

Sistemas Operacionais

Estruturas do Sistema Operacional (cap. 3)



Sumário

- Componentes de um S.O.
- Serviços do sistema
- Chamadas de sistema
- Diferentes estruturas de S.O.
- Máquinas virtuais

Vamos saltar as seções 3.7 e 3.8
(noções de projeto, implem. e geração de um S.O.)



Sistemas Operacionais – Estruturas do S.O.

2

Componentes usuais de um S.O.

- Gerência de processos
- Gerência da memória principal
- Gerência de arquivos
- Gerência do sistema de E/S
- Gerência da memória secundária
- Sistemas de rede
- Sistema de proteção
- Sistema interpretador de comandos



Sistemas Operacionais – Estruturas do S.O.

3

Gerência de processos

- Processo: programa em execução
- Precisa certos recursos:
 - CPU, memória, arquivos, dispositivos de E/S
- É função do S.O. controlar esses processos
 - Criação e destruição de processos
 - Suspensão e retomada de execução
 - Garantia de mecanismos para sincronização e comunicação entre processos



Sistemas Operacionais – Estruturas do S.O.

4

Gerência da memória principal

- A memória é um vetor de bytes endereçáveis
- Forma um depósito de dados rapidamente acessíveis, compartilhado por dispositivos e CPU
- Dispositivo de armazenamento **volátil**
- Responsabilidades do S.O. sobre a memória:
 - manter o controle de que partes da memória estão sendo usadas e por quem
 - decidir que processos ganham acesso
 - alocar e desalocar espaço sob demanda



Sistemas Operacionais – Estruturas do S.O.

5

Gerência de arquivos

- Arquivo: coleção de dados relacionados
 - programas (fontes, objetos e executáveis, dados)
- Responsabilidades do S.O. sobre arquivos:
 - Criação e remoção de arquivos
 - Criação e remoção de diretórios (índices)
 - Suporte a primitivas de acesso e manipulação
 - Mapeamento para o armazenamento secundário
 - Cópias de segurança (*backup*)



Sistemas Operacionais – Estruturas do S.O.

6

Gerência do sistema de E/S

- E/S: relacionamento com dispositivos periféricos
- Tarefas relacionadas:
 - Controle dos recursos de *buffers/caches*
 - Definição de uma interface geral para acionamento dos dispositivos
 - Implementação dos acionadores de dispositivos específicos (*device drivers*)

Gerência de armazenamento secundário

- Arm. sec: sistema de armazenamento de dados não volátil (para complementar mem.)
- A maioria dos sistemas modernos usa discos como o principal meio de armazenamento
- Tarefas do S.O. relativas ao armaz. sec.:
 - Controle do espaço livre
 - Alocação do armazenamento
 - Mapeamento de arquivos em discos
 - Escalonamento de discos

Sistema de rede

- Em um sistema distribuído, processadores têm cada um seu clock e sua memória particulares
- A forma de conexão entre eles é através de uma rede de comunicação
- Tarefas do S.O. com relação à rede:
 - controlar o acesso à rede pelos processos
 - garantir a disponibilidade de protocolos comuns
 - oferecer recursos eficientes

Sistema de proteção

- Usualmente, um S.O. garante acesso a diversos programas, processos e usuários
 - é essencial que esses acessos sejam controlados e protegidos contra abusos
- Tarefas do S.O. para proteção:
 - diferenciação de acessos autorizados ou não
 - definição dos controles a serem impostos
 - criação de formas de fazer valer as regras

Sistema interpretador de comandos

- O S.O. deve prover formas de comunicação com os usuários para receber deles as instruções sobre o que fazer
- Interpretadores de comandos podem tomar formas variadas e diferentes nomes:
 - interpretador de linha de comando
 - shell (Unix/Linux)
 - sistemas gerenciadores de janelas

Serviços do Sistema Operacional

- Execução de programas
- Controle de operações de E/S
- Manipulação de arquivos e sistemas de arq.
- Comunicação (tanto local quanto de rede)
- Detecção de erros

Serviços adicionais do S.O.

- Alguns serviços não focam no usuário diretamente, mas garantem a operação eficiente
 - Alocação de recursos
 - Contabilidade
 - Proteção

Chamadas de sistema (sys. calls)

- Definem a interface entre programas em execução e o S.O.
- Implementam o chaveamento de níveis de proteção discutidos anteriormente
- Disponíveis como instruções em linguagem de máquina
- Linguagens de alto nível costumam oferecer bibliotecas que encapsulam tais chamadas

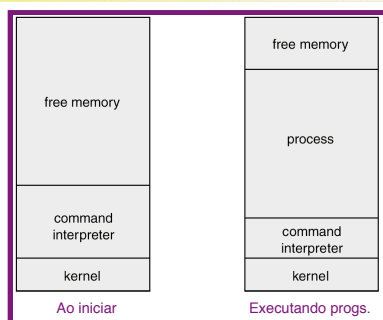
Chamadas de sistema (sys. calls)

- Por se tratar de uma mudança de nível de proteção, a passagem de dados é controlada
- Três métodos são comumente usados para se passar parâmetros entre processos e o S.O.:
 - passagem através de registradores da CPU
 - armazenamento em uma tabela em memória, cujo endereço é passado através de um reg.
 - armazenamento na pilha (*stack*) do programa, de onde o S.O. extrai os dados

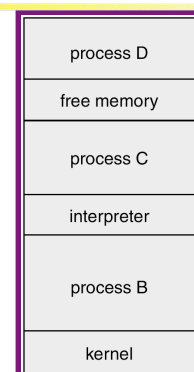
Tipos de chamadas de sistema

- Controle de processos
- Gerência de arquivos
- Gerência de dispositivos
- Manutenção de informações do sistema
- Comunicação

Execução no MS-DOS

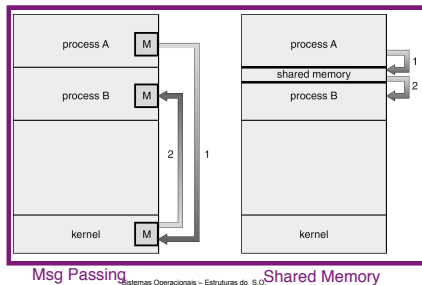


UNIX executando diversos progs.



Modelos de comunicação

- Com. pode se dar pelo envio de mensagens ou pelo compartilhamento de memória



19

Programas do sistema (comandos)

- Um nível de abstração/funcionalidade adicional
- Acabam por definir o S.O. na visão dos usuários
 - Manipulação de arquivos
 - Informações de status do sistema
 - Modificação de arquivos
 - Suporte a linguagens de programação
 - Controle e execução de programas
 - Comunicação
 - Aplicações de uso comum

Sistemas Operacionais – Estruturas do S.O.

20

Estrutura do S.O.

- Sistemas operacionais estão entre os sistemas de software mais complexos existentes
 - No passado, eram escritos como um único programa, em linguagem de máquina
 - O aumento da complexidade levou à busca por outras formas de organização interna
 - Camadas
 - Módulos
 - Micro-kernel
 - Máquinas virtuais

Sistemas Operacionais – Estruturas do S.O.

21

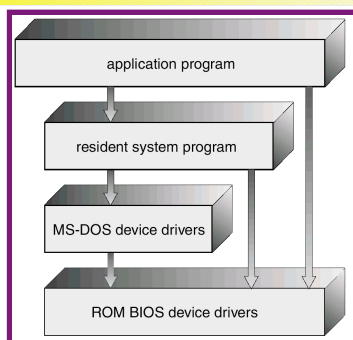
Estrutura do MS-DOS

- Escrito para microprocessadores simples, com poucos recursos de proteção
 - máximo de funcionalidade no menor espaço
 - não é dividido em camadas ou módulos
 - apesar de ter alguma estrutura, interfaces e níveis de funcionalidade não são bem definidos nem isolados (inclusive por limitações dos computadores em que executava)

Sistemas Operacionais – Estruturas do S.O.

22

Estrutura de camadas do MS-DOS



Sistemas Operacionais – Estruturas do S.O.

23

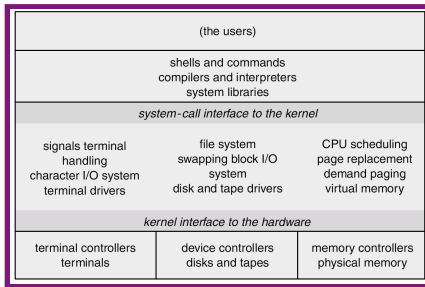
Estrutura do S.O. Unix

- O Unix original foi talvez o primeiro S.O. a ser escrito em uma linguagem de "alto nível", C
- Originalmente limitado pelo HW da época, com o tempo foi crescendo, gerando versões diferentes
- O Linux é derivado conceitualmente do Unix, apesar de ter sido desenvolvido separadamente
- Originalmente, duas partes principais apenas:
 - O núcleo do sistema, ou kernel
 - Os programas do sistema: *shells*, editores, etc.

Sistemas Operacionais – Estruturas do S.O.

24

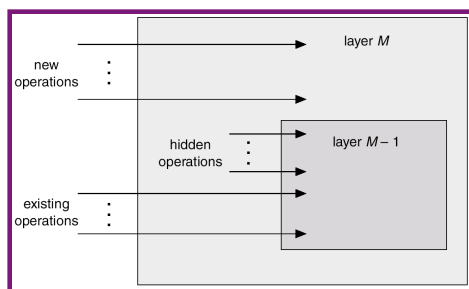
Estrutura do Unix



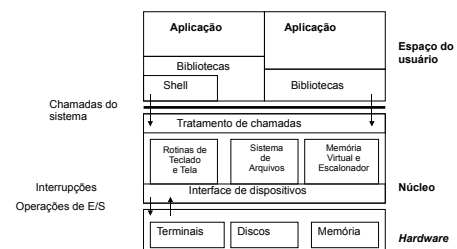
Enfoque em camadas

- O S.O. é dividido em níveis, cada um construído sobre os níveis inferiores
 - O nível 0 é o hardware
 - O nível mais alto é a interface com o usuário
- Com a modularidade, cada nível se torna mais fácil de construir, usando as funções dos níveis inferiores

Enfoque em camadas



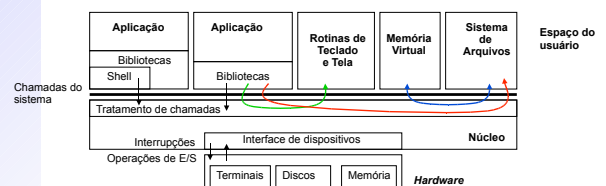
Estrutura em camadas do Linux



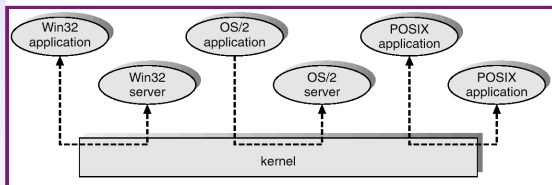
Estrutura em micro-kernel

- Sistemas com muita funcionalidade dentro do kernel podem ser tornar muito complexos
- Uma opção é manter o menor kernel possível e mover todo o resto para o espaço do usuário
- Tarefas são executadas por processos que trocam mensagens entre si
 - Menos código em modo protegido (kernel)
 - Mais fácil de estender (novos processos)
 - Mais fácil de portar para outras arquiteturas

Estrutura em micro-kernel do Mach



Estrutura inicial do Windows NT



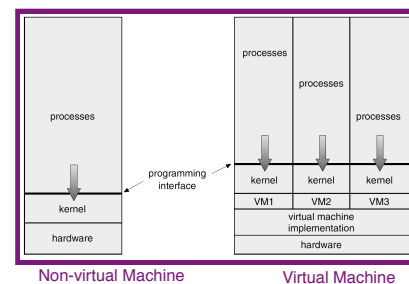
Máquinas virtuais

- Extensão do enfoque em camadas:
 - Um S.O. pode ser colocado sobre uma outra camada que o separa do HW
- Uma máquina virtual é uma abstração de SW e HW oferecida para um S.O. como se fosse a máquina pura onde ele espera executar
- O nível inferior (que cria as máquinas virtuais) oferece uma interface equivalente ao HW
- Diversos S.O. podem executar de forma protegida uns dos outros

Máquinas virtuais

- Os recursos do computador real são compartilhados entre as máquinas virtuais
 - Escalonamento da CPU pode dar a impressão que cada S.O. tem seu próprio processador
 - Recursos restritos podem ser virtualizados de forma que cada S.O. veja um recurso só seu

Máquinas virtuais



Máquinas virtuais

- Conceito originalmente utilizado em mainframes IBM (VM/CMS)
 - Cada usuário tinha a visão que seu terminal se comportava como um computador individual
 - Na verdade, terminais estavam ligados a um grande mainframe que executava as máq. virtuais
 - Impressoras e leitoras de cartões eram virtualmente conectadas a cada máquina virtual

Máquinas virtuais

- Máquinas virtuais são completamente isoladas do HW real e das demais máquinas
- A camada de virtualização adiciona overheads
- Não é possível compartilhar recursos de forma flexível – cada máq. tem sua quota do recurso
- Recurso muito útil para pesquisa e desenvolvimento de S.O.: não exige um HW isolado para desenvolvimento e testes
- Implementação é difícil pelo esforço necessário para isolar e duplicar o HW

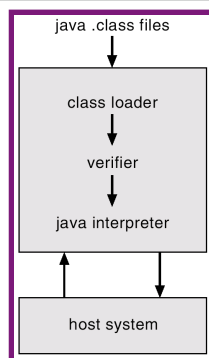
Máquinas virtuais

- Existem diversos tipos de virtualização dependendo de como a abstração é criada
 - Interpretadores: programas (SW) que processam cada instrução que seria entregue ao HW
 - Paravirtualização: o S.O. que executa na máquina virtual sabe da virtualização e colabora com o nível inferior de virtualização (hipervisor)
 - Panaevirtualização: o núcleo do S.O. é virtualizado (várias máquinas virtuais c/ um S.O.)
 - Virtualização com apoio do HW: processadores mais novos oferecem proteção para máqs. virtuais

Máquina virtual Java

- Máquina virtual do tipo interpretado
- Programas Java são compilados para instruções (*bytecode*) de uma máq. virtual java (JVM)
- Um interpretador que emula a JVM em cada HW precisa ser desenvolvido e inclui um carregador e um verificador de classes compiladas
- Compiladores *just-in-time* (JIT) aumentam desempenho
 - transformam *bytecode* em instruções da máquina real

Máquina virtual Java (JVM)



Observações/erros no capítulo

- O livro diz que o WinNT é um micro-kernel e essa é a interpretação da fig. 3.10
 - Apenas as primeiras versões do NT eram micro-kernel. Como ele tinha um desempenho muito ruim, foi tudo transferido para o kernel de novo.
 - Não existem "servidores" posix, os/2, win32. Hoje esses são módulos dentro do kernel.
- O sistema operacional Apple Mac OS X é um sistema em camadas baseado no kernel do Unix FreeBSD, não no Mach

Estamos saltando: 3.7 e 3.8
(projeto, implementação e geração)