

# Resumão de Sistemas Operacionais

---

*Autor: Adonai Estrela Medrado (adonaimedrado@hotmail.com)*

*Revisado em 26 de junho de 2010*

Construído tendo como referência SIMÕES, Ricardo de Magalhães. **Módulo de Sistemas Operacionais**. Vitória: ESAB, 2008. Disponível apenas para alunos.

Este material é apenas um resumo disponibilizado para uso acadêmico e didático. Ele não pretende esgotar o assunto ou abordá-lo em sua integralidade. A sua utilidade é relembrar alguns conceitos e estimular o aprofundamento e a pesquisa.

## Fundamentos

O sistema operacional é um *software* especial que controla recursos compartilhados. Os programas que trabalham sob o sistema operacional utilizam das Chamadas do Sistema (*System Calls*) para ter acesso a estes recursos.

As principais funções de um sistema operacional são 1) *tratamento de interrupção* (evento inesperado externo ao programa em execução e independente de instrução sendo executada; sempre são geradas por dispositivos; paralisa momentaneamente a execução do programa atual); 2) *tratamento de exceção* (originada por algum programa em execução; o programa que executou a operação inválida é avisado, caso não saiba tratá-la o aviso será encaminhado ao usuário; paralisa momentaneamente a execução do programa atual); 3) *gerenciamento de programas* (colocar em execução, administrar e finalizar programas); 4) *gerenciamento de memória RAM* (incluindo verificar disponibilidade); 5) *gerenciamento de arquivos* (controlar a criação, edição e exclusão de arquivos); 6) *gerenciamento de dispositivos* (verificação de utilização, disponibilidade, liberação, etc.); e 7) *suporte a redes* (gerenciamento e acesso às redes de computadores). Atualmente passou a ser requisito também o *sistema de segurança* (somente programas permitidos executam; somente usuários permitidos acessam os recursos e informações).

## História

Em 1950 os computadores começaram a utilizar os transistores e o Univac 1103 surge como primeiro computador a utilizar um sistema operacional (SO) o ERA. Em 1976

Steve Jobs e Stephan Wozniak inventam o primeiro computador pessoal, o Apple I. Em 1981 a IBM lança o PC (*Personal Computer*) com o processador Intel i8080.

Também em 1981 foi desenvolvida a primeira interface gráfica pela Xerox, a partir daí vários sistemas gráficos foram lançados ainda na década de 80. O Apple Macintosh (1984) foi o primeiro computador popular com interface gráfica.

O Windows começou seu desenvolvimento em 1983 como apenas um ambiente gráfico de operação, funcionando sob o MS-DOS. O Windows 1.0 foi lançado em 1985, em 1987 o Windows 2.0. Estas duas versões não foram muito difundidas, talvez devido às limitações de *hardware* da época. Em 1990 surge a versão 3.0 que se torna padrão nas máquinas IBM-PC e compatíveis. Ainda nos anos 90 foram lançadas duas frentes de trabalho, uma para servidor e computação profissional (Windows NT Server/Workstation) e outra para usuário doméstico (Windows 95). A interface provida pelo Windows 95 era totalmente nova tornando mais fácil a utilização dos usuários iniciantes. O sistema de arquivos FAT32 veio na segunda versão do Windows 95 (em 1996), o NTFS com o Windows NT. Outro lançamento importante foi o Windows 2000 (em 2000) com melhoria no gerenciamento de processos, memória e dispositivo. O Windows XP lançado em 2001 herda o *kernel* do Windows 2000. O Windows 2003 e 2008 seguem a linha de servidores, e o Windows Vista e 7 segue a linha sistemas operacionais para uso pessoal com melhora na interface e nas funcionalidades.

O Linux começou seu desenvolvimento em 1991 e recebe apoio de várias empresas como a IBM, Oracle e Red Hat. Em 1996 foi lançado o Linux Kernel 2 base das distribuições atuais. Uma distribuição é um "*kit*" que contém o SO e um conjunto de ferramentas (ex.: OpenSuse, Debian, Ubuntu, etc). O Linux é utilizado em ambientes acadêmicos e científicos, mas começa a ser utilizado em ambientes empresariais e domésticos. Ele adota padrões estabelecidos no mercado, dentre eles o POSIX, e é bastante utilizado nos servidores para Internet.

## **Classificação**

Os SOs podem ser classificados de acordo com 1) *características básicas de arquitetura do sistema* (Monolítico, *Microkernel* ou Híbrido); 2) *capacidade de executar tarefas* (Monotarefa ou Multitarefa); e 3) *quantidade de usuários que podem operar o sistema* (Monousuário ou Multiusuário).

Um sistema monolítico é construído como uma coleção de rotinas onde cada uma pode chamar qualquer outra. Um sistema *microkernel* possui separação de funções chamadas Servidores. Um sistema híbrido executa os servidores externos em um modo protegido.

Um sistema monotarefa executa uma tarefa de cada vez. Um sistema multitarefa executa várias tarefas simultaneamente e subdivide-se em 1) *sistemas de tempo compartilhado* (*time sharing systems* - divide o tempo do processador em pequenos intervalos - *time slice*; quem define o tempo de execução de cada programa é o sistema operacional); 2) *sistemas de tempo real* (tempo de execução é definido pelo próprio programa; o programa fica em execução durante o tempo que achar necessário ou até que outro com maior prioridade chegue à fila; o conceito de *time slice* não se aplica). Troca de contexto é uma operação que ocorre nos sistemas multitarefa quando se altera o programa que está sendo executado.

Um sistema monousuário não faz distinção entre os usuários, foi projetado para ser utilizado por um único usuário. Um sistema multiusuário permite diversos usuários garantindo a separação e não interferência entre as tarefas de cada um.

## Processos

O processo é uma instância de um programa em execução.

No controle e execução do processo pelo SO as seguintes informações necessitam ser armazenadas: 1) código executável; 2) espaço de memória utilizado (com distinção entre código, dados e estruturas de controle); 3) descritores do SO (tabela de arquivos, etc.); 4) atributos de segurança; e 5) estado do processo (executando, em espera, aguardando recurso, etc.).

Há três tipos básicos de gerenciadores de processo 1) *gerenciador por admissão* (gerenciador escolhe qual processo irá executar e quando a depender da sua prioridade; geralmente utilizado em sistemas de tempo real); 2) *gerenciador por tempo médio*; e 3) *gerenciador despachantes* (define os processos a serem executados de acordo com eventos ocorridos no sistema).

Os Gerenciadores de Processo possuem algoritmos para garantir a troca de processos de maneira correta e organizada. São exemplos de estratégias utilizadas 1) *FIFO* (*First In First Out*) onde um programa abre um canal unidirecional com um segundo e

escreve dados que chegarão ao destino na mesma ordem em que foram escritos; 2) *PIPES unidirecionais* (similar ao FIFO, porém com o controle feito pelo SO e não pelo programa); 3) *fila de mensagem - message queue* (baseia-se em área administrada pelo SO para troca de mensagem entre processos); e 4) *memória compartilhada* (semelhante à fila de mensagem, mas o controle é feito dentro do programa e não pelo sistema operacional).

## **Threads**

*Threads* são linhas de execução independentes de um processo. Uma *thread* compartilha os recursos do processo que a criou. Processadores que conseguem lidar simultaneamente com mais de uma *threads* são caracterizados *Multi-Threading* ou *Hiper-Threading*.

Os recursos utilizados para fazer a troca de contexto entre *threads* são muito menores do que aqueles necessários para fazer a troca de contexto entre processos.

Existem duas formas de troca de contexto das *threads* 1) preemptiva (controle do sistema operacional; *thread* pode ser interrompida em momento crítico) e 2) cooperativa (controle da *thread*; uma *thread* pode tomar o controle do processador).

A realização de operações simultâneas pelas *threads* são chamadas de Condição de Corrida ou Condição de Disputa e exigem mecanismos de comunicação e sincronização que podem ser criados pelo sistema operacional.

## **Comunicação entre Processos e Threads**

Os sistemas operacionais possuem em sua API um conjunto de funções necessárias para a comunicação entre processos e *threads*. No Windows as COM (*Component Object Model*) e recentemente o *Windows Communication Foundation*, no UNIX POSIX Core Services (para processos) e a POSIX-Thread ou PThread (para *threads*) e no MacOS X o *Core Foundation*.

## **Sincronização**

A sincronização dos recursos do computador é realizada pelo SO multitarefas e é feita tanto para dados (para manter dados em integridade) quanto para processos (evitar conflito na utilização dos recursos).

Os mecanismos para controle de sincronização, dentre outros, podem ser 1) *barreiras* (criadas e gerenciadas pelo programa; programa entra em estado de espera até

que todos os processos pertencentes ao mesmo programa também entrem nesta estado); 2) semáforos (podem ser implementados tanto pelo SO quanto pelo programa; consistem em variáveis de controle que indicam quantos processos podem compartilhar um recurso); 3) trava simples (impede a utilização do recurso por outro processo), trava especial (sinaliza quando se tenta utilizar um recurso já em uso) e trava compartilhada (um único processo recebe permissão de leitura e escrita e os demais somente de leitura). A barreira e o semáforo permitem compartilhamento de recurso, a trava simples não. A sincronia exige a existência de uma instrução capaz de ao mesmo tempo "verificar e, se possível, travar".

### **Deadlock (Bloqueio)**

Um bloqueio acontece quando um processo aguarda um recurso que, caso o sistema operacional não intervenha, nunca será liberado. É condição para existência do *Deadlock* que apenas o processo que utiliza um recurso seja capaz de liberá-lo.

São exemplos de estratégias para prevenir *Deadlock* 1) uso de ordenação por precedência dos recursos (os recursos devem ser alocados seguindo um ordem); 2) determinação de tempo de alocação; 3) determinação que nenhum processo terá acesso exclusivo aos recursos; e 4) pré-alocação de todos os recursos no início da execução do programa.

O tratamento do *Deadlock* pode ser feito com a finalização dos processos envolvidos e sua subsequente reinicialização sequencial.

### **Gerenciamento de Memória**

O gerenciamento de memória tem por objetivo 1) oferecer uma área de armazenamento para os processos; 2) proteger os processos contra falhas de terceiros; e 3) prover um desempenho satisfatório aos usuários. Deseja-se também compartilhamento de memória entre processos e acesso transparente à memória.

A memória é caracterizada como compartilhada se mais de um programa pode acessá-la simultaneamente. A memória compartilhada pode ser criada eletricamente (*hardware* destinado a este fim; utilizada em supercomputadores) ou logicamente (criada através de estruturas de dados especiais).

O coletor de lixo é um papel incluso em alguns gerenciadores de memória. Sua tarefa é identificar dados não mais utilizados na memória e liberar o espaço para uso.

## Memória Virtual

O mecanismo de memória virtual torna o uso da memória principal mais eficiente. Ele cria um ambiente de memória contínua a partir de um ambiente de memória fragmentada, podendo utilizar dispositivos de armazenamento para salvamento temporário. Sua implementação depende do modo de endereçamento de memória utilizado pelo processador. Nos processadores da família Intel este recurso já está presente desde o 80286 lançado em 1982. A Microsoft utiliza este modelo a partir do Windows 3.

Neste mecanismo chama-se página uma área contínua de dados. O tamanho da página varia de acordo com a quantidade de memória do computador, mas geralmente é de pelo menos 2 kilobytes. A tabela de paginação pode ser única para o sistema ou específica para cada aplicação e mapeia o endereço de memória virtual aos endereços físicos.

Chama-se paginação a técnica utilizada para transferência de dados inativos da memória principal para a secundária (*Page-Out*) e transferência de dados ativos de volta para a memória principal (*Page-In*). Na recolocação de uma página na memória principal é necessário fazer a atualização do mapeamento. Chama-se *Page Fault* (Falha de Página) a detecção pelo gerenciador de memória que uma informação requisitada não está na memória principal. Uma *Page Swap* (Troca de Página) ocorre quando, por falta de espaço livre na memória principal, uma página tem que ser transferida para a memória secundária para que outra seja colocada na memória principal; geralmente a página candidata para *Swap* é a menos utilizada.

Algumas páginas não podem ser removidas da memória principal a exemplo daquelas que armazenam 1) a tabela de paginação; 2) rotinas de tratamento de interrupção; e 3) informações manipuladas pelos dispositivos com recurso de acesso direto à memória (DMA).

## Segmentação da Memória

O mecanismo de segmentação da memória particiona a memória em segmentos de dados. Cada segmento possui uma localização física, uma permissão de acesso, um tamanho e um identificador – dependendo da implementação pode existir ainda um indicador de se a página está na memória principal ou na memória secundária. O processo só acessa o segmento que tiver permissão, caso tente fazer um acesso não permitido ele será negado e será gerado um aviso de erro. Este mecanismo precisa ser viabilizado por

*hardware* via Unidade de Gerenciamento de Memória (*Memory Management Unit, MMU*) que faz a mediação de todo o processo sendo que os processos não acessam diretamente a área de memória do computador. Solicitações que não sejam permitidas não serão atendidas pela MMU e gerarão um erro que será passado ao processo. Um processo pode permitir que outros processos além dele acessem a um segmento.

É possível uma combinação entre os mecanismos de paginação e segmentação (cada segmento pode ser dividido em páginas).

### ***Driver***

Um *driver* de dispositivo é um conjunto de instruções específicas para tratamento de um dispositivo que depende diretamente do sistema operacional. Funcionam em duas camadas: comandos lógicos (acessíveis a todos os programas) e comandos físicos (utilizados pelo *driver* para executar os comandos lógicos solicitados no dispositivo). Podem existir *drivers* genéricos, mas geralmente não são capazes de utilizar todos os recursos fornecidos por um dispositivo.

O *firmware* é um pequeno programa que se encontra nos circuitos eletrônicos de um dispositivo. Alguns *firmwares* podem funcionar como *drivers*, mas necessitam antes serem carregados na memória.

### **Sistema de Arquivos**

Um sistema de arquivos define a maneira como os dados serão gravados e gerenciados nos dispositivos de armazenamento acessados pelo computador. Os sistemas de arquivos trabalham com blocos de dados. Cada bloco de dado contém uma parte do dado que forma o arquivo.

Geralmente um arquivo tem propriedades associadas como modo de acesso (leitura, escrita, execução), data da criação, data de alteração, nome e extensão. O conjunto de propriedades disponíveis depende do sistema de arquivos.

Os diretórios são utilizados pelo usuário para organizar os arquivos. A tabela de arquivos é utilizada pelo Sistema Gerenciador de Arquivos para guardar a organização física dos arquivos.

Existem dois tipos de sistemas de arquivos 1) sistemas de disco (funciona em uma unidade de armazenamento) e 2) sistema de arquivo em rede (armazenamento é feito em outro computador da rede).

O NTFS é um sistema de arquivos de disco que utiliza estrutura de Árvore B+ e que foi desenhada para minimizar a fragmentação. A Tabela de Arquivos do NTFS chama-se Tabela Mestre de Arquivos. Todas as informações relativas ao arquivo, inclusive seu conteúdo são tratados como metadados. Os meta-arquivos auxiliam na organização dos metadados. Arquivos pequenos (<800 bytes) não criptografados ou comprimidos são armazenados na própria Tabela Mestre e são denominados *arquivos residentes*.

O NTFS permite 1) nome alternativo; 2) cota de armazenamento; 3) pontos de montagem; 4) ponto de junção; 5) ligação entre arquivos; 5) gerenciamento hierárquico do armazenamento; 6) controle de versão dos arquivos (via *shadow copy*); 7) compressão de arquivos; 8) armazenagem unificada de arquivos (múltiplas cópias de um mesmo arquivo são armazenadas uma única vez com referências múltiplas na tabela mestre); e 9) criptografia.

Outro sistema de arquivos de disco é o *Linux Extented File System* que na sua versão 2 (conhecida como ext2 e desenvolvida em 1993) pode gerenciar tamanho máximo de 4 TB. Em 2001 foi desenvolvido o ext3 com *Journaling* (otimiza operações de busca e proteção contra falhas no sistema). Em 2006 surge o ext4 aumentando o tamanho máximo do disco para 1 EB. Os ext2, 3 e 4 separam explicitamente as propriedades do arquivo (*inode*) do dado. O *Journaling* tem uma estrutura semelhante a uma Árvore-B+. O ext3 possui as seguintes características 1) definição padrão dos atributos de um arquivo; 2) definição padrão do grupo de usuário a um diretório; 3) definição do tamanho da unidade básica de informação na criação do arquivo; 4) checagem da integridade do sistema de arquivos em momentos definidos.

## **Sistema Operacional de Rede**

Um sistema operacional de redes de computadores faz o gerenciamento da rede de comunicação e dos recursos disponíveis na rede (ex.: Windows 2008 Server). Em termos gerais ele deve prover 1) suporte à protocolos de comunicação em rede; 2) mecanismos de segurança; 3) gerenciamento de arquivos distribuídos; 4) gerenciamento de dispositivos distribuídos; 5) gerenciamento de usuários.



Há sistemas operacionais de rede especializados que funcionam em dispositivos específicos como, por exemplo, o Cisco IOS que funciona em roteadores.

### **Gerenciamento de Usuário**

Para efetuar o gerenciamento de usuários deve ser possível definir: 1) mecanismos de proteção das informações, 2) quem e quando poderá utilizar o computador; e 3) que recursos poderão ser utilizados por cada usuário.

Autenticação é o processo de confirmar a identidade do usuário e sua permissão para realizar uma tarefa. Permissão é um conjunto de regras que define como o usuário pode utilizar os recursos do computador.

Facilita-se o gerenciamento de usuário a partir da criação de grupos de usuário, definindo-se permissões para os grupos.

### **Active Directory**

O *Active Directory* é baseado no LDAP<sup>1</sup> e tem por objetivo fornecer uma forma centralizada de autenticação e autorização de serviços. Ele permite definir políticas de instalação e atualização de programas. Sua estrutura mais alta é a Floresta. Cada Floresta possui uma ou mais Árvores. Cada Árvore possui um ou mais Domínios. Os Domínios podem agrupar seus objetos em Unidades Organizacionais facilitando o gerenciamento. Os usuários são registrados no Domínio. As informações do *Active Directory* são armazenadas em um ou mais Controladores de Domínio, cada um com sua cópia do diretório que deverá ser sincronizada em caso de alteração. Para que um usuário de um Domínio tenha acesso aos recursos de outro Domínio deve-se utilizar um Relacionamento de Confiança. Por padrão um Domínio já tem um relacionamento de confiança com todos os Domínios da sua Árvore. Os Relacionamento de Confiança podem ser 1) de direção única (de A para B, mas não de B para A); 2) de direção dupla. Além disto, as seguintes denominações são utilizadas 1) domínio confiável (tem usuários confiáveis); 2) domínio confiante (permite acesso de usuários confiáveis); 3) relacionamento transitivo (estendível a outros Domínios Confiáveis da Árvore); 4) relacionamento explícito (relacionamento criado pelo

---

<sup>1</sup> Um LDAP (*Lightweight Directory Access Protocol*) é composto por objetos abstratos que representam recursos (computadores, impressoras, arquivos, etc.) e usuários de uma rede de computadores. Dentre outras operações ao se conectar a um servidor LDAP é possível 1) fazer buscas no diretório; 2) adicionar, modificar e apagar objetos; 3) requerer canal de comunicação seguro.

administrador); 5) relacionamento cruzado (entre Domínios de Árvores diferentes); 6) Floresta Confiável (relacionamento feito diretamente na raiz, propaga-se entre todos os objetos do diretório).

## Segurança

A segurança depende dos mecanismos de segurança oferecidos pelos sistemas e da configuração. Na década de 1970 foi especificado um sistema de segurança por encomenda da Agencia de Segurança dos EUA. Ele foi denominado de Modelo Bell-LaPadula (MBL) e influencia até hoje os desenhos dos sistemas, porém não foi construído como proposto, pois seus vários mecanismos de segurança impunham sérias restrições de desempenho.

Algumas estratégias dos sistemas de segurança envolvem: 1) executar apenas programas testados; e 2) executar qualquer programa e verificar as operações realizadas. Um mecanismo útil é colocar em níveis diferentes o código do sistema operacional e o código do usuário. Proteções de *hardware* também são utilizadas. Outros mecanismos e técnicas incluem 1) firewall; 2) criptografia; 3) cópia de segurança dos dados; e 4) programa antivírus.

A manutenção da segurança deve ser garantida pelo uso e atualização constante dos mecanismos de segurança por parte do usuário/administrador.

## Sistemas Embutidos

Os sistemas embutidos por se destinarem a uma função específica e dedicada são sistemas reduzidos que incluem o dispositivo eletrônico a ser utilizado. Geralmente têm tamanho menor (se comparados aos sistemas não embutidos) e são otimizados para a função que irá desempenhar. Exemplo de sistemas operacionais para sistemas embutidos são LiMO, Windows Mobile e Symbian.

## Máquina Virtual

Uma máquina virtual é uma máquina real construída logicamente. Podem ser divididas em duas categorias: Máquinas Virtuais Completas (simula toda a estrutura de um computador real; permite a instalação de um SO; ex.: aquela criada pelo Virtual PC) e Máquinas Virtuais Binárias (para execução de programa específico; ex.: aquela criada pela Java Virtual Machine). A máquina que está servindo de base para a máquina virtual é chamada de Sistema Hospedeiro (*Host*), a máquina virtual é o Sistema Convidado (*Guest*).

Em qualquer dos casos, o sistema operacional hospedeiro é responsável por gerenciar apenas os recursos do processo que gera o ambiente virtual.

### **Sistema Operacional para Internet**

Um sistema operacional para Internet fornece via Internet o mesmo conjunto de serviços de um sistema operacional convencional. Algumas tentativas de desenvolver este conceito foram: WebOS (hoje Hyperoffice), YouOS, Desktop Two, G.ho.st (hoje utilizando o conceito de nuvem) e EyeOS.

### **Computador de Alto Desempenho**

Um computador de alto desempenho (supercomputador) é aquele capaz de resolver um problema que não pode ser resolvido em tempo satisfatório por um computador de pequeno porte. Os supercomputadores podem ser divididos em dois tipos 1) sistemas com múltiplos processadores e 2) sistemas com múltiplos computadores (*Cluster Grid*). A otimização e desenho de um sistema operacional para um computador de alto desempenho depende da sua finalidade e da sua arquitetura.