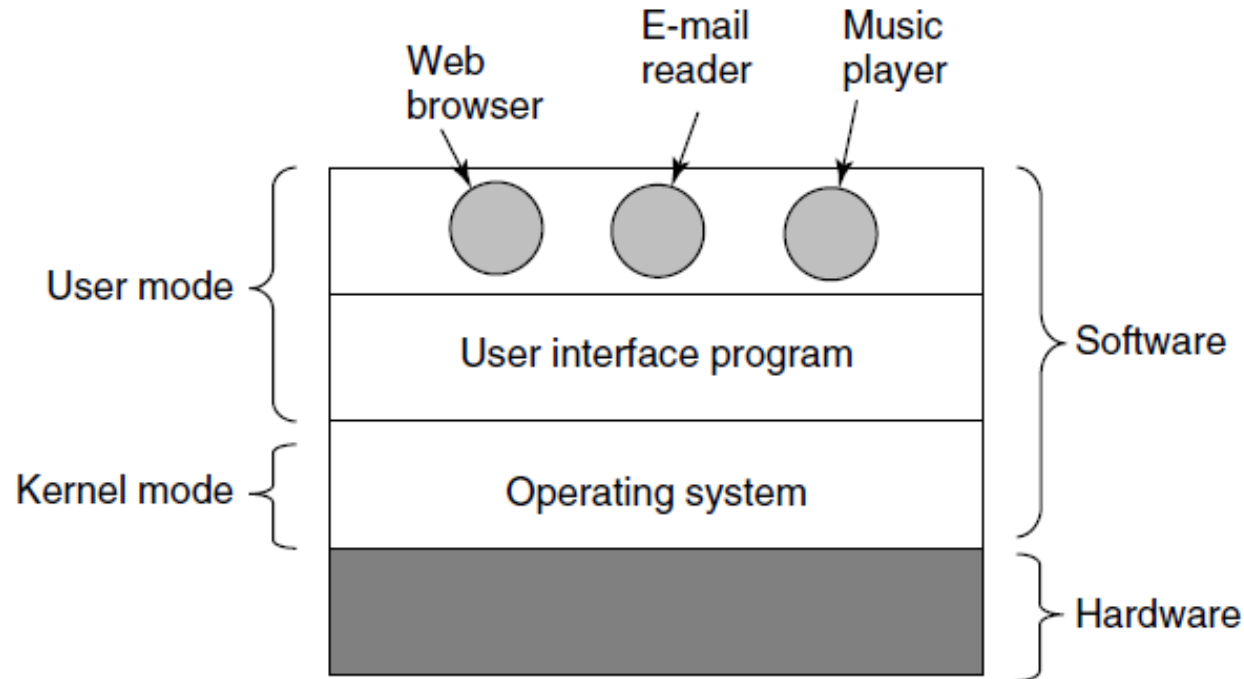


Aula 2 – Boot e Kernel

Sistemas Operacionais

- Elo entre o Hardware e o Usuário



Boot

- SO costuma está em uma memória não volátil
- Carga do SO
 - processo de copiar esse sistema para a memória e transferir o controle da CPU

Boot

- Processo de boot de um sistema baseado na arquitetura Intel x86.
 - BIOS - Binary Input Output System
 - CMOS - A memória não-volátil
 - POST - Power On Self Test
 - Bootloader – MBR (Master Boot Record)

Boot

- Boot process
 - <https://www.youtube.com/watch?v=PSnGuvylWBI>

Boot

- MBR → Registro Mestre de Inicialização
 - conceito de MBR foi introduzido ao público em 1983 com o PC DOS 2.0
 - contém 512 bytes de informação da estrutura organizacional do disco

Boot

- Estrutura do MBR

Estrutura do Master Boot Record :

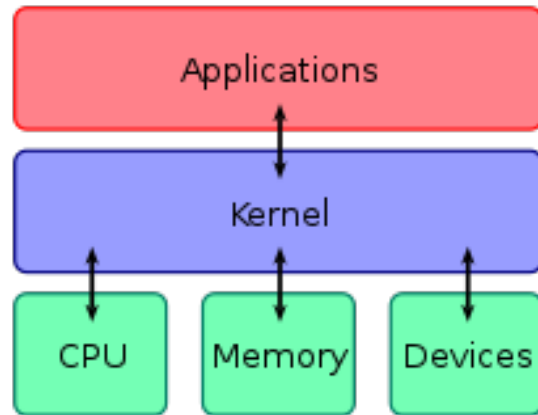
Endereço		Descrição		Tamanho em bytes
Hex	Dec			
0000	0	Código de arranque do SO		440 (max. 446)
01B8	440	Assinatura opcional		4
01BC	444	normalmente nulo ; 0x0000		2
01BE	446	Tabela de partições primarias (Quatro entradas de 16 bytes (IBM Partition Table scheme))		64
01FE	510	55h	MBR signature; 0x55AA	2
01FF	511	AAh		
Tamanho total do MBR : 440 + 4 + 2 + 64 + 2 =				512

Boot

- Trocar de SO
 - a MBR é sobrescrita com um novo gerenciador de boot
- Ter dois ou mais SO's instalados
 - Boot manager
 - Programa instalado na trilha MBR
 - Windows NT/2000/XP traz o NTLDR
 - Linux traz Lilo e o GRUB

Kernel

- Na maioria dos sistemas, é um dos primeiros **programas** carregados no boot



Kernel

- Funções
 - Mediar acesso à CPU
 - Controlar memória RAM
 - armazena as instruções e os dados de programas
 - Gerenciar dispositivos de E/S

Kernel

- Funções
 - Controlar privilégios
 - Usuários e permissões
 - Chamadas de sistema
 - Relacionado a hardware
 -

Kernel

- Chamadas de Sistema

1. Controle de processos

- encerrar, abortar
- carregar, executar
- criar processo, encerrar processo
- obter/definir atributos do processo
- esperar hora, esperar evento, sinalizar evento
- alocar e liberar memória

Kernel

- Chamada de Sistema
 - 2. Gerenciamento de arquivos
 - criar arquivo, excluir arquivo
 - abrir, fechar
 - ler, gravar, reposicionar
 - obter/definir atributos do arquivo

Kernel

- Chamada de Sistema
 - 3. Gerenciamento de dispositivos
 - solicitar dispositivo, liberar dispositivo
 - ler, gravar, reposicionar
 - obter/definir atributos do dispositivo
 - conectar ou desconectar dispositivos logicamente

Kernel

- Chamada de Sistema
 - 4. Manutenção de informações
 - obter/definir a hora ou a data
 - obter/definir dados do sistema
 - obter/definir atributos do processo, arquivo ou dispositivo

Kernel

- Chamada de Sistema

- 5. Comunicações

- criar, excluir conexão de comunicações
 - enviar, receber mensagens
 - transferir informações de status
 - conectar ou desconectar dispositivos remotos

Kernel

- Chamada de Sistema
 - 6. Proteção
 - obter/definir permissões de arquivos

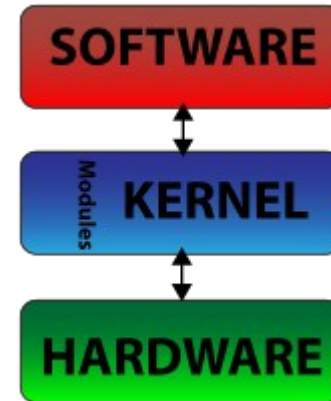
Kernel

- Abordagens de design
 - Kernel monolítico
 - Microkernel
 - Híbridos ou modulares

Kernel

- **Monolítico**

- É um programa único que contém todo o código necessário para executar todas as tarefas do kernel
- Tradicionalmente utilizados pelos sistemas operacionais Unix-like



Kernel

- **Monolítico**
 - Muitas chamadas do sistema são fornecidas
 - Kernels monolíticos modernos (Linux e FreeBSD) possuem a capacidade de carregar módulos em tempo de execução

Kernel

- **Monolítico**
 - Vantagens:
 - Menos software envolvido
 - Menor e mais eficiente
 - Menos bugs
 - Menos problemas de segurança

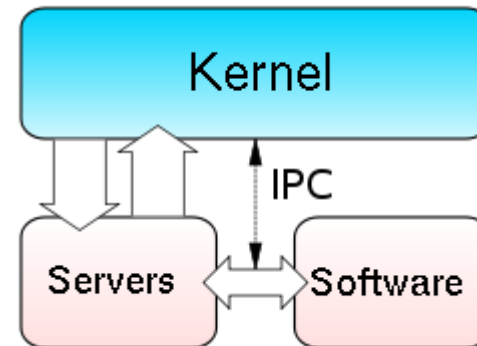
Kernel

- **Monolítico**
 - Desvantagens
 - Erros têm efeitos colaterais fortes
 - Muitas vezes se tornam muito grande e difícil de manter
 - Integração dos módulos as vezes não é simples
 - Não portáteis
 - Específicos para cada nova arquitetura

Kernel

- **Microkernel**

- conjunto de "servidores" que se comunicam através de um kernel "mínimo"
- Alguns sistemas que usam micro kernels são QNX e o Hurd



Kernel

- **Microkernel**

- Somente o que necessita estar em modo privilegiado estão no espaço do kernel
- Servidores ficam no espaço do usuário

Kernel

- **Microkernel**

- Vantagens

- A manutenção é geralmente mais fácil
 - Desenvolvimento facilitado
 - Maior disponibilidade
 - Recuperação da falha de um servidor

Kernel

- **Microkernel**

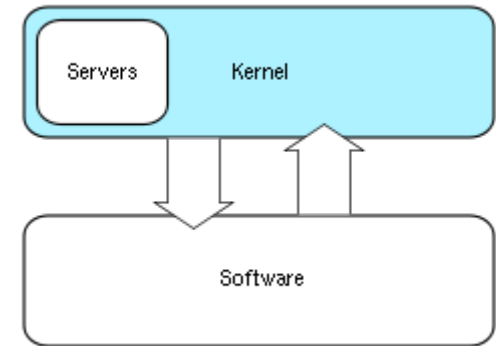
- Desvantagens

- Maior consumo de memória
 - Mais camadas de software
 - Possível perda de desempenho
 - Gestão de processos mais complicada

Kernel

- **Modular ou Híbrido**

- maioria dos sistemas operacionais comerciais
- Microsoft Windows NT 3.1, NT 3.5, NT 3.51, NT 4.0, 2000, XP, Vista, 7, 8, 8.1 e 10.
- MacOS usa um núcleo híbrido chamado XNU que é baseada no OSFMK 7.3 e FreeBSD



Kernel

- **Modular ou Híbrido**

- Semelhantes ao microkernel
- Com execução de alguns serviços no espaço do kernel para reduzir a sobrecarga
 - tais como a pilha de rede ou o sistema de arquivos
- incapazes de carregar módulos em tempo de execução por conta própria

Kernel

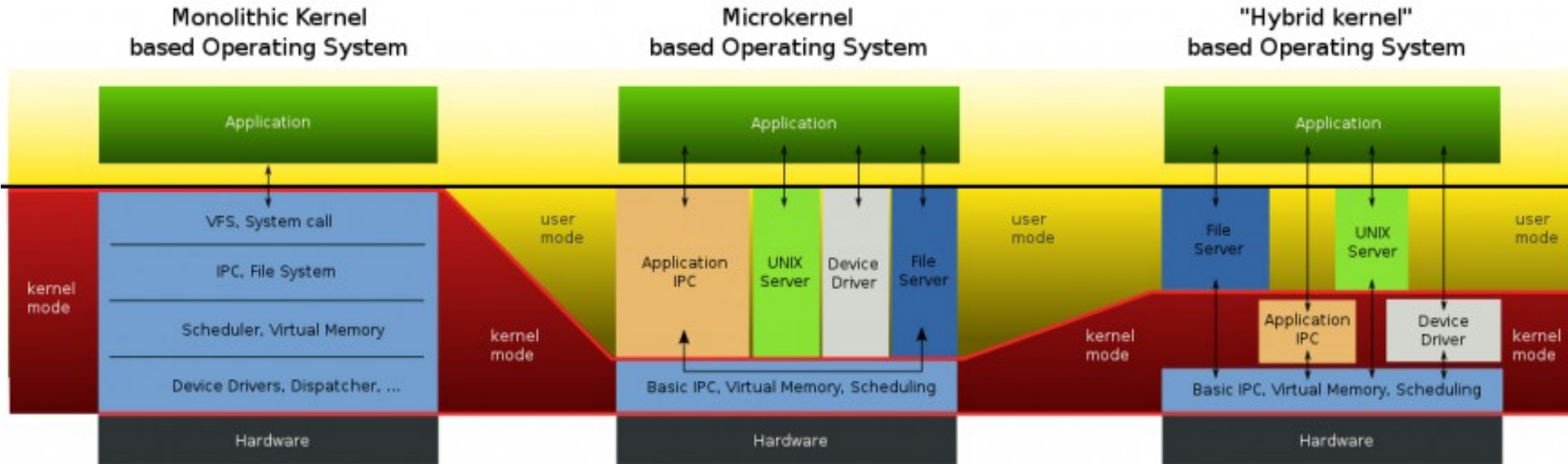
- **Modular ou Híbrido**
 - Vantagens
 - Desenvolvimento facilitado
 - Mais fácil a integração da tecnologia de terceiros

Kernel

- **Modular ou Híbrido**
 - Desvantagens
 - Mais interfaces → mais erros
 - Maior possibilidade de falhas de segurança
 - Manutenção pode ser mais difícil

Kernel

- Comparação



Kernel

- MINIX versus LINUX



Andrew S. Tanenbaum



Linus Torvalds