

Aula 02 – Tipos de S.O. e Estrutura de um S.O.

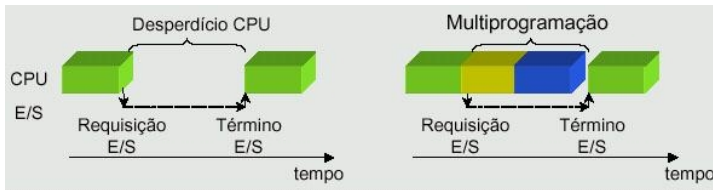
Norton Trevisan Roman
Clodoaldo Aparecido de Moraes Lima

3 de setembro de 2014

Apostila baseada nos trabalhos de Jó Ueyama, Kalinka Castelo Branco, Antônio Carlos Sementille, Carlos Maziero e nas transparências fornecidas no site de compra do livro “Sistemas Operacionais Modernos”

Multiprogramação

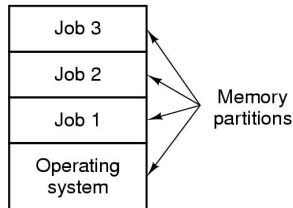
- A ideia é manter vários programas em memória ao mesmo tempo.
- Enquanto um programa aguarda E/S, outro pode ser executado



Multiprogramação

- Princípio histórico:

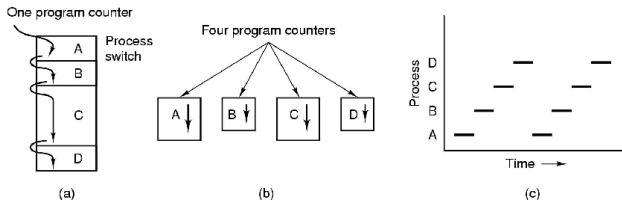
- Dividia-se a memória em diversas partes (partições) e aloca-se a cada uma dessas partes um serviço (programa, grosso modo).



- Mantinha-se na memória simultaneamente uma quantidade de serviços suficientes para ocupar 100% do tempo do processador, diminuindo a ociosidade.
- Importante: o hardware é que protegia cada um dos serviços contra acesso indevidos de outros serviços.

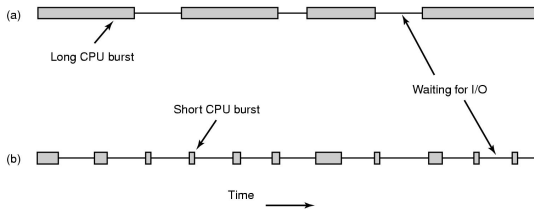
Multiprogramação

- Mecanismo de trocas rápidas de processos
 - Pseudoparalelismo: coleção de processos sendo executados alternadamente na CPU;
 - (a) Situação real no hardware
 - (b) Situação assumida em cada processo
 - (c) Embora, a cada instante, haja somente um rodando, em um intervalo longo, todos estão avançando



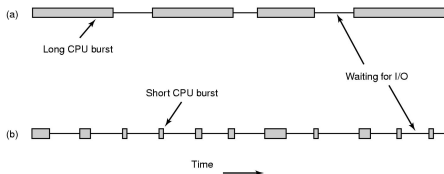
Multiprogramação

- Quando 2 ou mais processos estão simultaneamente no estado “pronto” (veremos mais adiante), eles competem pela CPU ao mesmo tempo
 - Cabe ao SO escolher qual executará – escalonamento
- Surtos de uso da CPU
 - Quase todos os processos alternam surtos de computação com requisições de E/S



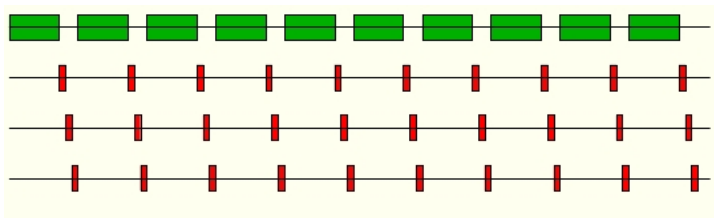
Multiprogramação

- a) Processos CPU-bound (orientados à CPU): processos que utilizam muito o processador
 - Tempo de execução é definido pelos ciclos de processador
- b) Processos I/O-bound (orientados à E/S): processos que realizam muito E/S
 - Tempo de execução é definido pela duração das operações de E/S



Multiprogramação

- Ideal: existir um balanceamento entre processos CPU-bound e I/O-bound

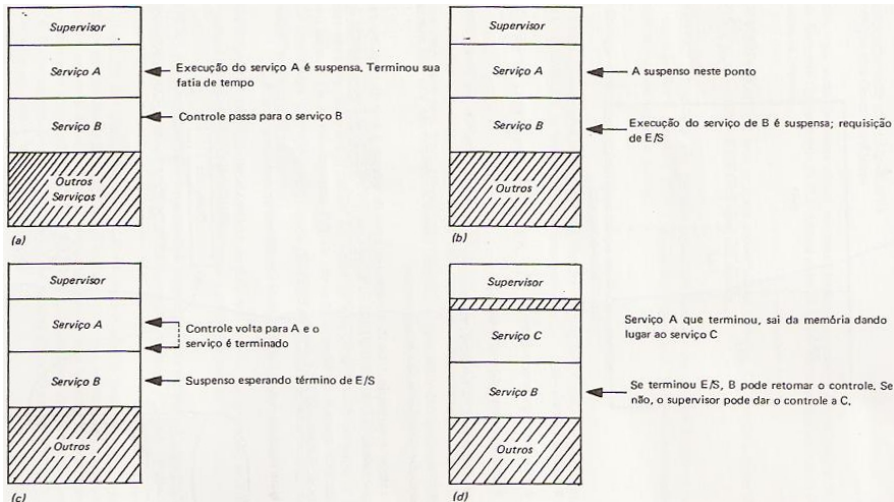


Processos de surto curto preenchem as lacunas de uso da CPU deixadas por processos mais longos.

Multiprogramação

- O S.O. Entrega o controle para o aplicativo de maior prioridade, que então é executado
- O aplicativo pode ser interrompido quando:
 - O serviço é terminado
 - O programa desocupa a memória
 - É detectado algum tipo de erro
 - O programa desocupa a memória
 - É requisitada uma operação de E/S
 - Suspenso até a operação de E/S terminar
 - Termina a fatia de tempo alocada a ele

Multiprogramação



Tipos de S.O.

- Classificação quanto à interação permitida
 - Sistemas em Batch (Lote)
 - Sistemas de Tempo Compartilhado (Interativo)
 - Sistemas de Tempo Real
 - fator determinante: tempo de resposta

Tipos de S.O. – Lote

- Histórico da operação (não batch):
 - Cada programa (job) escrito e perfurado em um cartão por um programador era entregue ao operador da máquina para que fosse processado
 - O processador ficava ocioso entre a execução de dois programas, ou à espera de dados de entrada ou pelo término de uma impressão
- Operação (batch):
 - Consistia de submeter ao computador um lote (batch) de programas de uma só vez.

Tipos de S.O. – Lote

- Os jobs dos usuários são submetidos em ordem sequencial para a execução
- Não existe interação entre o usuário e o job durante sua execução



Tipos de S.O. – Lote

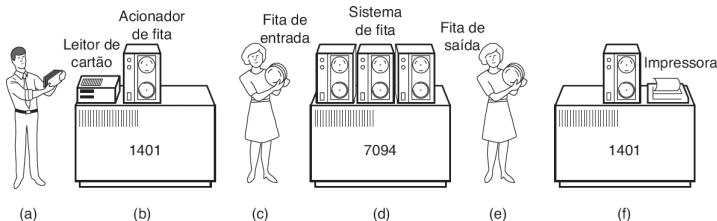
Procedimento (histórico)

- Armazenam-se os vários programas e seus dados em uma fita magnética
- Submete-se essa fita ao processamento
 - A CPU pode processar seqüencialmente cada job, diminuindo o tempo de execução dos jobs e o tempo de transição entre eles.
- Em vez do programa mandar sua saída diretamente à impressora, direciona-as a outra fita, que é submetida à impressora

Tipos de S.O. – Lote

Spooling

- Esta tipo de processamento é chamado **Spooling**
 - *System Peripheral Operations OffLine*
 - Foi a base dos sistemas batch antigos

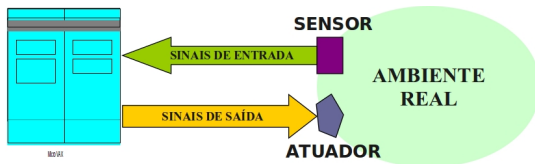


FMS (Fortran Monitor System) – Processamento: IBSYS – SO IMB para o 7094

- (a) Traz os cartões ao 1401 (c) Coloca a fita no 7094, que faz a computação (d)
- (b) Lê os cartões para uma fita (e) Coloca a fita no 1401, que imprime a saída (f)

Tipos de S.O. – de Tempo Real

- Servem aplicações que atendem processos externos, e que possuem tempos de resposta limitados
- Sinais de interrupções comandam a atenção do sistema
- Projetados para aplicações específicas (linhas de montagem, controle de caldeira)



Tipos de S.O. – de Tempo Real

- Sistema de tempo real crítico (Hard real time):
 - As ações *precisam* ocorrer em um determinado instante (ou intervalo)
 - O sistema deve garantir que uma determinada ação ocorrerá em um determinado momento
- Sistema de tempo real não crítico (Soft real time):
 - O descumprimento ocasional de algum prazo, embora indesejável, é aceitável
 - Tarefa crítica tem maior prioridade que as demais
 - Ex: sistemas de áudio ou multimídia

Tipos de S.O.

- Classificação segundo o Porte
 - S.O.s de Computadores de grande porte
 - S.O.s de Servidores
 - S.O.s de Multiprocessadores
 - S.O.s de Computadores Pessoais
 - S.O.s Embarcados
 - S.O.s de Cartões Inteligentes (smart cards)

Tipos de S.O.

- Visão do Usuário da Linguagem de Comando
 - As linguagens de comando são específicas de cada sistema

<i>Linux/Unix</i>	<i>Windows</i>
ls	dir
cp	copy
rm	del

Tipos de S.O.

- Visão do Usuário das Chamadas do Sistema
 - Permitem um controle mais eficiente sobre as operações do sistema e um acesso mais direto sobre as operações de hardware (especialmente a E/S).

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

Tipos de S.O.

- SOs são normalmente grandes e complexas coleções de rotinas de software \Rightarrow projetistas devem dar grande ênfase à sua organização interna e estrutura
 - Monolítica
 - Camadas
 - Micro-núcleo
 - Máquina virtual
 - Cliente-servidor

Tipos de S.O. – Estrutura Monolítica

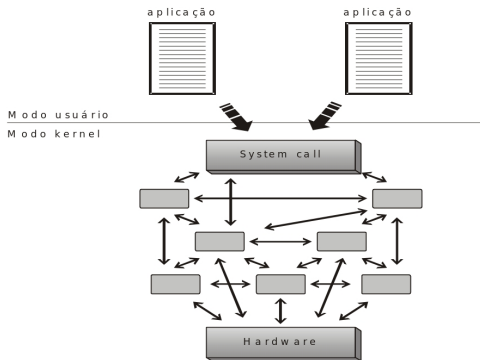
- Forma mais primitiva e comum de S.O.
- Escrito como uma coleção de procedimentos
 - Cada um pode chamar os demais se quiser
 - Todos os procedimentos são visíveis uns aos outros
 - Os procedimentos executam sobre o hardware, como se fossem um único programa.
 - O S.O. inteiro trabalha no espaço de núcleo



Tipos de S.O. – Estrutura Monolítica

- Ex:

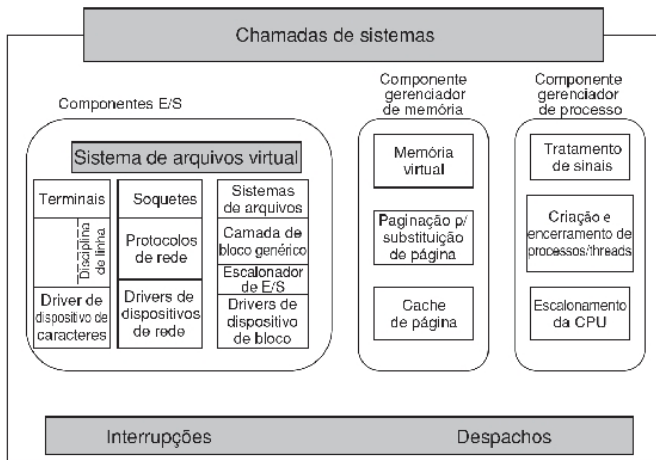
- MS-DOS
- Windows
- Unix e Linux
(embora sejam compostos de módulos, em que há informação oculta do resto do sistema)



Tipos de S.O. – Estrutura Monolítica

- Construção de SO Monolítico
 - Compila-se todos os procedimentos com o objetivo de criar os módulos-objeto.
 - Juntam-se todos os módulos-objeto usando o linker, criando-se um único programa executável chamado kernel.
- Prós
 - Desempenho
- Contras
 - Difícil manutenção
 - Também considerada como uma “grande bagunça”

Tipos de S.O. – Estrutura Monolítica



Estrutura do núcleo do Linux

Tipos de S.O. – Estrutura Monolítica

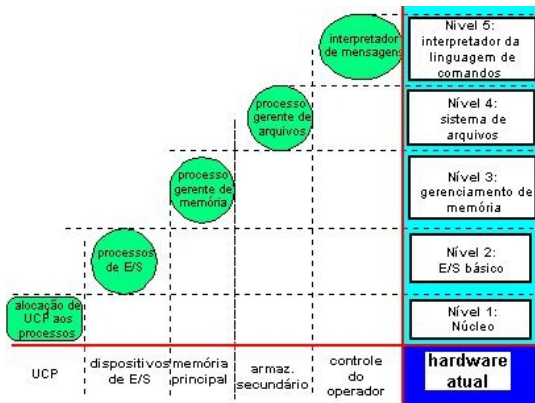
- Sistemas de Camadas – Estrutura Hierárquica de Níveis de Abstração
 - A idéia básica é criar um S.O. como uma hierarquia de níveis de abstração
 - Cada camada construída sobre a imediatamente inferior
 - A cada nível, os detalhes de operação dos níveis inferiores podem ser ignorados.
 - Cada nível pode confiar nos objetos e operações fornecidas pelos níveis inferiores.
 - Ex:
 - MULTICS, OpenVMS, Windows NT

Tipos de S.O. – Estrutura Monolítica

- Sistemas de Camadas – Princípios
 - Modularização:
 - Divisão de um programa complexo em módulos hierárquicos de menor complexidade.
 - Os módulos interagem através de interfaces bem definidas.
 - Conceito de “Informação Escondida”:
 - Os detalhes das estruturas de dados e algoritmos são confinados em módulos.
 - Externamente, um módulo é conhecido por executar uma função específica sobre objetos de determinado tipo.

Tipos de S.O. – Estrutura Monolítica

- Sistemas de Camadas

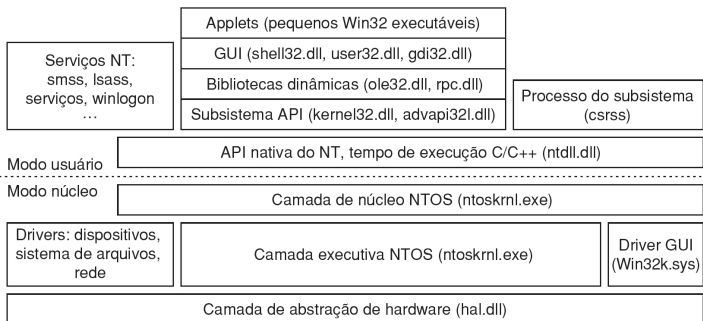


Tipos de S.O. – Estrutura Monolítica

- Sistemas de Camadas
 - Prós:
 - O isolamento dos serviços do sistema operacional facilita a manutenção.
 - A hierarquia protege as camadas mais internas.
 - Contras:
 - Diminuição do desempenho por conta da quantidade de mudança de modos de acesso.

Tipos de S.O. – Estrutura Monolítica

- Sistemas de Camadas (aproximadamente)



Estrutura do núcleo do NT

Tipos de S.O. – Estrutura de MicroKernel

- Busca tornar o núcleo do S.O. o menor possível;
- Os serviços são disponibilizados em processos;
- Cada processo é responsável por gerenciar um conjunto específico de funções (ex: gerência de memória, gerência de arquivos, gerência de processos etc.);
 - Dois tipos: processo cliente e processo servidor;
- A principal função do núcleo é gerenciar a comunicação entre esses processos.

Tipos de S.O. – Estrutura de MicroKernel

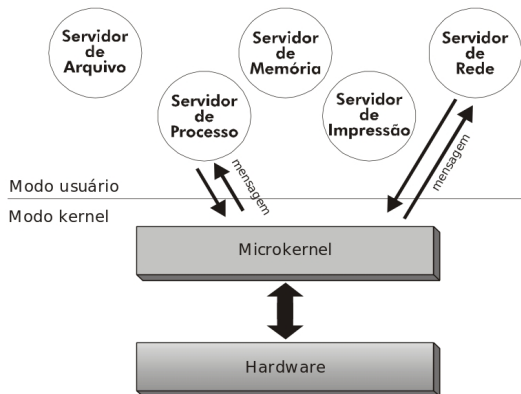
- O MicroNúcleo (microkernel) incorpora somente as funções de baixo nível mais vitais
 - Fornece uma base sobre a qual é contruído o resto do S.O.
- A maioria destes sistemas são construídos como coleções de processos concorrentes
- Núcleo fornece serviços de alocação de UCP e de comunicação aos processos (IPC).
 - A comunicação entre processos (IPC – Inter-Process Communication) é o grupo de mecanismos que permite aos processos transferirem informação entre si.

Tipos de S.O. – Estrutura de MicroKernel

- Ex:

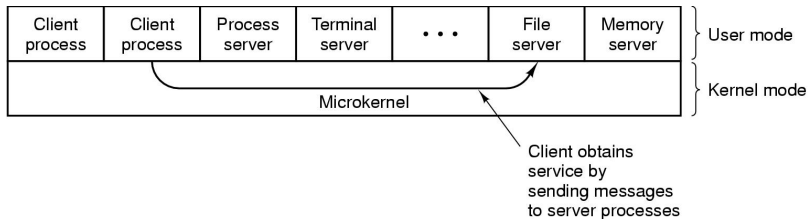
- Minix

- Symbian †



Tipos de S.O.

- Estrutura de microkernel



Tipos de S.O.

- Estrutura de MicroKernel

- Prós:

- Maior proteção do núcleo: todos os processos são executados em modo usuário
 - Alta disponibilidade: se um servidor falhar, o sistema não ficará altamente comprometido;
 - Maior eficiência: a comunicação entre serviços poderá ser realizada entre vários processadores ou até mesmo várias máquinas distribuídas;
 - Melhor confiabilidade e escalabilidade;

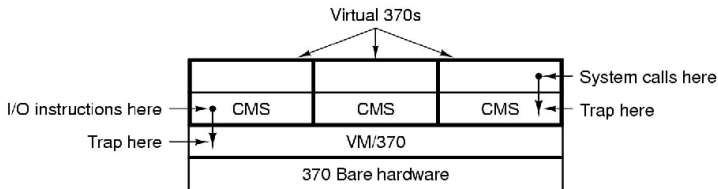
- Contras:

- Grande complexidade para sua implementação
 - Menor desempenho devido à necessidade de mudança de modo de acesso (modo usuário e núcleo)

Tipos de S.O.

- Máquina virtual

- O Modelo de Máquina Virtual ou Virtual Machine (VM), cria um nível intermediário entre o hardware e o S.O., denominado Gerência de Máquinas Virtuais (GMV).
 - Ex: IBM VM/370

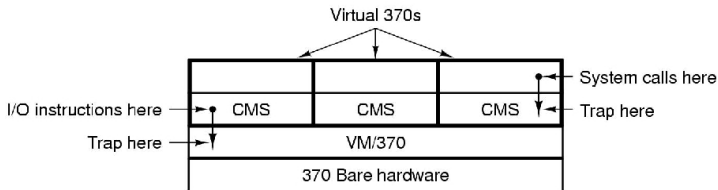


Estrutura do VM/370 (IBM) com o S.O. CMS

Tipos de S.O.

- Máquina virtual

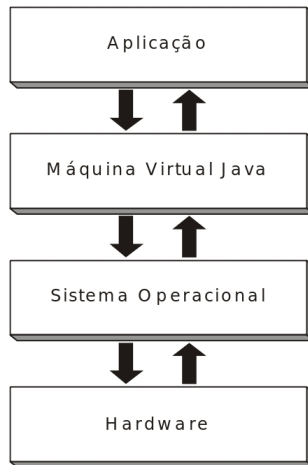
- Este nível (GMV) cria diversas máquinas virtuais independentes, onde cada uma oferece uma cópia virtual do hardware, incluindo modos de acesso, interrupções, dispositivos de E/S, etc.



Estrutura do VM/370 (IBM) com o S.O. CMS

Tipos de S.O.

- Máquina virtual
 - Um outro exemplo de utilização desta estrutura ocorre na linguagem Java. Para executar um programa Java é necessário uma máquina virtual Java (Java Virtual Machine – JVM)



Tipos de S.O.

- Máquina Virtual

- Prós:

- Cria isolamento total de cada VM

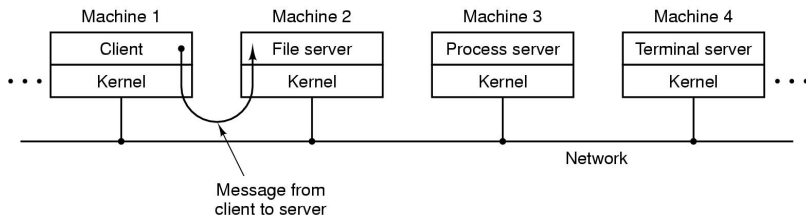
- Contras:

- Grande complexidade, devido à necessidade de compartilhar e gerenciar os recursos de hardware entre as diversas VMs

Tipos de S.O.

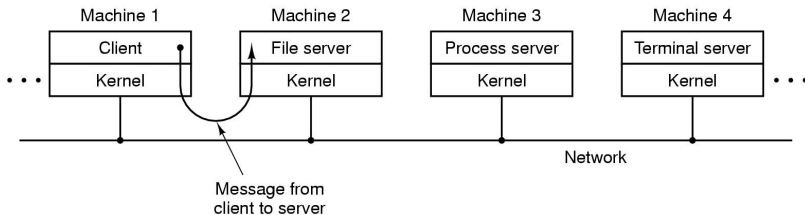
- Modelo Cliente-Servidor

- Implementa-se o máximo do SO como processos do Usuário (processos cliente)
- Serviços (ex: ler um arquivo) são feitos por requisição do cliente ao processo servidor, que executa e devolve a resposta



Tipos de S.O.

- Modelo Cliente-Servidor
 - O kernel trata apenas da comunicação entre clientes e servidores
 - Há servidor de arquivos, memória, processos etc
 - Variação da ideia de micro-núcleo



Tipos de S.O.

- Modelo Cliente-Servidor

- Prós:

- Kernel mínimo
 - Adaptabilidade ao uso em sistemas distribuídos
 - Cada servidor roda em modo Usuário – não tem acesso direto ao hardware
 - Erros em um deles não afetarão a máquina toda

- Contras:

- Lentidão
 - Complexidade pela necessidade de se carregar comandos em registradores de controladoras etc.

Referências Adicionais

- Discussão entre Andrew Tanenbaum e Linus Torvalds sobre modelo de micronúcleo × modelo monolítico
 - <http://oreilly.com/catalog/opensources/book/appa.html>