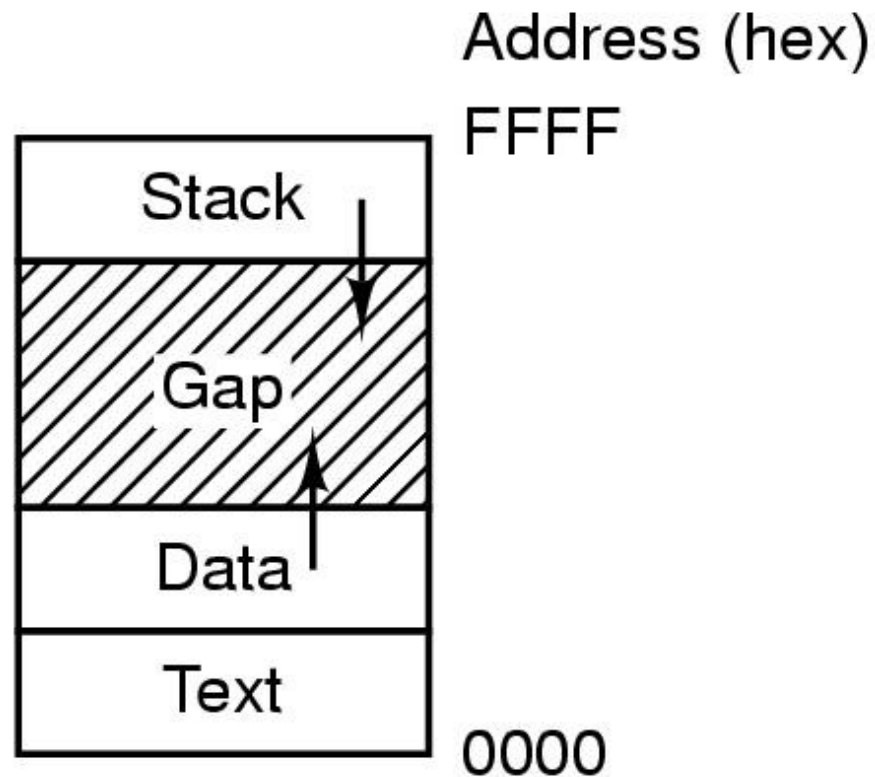
O shell

```
while (TRUE) {                                /* repeat forever */  
    type_prompt( );                            /* display prompt */  
    read_command (command, parameters)        /* input from  
    terminal */  
  
    if (fork() != 0) {                          /* fork off child process */  
        /* Parent code */  
        waitpid( -1, &status, 0);             /* wait for child to exit */  
    } else {  
        /* Child code */  
        execve (command, parameters, 0); /* execute command */  
    }  
}
```

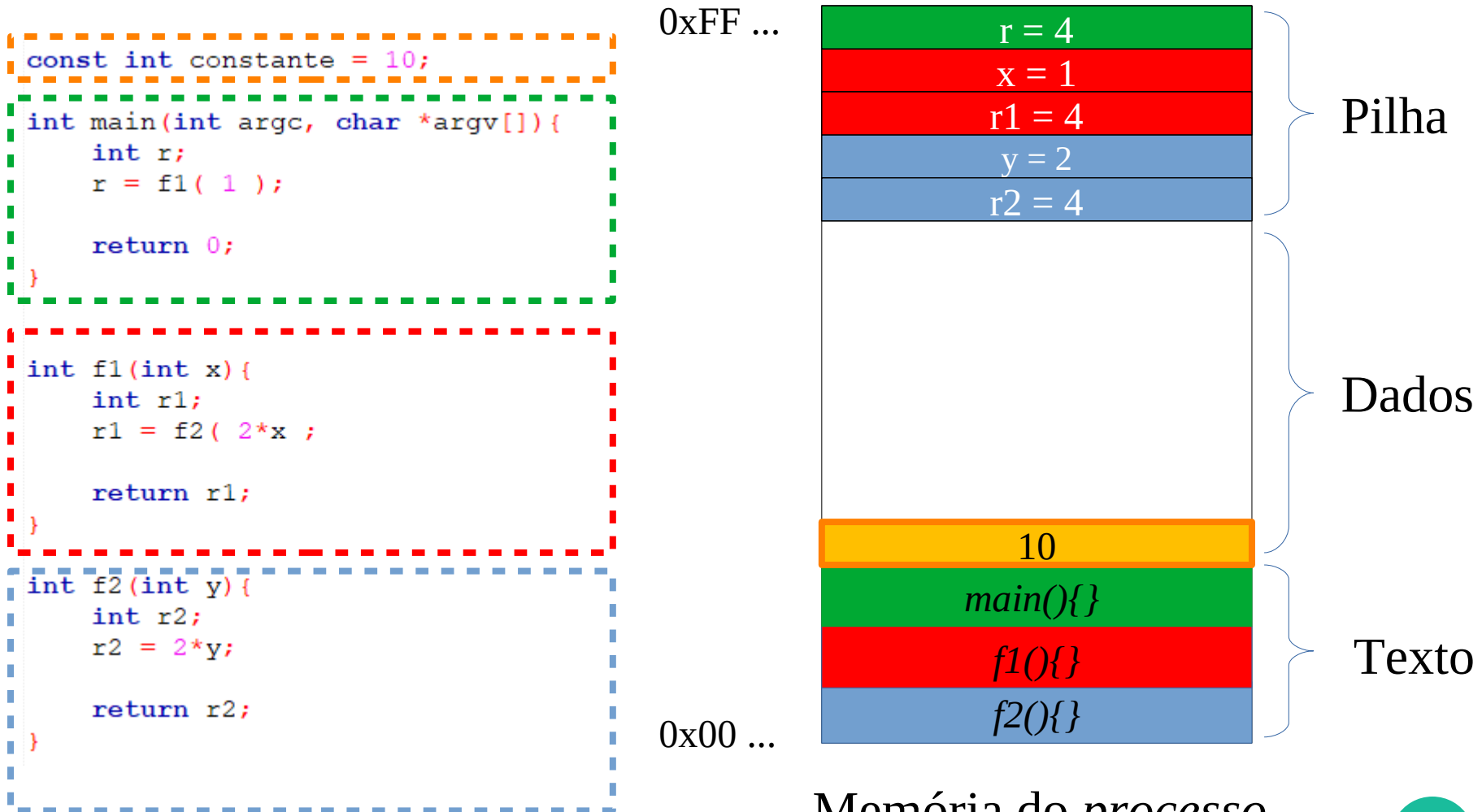
Espaço de endereçamento de processos

- **Processos possuem 3 segmentos de memória principais:**
 - Pilha: Responsável por armazenar parâmetros, variáveis, e valores de retorno de procedimentos e funções
 - Dados: Divididas em 2 partes:
 - Estáticos: constantes e variáveis globais do programa
 - Dinâmicas: memória dinâmica (malloc/free do C)
 - Texto: Código de máquina do programa

Espaço de endereçamento de processos



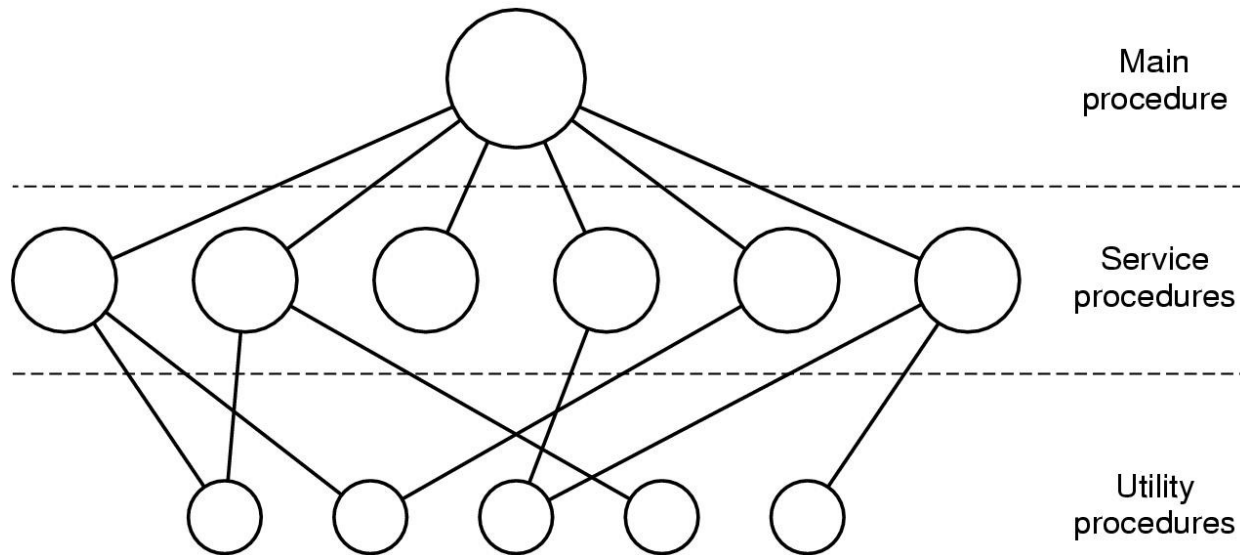
Espaço de endereçamento de processos



Memória do *processo*

Estrutura de Sistemas Operacionais

Kernel monolítico

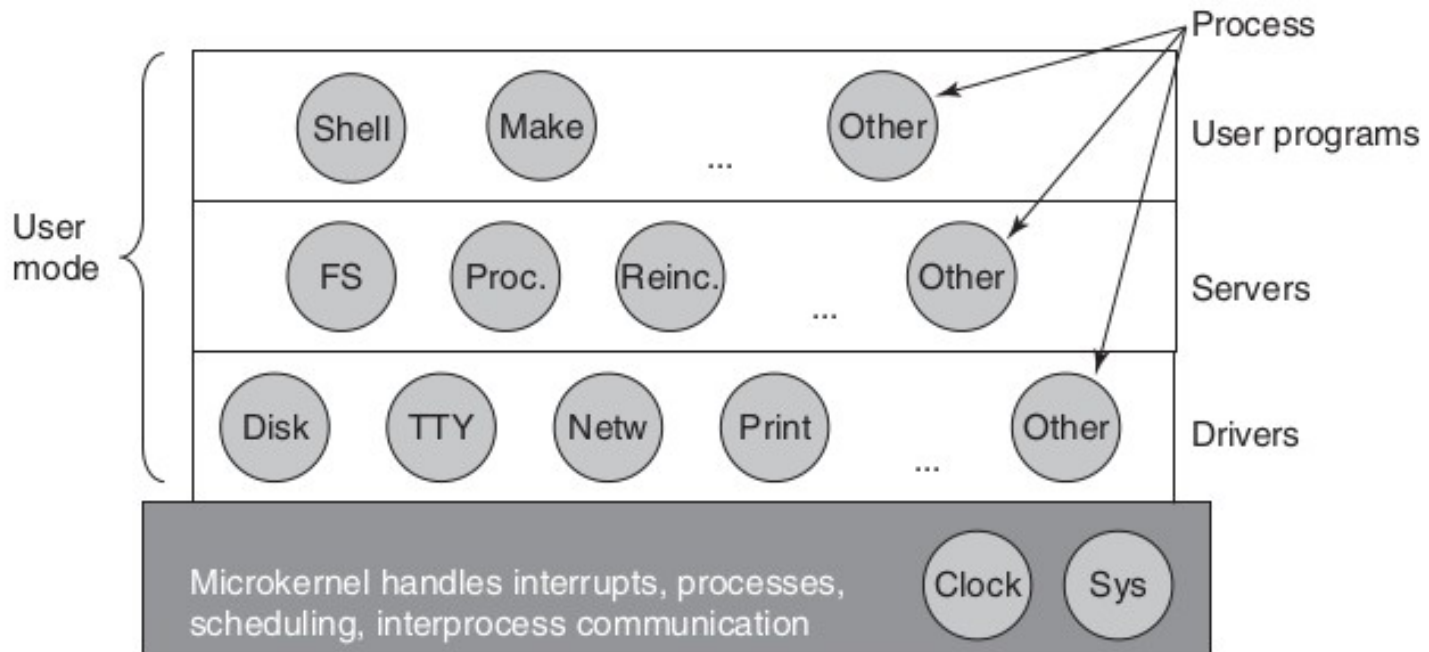


+ Desempenho

- Estabilidade e segurança

Estrutura de Sistemas Operacionais

Cliente-servidor (Microkernel)

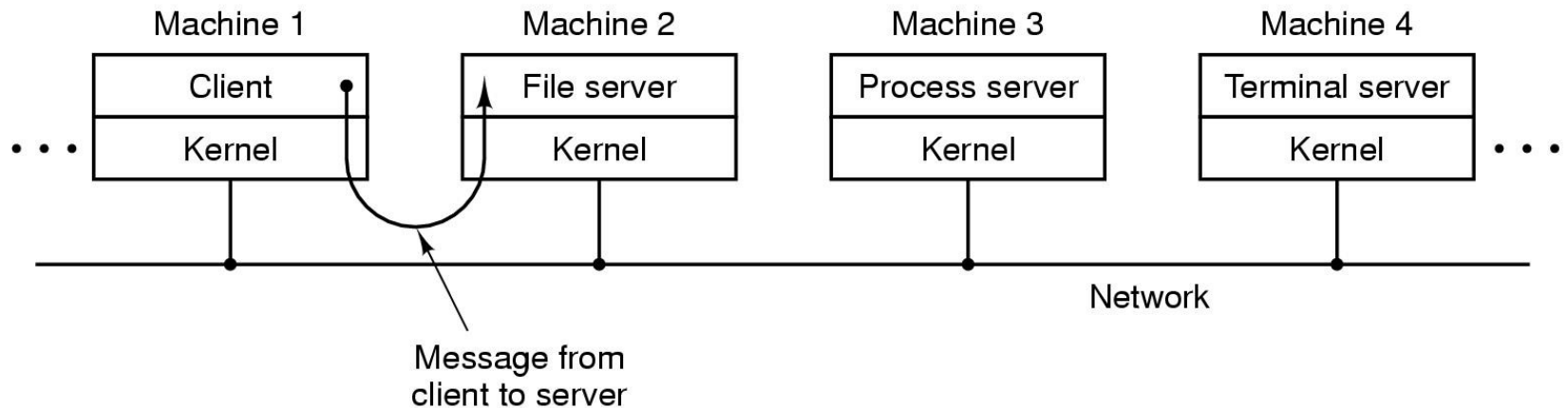


- + **Segurança e estabilidade**
- **Desempenho**

Estrutura de Sistemas Operacionais

Cliente-servidor (Sistema distribuído)

- **Cliente-servidor distribuído**



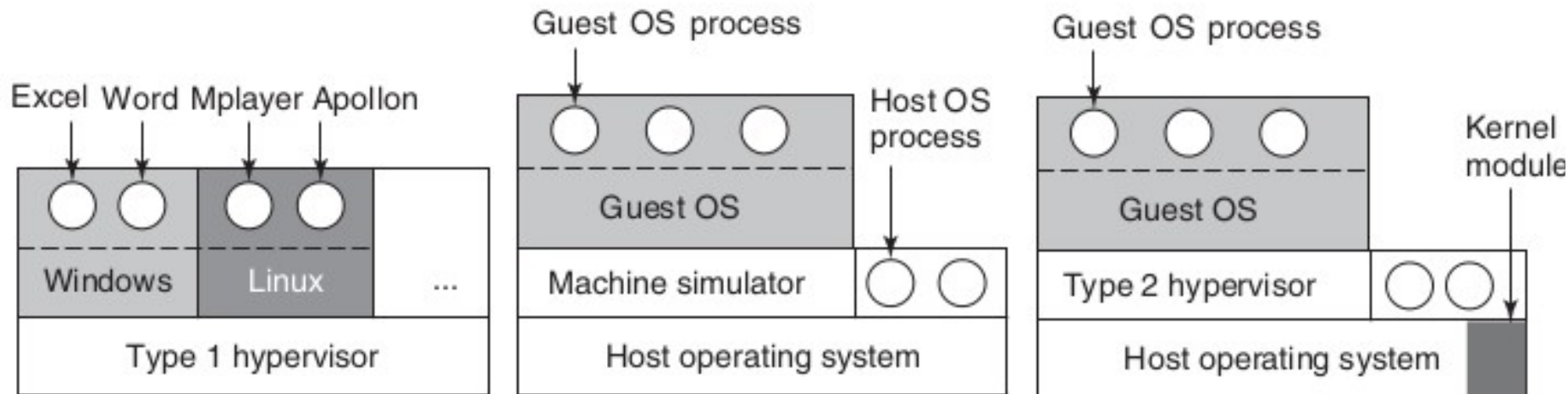
Estrutura de Sistemas Operacionais

Máquinas Virtuais

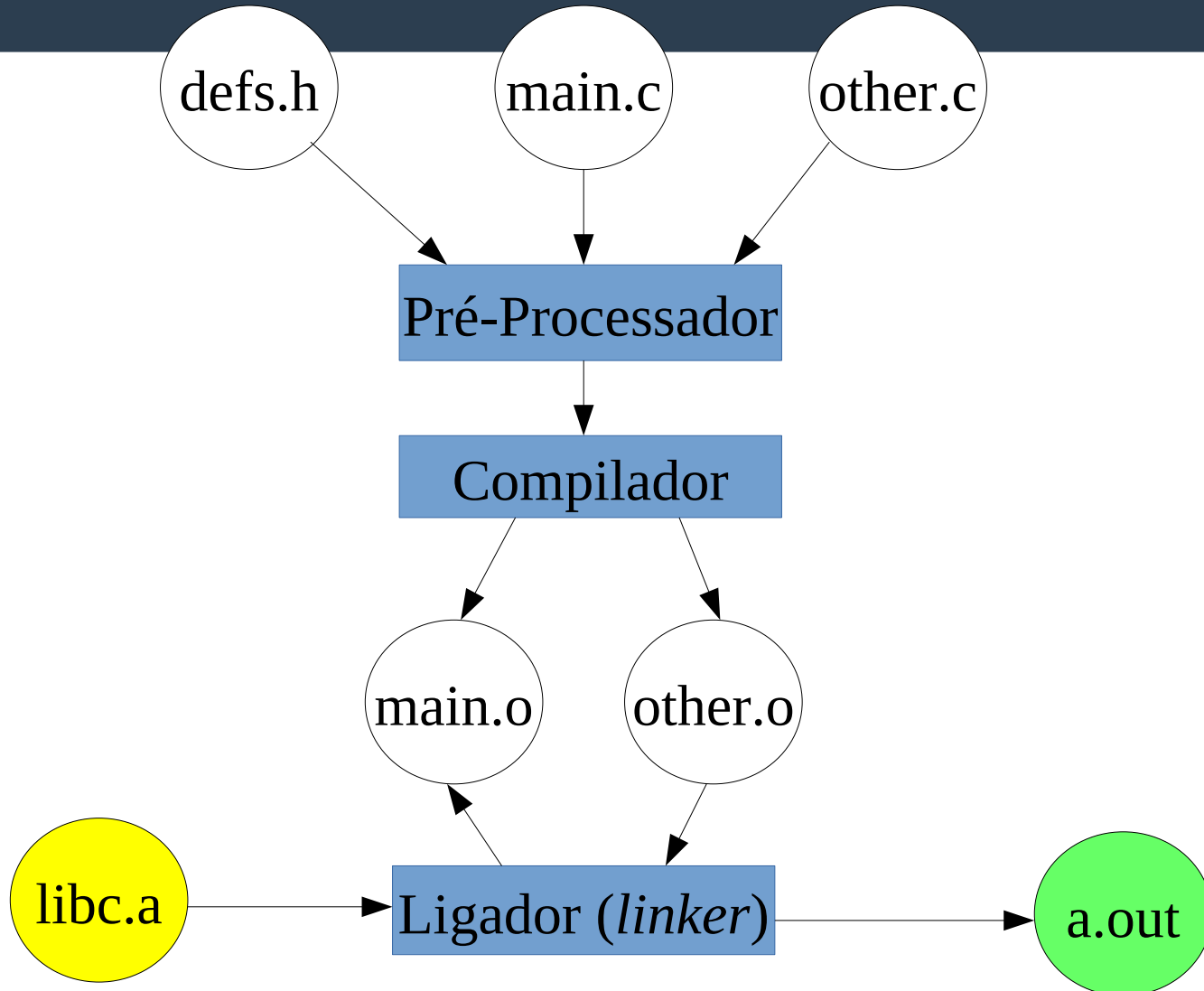
- **Conceito introduzido em (Popek, Goldberg; 1974)**
- ***Traps* desviam para um *monitor de máquinas virtuais* (chamado de *hipervisor*) que provê a ilusão de uma máquina real garantindo:**
 - **Segurança:** a máquina virtual só é acessível pelo hipervisor
 - **Eficiência:** o comportamento do programa executado pela máquina virtual deve ser idêntico ao comportamento executado sobre a máquina real.
 - **Fidelidade:** o código executado pela máquina virtual deve ser executado sem a intervenção do hipervisor
- **Permite a execução de múltiplos sistemas operacionais**

Estrutura de Sistemas Operacionais

Máquinas Virtuais



O modelo de execução C



Referências

- **TANENBAUM, A.S. *Sistemas Operacionais Modernos*. 3ª. ed.**
 - Seções 1.5.1 até 1.5.6, 1.6, 1.7.1 até 1.7.5, 1.8
- **Slides originais do Tanenbaum**
 - http://www.cs.vu.nl/~ast/books/book_software.html

Referências

- Documentário *Triumph of the Nerds*: criação do computador pessoal e história da criação do Macintosh/Apple e Windows/Microsoft
 - Parte 1
 - Parte 2
 - Parte 3
- **Nerdologia: A criação dos computadores pessoais**
- **How the IBM-PC won then lost the Personal Computer market**

Referências

- **Windows 11:**

- <https://arstechnica.com/gadgets/2021/10/windows-11-the-ars-technica-review/>
- <https://arstechnica.com/gadgets/2021/12/microsoft-rolls-out-revamped-notepad-app-to-windows-11-insiders/>
- <https://arstechnica.com/gadgets/2022/05/windows-11s-first-yearly-update-is-almost-done-heres-what-is-and-isnt-part-of-it/>

Exercícios

- **TANENBAUM, A.S. *Sistemas Operacionais Modernos*. 3ª. ed.**
 - Capítulo 1: 1, 5, 7, 8, 9, 10, 12, 13, 14, 17, 22, 25, 26, 29
 - Experiência: Exercício 31 (Faça em uma VM)