



Sistemas Operacionais

João Batista Pinto Neto



**Cuiabá-MT
2014**



Presidência da República Federativa do Brasil
Ministério da Educação
Secretaria de Educação Profissional e Tecnológica
Diretoria de Integração das Redes de Educação Profissional e Tecnológica

© Este caderno foi elaborado pelo Instituto Federal de Educação, Ciência e Tecnologia de Rondônia-RO, para a Rede e-Tec Brasil, do Ministério da Educação em parceria com a Universidade Federal de Mato Grosso.

Equipe de Revisão

Universidade Federal de Mato Grosso – UFMT

Coordenação Institucional
Carlos Rinaldi

Coordenação de Produção de Material Didático Impresso
Pedro Roberto Piloni

Ilustração
Tatiane Hirata

Diagramação
Tatiane Hirata

Revisão de Língua Portuguesa
Dinaura Batista de Pádua

Revisão Final
Claudinet Antonio Coltri Junior

Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO

Campus Porto Velho Zona Norte

Direção-Geral
Miguel Fabrício Zamberlan

Direção de Administração e Planejamento
Gilberto Laske

Departamento de Produção de EaD
Ariadne Joseane Felix Quintela

Coordenação de Design Visual e Ambientes de Aprendizagem
Rafael Nink de Carvalho

Coordenação da Rede e-Tec
Ruth Aparecida Viana da Silva

Projeto Gráfico

Rede e-Tec Brasil / UFMT

Sistemas Operacionais - Informática para Internet

P6596s Pinto Neto, João Batista.

Sistemas operacionais / João Batista Pinto Neto; org. Instituto Federal de Educação, Ciência e Tecnologia; Universidade Federal do Mato Grosso - Cuiabá : UFMT; Porto Velho: IFRO, 2014.

90 p. ; -- cm.
Curso Informática para internet.

ISBN 978-85-68172-17-9

1. Informática – Redes de computadores - Gerência. 2. Sistema Operacional – Conceitos e funções . I. Instituto Federal de Educação, Ciência e Tecnologia. II. Universidade Federal do Mato Grosso. III. Título.

CDD 005.43
CDU 004.78

Ficha Catalográfica Elaborada pela Bibliotecária Cleuza Diogo Antunes CRB 11/864

Apresentação Rede e-Tec Brasil

Prezado(a) estudante,

Bem-vindo(a) à Rede e-Tec Brasil!

Você faz parte de uma rede nacional de ensino que, por sua vez, constitui uma das ações do Pronatec - Programa Nacional de Acesso ao Ensino Técnico e Emprego. O Pronatec, instituído pela Lei nº 12.513/2011, tem como objetivo principal expandir, interiorizar e democratizar a oferta de cursos de Educação Profissional e Tecnológica (EPT) para a população brasileira propiciando caminho de acesso mais rápido ao emprego.

É neste âmbito que as ações da Rede e-Tec Brasil promovem a parceria entre a Secretaria de Educação Profissional e Tecnológica (Setec) e as instâncias promotoras de ensino técnico, como os institutos federais, as secretarias de educação dos estados, as universidades, as escolas e colégios tecnológicos e o Sistema S.

A educação a distância no nosso país, de dimensões continentais e grande diversidade regional e cultural, longe de distanciar, aproxima as pessoas ao garantir acesso à educação de qualidade e ao promover o fortalecimento da formação de jovens moradores de regiões distantes, geográfica ou economicamente, dos grandes centros.

A Rede e-Tec Brasil leva diversos cursos técnicos a todas as regiões do país, incentivando os estudantes a concluir o ensino médio e a realizar uma formação e atualização contínuas. Os cursos são ofertados pelas instituições de educação profissional e o atendimento ao estudante é realizado tanto nas sedes das instituições quanto em suas unidades remotas, os polos.

Os parceiros da Rede e-Tec Brasil acreditam em uma educação profissional qualificada – integradora do ensino médio e da educação técnica – capaz de promover o cidadão com capacidades para produzir, mas também com autonomia diante das diferentes dimensões da realidade: cultural, social, familiar, esportiva, política e ética.

Nós acreditamos em você!

Desejamos sucesso na sua formação profissional!

Ministério da Educação
Abril de 2014

Nosso contato
etecbrasil@mec.gov.br



Indicação de Ícones

Os ícones são elementos gráficos utilizados para ampliar as formas de linguagem e facilitar a organização e a leitura hipertextual.



Atenção: indica pontos de maior relevância no texto.



Saiba mais: oferece novas informações que enriquecem o assunto ou “curiosidades” e notícias recentes relacionadas ao tema estudado.



Glossário: indica a definição de um termo, palavra ou expressão utilizada no texto.



Mídias integradas: remete o tema para outras fontes: livros, filmes, músicas, *sites*, programas de TV.



Atividades de aprendizagem: apresenta atividades em diferentes níveis de aprendizagem para que o estudante possa realizá-las e conferir o seu domínio do tema estudado.



Refleta: momento de uma pausa na leitura para refletir/escrever sobre pontos importantes e/ou questionamentos.



Palavra do Professor-autor

Prezado(a) estudante,

Bem-vindo(a) ao Curso Técnico Informática para Internet!

Estamos iniciando mais uma disciplina do nosso curso que vai enriquecer o seu conhecimento sobre o mundo da informática. Usei o termo “estamos” porque realmente vamos caminhar juntos. Vou estar ao seu lado em cada página, orientando, sugerindo, chamando sua atenção para conceitos importantes, provocando a sua curiosidade, avaliando o desenvolvimento do seu aprendizado, enfim, realizando essas e muitas outras tarefas em parceria com você. Embora este caderno seja um de vários necessários para sua formação profissional, dedique seu valioso tempo e sua atenção para adquirir novos conhecimentos e desenvolver competências e habilidades a partir da leitura do seu conteúdo, da realização das atividades e da interação com o ambiente virtual.

Bons estudos!

Professor João Batista Pinto Neto



Apresentação da Disciplina

Prezado(a) estudante,

O cenário atual do mercado de tecnologia da informação apresenta, segundo levantamentos recentes realizados pela equipe do Instituto Federal de Rondônia - IFRO, uma significativa necessidade de profissionais qualificados a pesquisar, conhecer, negociar, enfim, explorar os recursos da grande rede chamada Internet a fim de atender a usuários domésticos e empresas. Esta disciplina, em conjunto com as demais, fornecerá ferramentas para que você adquira as habilidades e competências necessárias para fazer parte desse grupo especial de profissionais especialistas em informática para Internet.

Nesta disciplina, vamos explorar os sistemas operacionais. Um componente fundamental dos sistemas computacionais e que está presente em todos os computadores atuais, sejam eles domésticos ou empresariais. Portanto, esta disciplina está relacionada com as disciplinas de programação, redes de computadores e banco de dados, e vai exigir de você um grande esforço que será recompensado pelos conhecimentos adquiridos ao final da unidade.

O seu aprendizado em sistemas operacionais está estruturado da seguinte forma: na Aula 1, serão apresentados os fundamentos dos sistemas operacionais; na Aula 2, vamos estudar os tipos de sistemas operacionais, suas funções e serviços; o conceito de processo, a figura central de um sistema operacional, vamos explorar na Aula 3; na Aula 4, vamos estudar a multiprogramação e as principais classes de processos; a gerência do tempo do processador e a gerência da memória serão abordados nas aulas 5 e 6; o estudo da interação sistema operacional com os periféricos será o foco da aula 7; e, finalizando, na aula 8, você vai conhecer como o sistema operacional implementa os sistemas de arquivos.



Sumário

Aula 1. Construindo a base	13
1.1 O que é sistema operacional ?	14
1.2 Software ou hardware?	15
Aula 2. Funções e tipos de sistemas operacionais	21
2.1 Funções do sistema operacional	21
2.2 Tipos de sistemas operacionais	24
Aula 3. Processos	29
3.1 O que é um processo?	30
Aula 4. Multiprogramação	35
4.1 Gerenciando processos	35
4.2 Executando processos	38
4.3 Tipos de processos	39
Aula 5. Escalonamento da UCP	45
5.1 Modo usuário e modo núcleo	47
5.2 Introdução ao escalonamento da UCP	49
Aula 6. Gerenciando a memória	55
6.1 A memória principal	55
6.2 Alocação de memória	57
6.3 Memória virtual	60
Aula 7. Gerência de Entrada e Saída	65
7.1 Dispositivos de Entrada e Saída	65
7.2 Interrupções	68
Aula 8. Sistema de arquivos	71
8.1 O que é um arquivo?	72
8.2 Sistemas de arquivos	74
8.3 Diretórios	75



Palavras Finais	78
Guia de Soluções	79
Referências	87
Obras Consultadas	88
Bibliografia Básica	88
Currículo do Professor-autor	89



Aula 1. Construindo a base

Objetivos:

- compreender o papel do sistema operacional na visão do usuário e na visão do *hardware*;
- distinguir o que é *hardware* e o que é *software*; e
- conceituar e identificar os principais tipos de *software*.

Olá! Preparado(a) para iniciar a nossa jornada? Espero que sim. Bem, nesta aula, vamos construir a base que vai sustentar o aprendizado sobre os sistemas operacionais. Vamos em frente?

Era uma vez...

... Em 1975, William abandonou o curso de direito em Harvard nos Estados Unidos para abrir uma empresa de fundo de quintal chamada Microsoft, em parceria com seu amigo Allen. A empresa produzia programas em linguagem BASIC para microcomputadores *Apple*, *Commodore* e *Altair*. Em 1980, a poderosa IBM iniciou o projeto *Chess* para construir um computador pessoal e convidou a pequena Microsoft para desenhar um sistema operacional para a nova máquina.



Fonte: sxc.hu

Fig. 1.1 – Interface de comando do MS-DOS



William e Allen desenvolveram o *Microsoft Disk Operating System*, ou MS-DOS. William não só vendeu o novo sistema operacional como convenceu a poderosa americana dos computadores (IBM) a tornar pública a especificação dos PCs para que outras empresas pudessem produzir *software* para a máquina. O resultado foi a proliferação de licenças para MS-DOS, alcançando mais de dois milhões de cópias vendidas em 1984. Adaptado de Yourdictionary, (YOURDICTIONARY, William Henry Gates III Biography, disponível em <<http://biography.yourdictionary.com/william-henry-gates-iii>>, acesso em 26 jan.2013).

.Essa é a história do MS-DOS, o vovô dos sistemas operacionais modernos e do seu criador William, que, com certeza, você já ouviu falar. Não? William Henry Gates III é nada mais nada menos que Bill Gates, uma das maiores fortunas do planeta, fundador da Microsoft, fabricante do sistema operacional mais usado no mundo atualmente, o Windows. Agora que acabamos com as formalidades, vamos entrar no clima da aula e conceituar sistema operacional.

1.1 O que é sistema operacional ?

Se você já usou um computador antes, então você já encarou um sistema operacional. Literalmente! Ao abrir um arquivo, ao usar um editor de texto ou ao acessar a Internet, você usou um sistema operacional para realizar essas tarefas. Ele é responsável por facilitar o uso do computador, mostrando uma interface amigável (Fig. 1.2) para que o usuário possa operar a máquina mesmo sem nenhum conhecimento do seu funcionamento.



Fonte: <http://informatica.hsw.uol.com.br/sistemas-operacionais3.htm>

Fig. 1.2 – Interface amigável de um sistema operacional



Quando você clica em um ícone na tela e uma janela se abre no monitor de vídeo, mostrando o relatório que você está escrevendo, ou uma planilha, ou uma imagem, foi o sistema operacional que interceptou o clique do mouse, identificou o objeto na área clicada, providenciou acesso ao disco e carregou na memória o aplicativo correspondente juntamente com o conteúdo do setor onde estavam armazenados os dados do objeto.

Santo clique! Pois bem. Você não precisa saber nada disso para abrir o seu aplicativo. Essa é uma das principais funções do sistema operacional. Facilitar a sua vida.

Um sistema operacional é um programa que atua como intermediário entre o usuário e o hardware de um computador (Silberschatz, 2010).



Nos primeiros sistemas operacionais, os aplicativos dos usuários tinham acesso direto aos periféricos, podendo, por exemplo, imprimir diretamente em uma impressora ou receber comandos de um controlador de jogo. Quando havia algum problema de comunicação entre o dispositivo e o aplicativo, o sistema travava, obrigando o usuário a reiniciar o computador manualmente.

O sistema operacional é um *software* básico de qualquer computador, pois fornece ao usuário uma interface conveniente e, ao mesmo tempo, gerencia o **hardware** controlando de forma ordenada e eficiente o acesso ao processador, memória e dispositivos de entrada e saída pelos aplicativos que os disputam.



Hardware

s.m. ing. Informática. Conjunto dos componentes eletrônicos de um computador (p. ex., processador, memória, disco rígido, monitor, equipamentos periféricos, etc.)

Tempo! Vamos fazer uma pausa para você refletir e reorganizar as ideias. Faça a atividade para fixar os conceitos.

Atividade 1.1

Foram mostradas nesta seção duas funções básicas do sistema operacional. Uma diz respeito ao usuário e a outra ao *hardware*. Quais são elas?



Software

s.m. ing. Conjunto de instruções armazenadas em disco(s) ou em chips internos do computador que determinam os programas básicos, utilitários ou aplicativos que ele tem para serem usados.

1.2 Software ou hardware?

Imagine o seu computador sem programas. Você liga, espera um pouco e o que acontece? Nada! Um computador sem programas não tem nenhuma utilidade. O programa ou **software** vai tornar o computador produtivo, permitindo a você armazenar informações em forma de dados, receber e enviar mensagens de correio eletrônico, ouvir música, assistir vídeos, entre outras





aplicações.

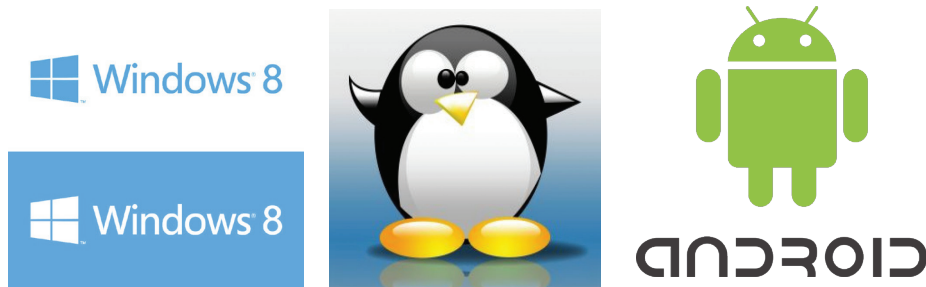
O termo *software* é usado para definir programas de uma maneira geral. Podemos classificar o *software* em três categorias básicas:



Segundo o site Baixaki Maranhão (Baixaki Maranhão. Android 2.2, o **Android**, sistema operacional do Google para dispositivos móveis, é hoje o mais popular entre os usuários. Segundo a empresa, todos os dias são ativados mais de um milhão e 300 mil aparelhos com este sistema. Lançada em 2010, a versão 2.2 apresentou, à época, atualizações que serviram como base para as posteriores. Disponível em < <http://baixakimaranhao.blogspot.com.br/2013/05/android-22.html> > Acesso em: 22 nov. de 2013.

- **Software de Sistema**

São programas responsáveis por controlar o *hardware*. Como, por exemplo, o sistema operacional Linux ou Windows ou ainda o Android. Podemos incluir nessa categoria os controladores (*drivers*) de dispositivo que fazem a interface entre os periféricos e o sistema operacional. Um teclado, por exemplo, precisa de um controlador para que o sistema operacional possa receber os dados que você digitar.



Fonte: <http://br.freepik.com>

- **Software de programação**

São programas para criar outros programas. Os desenvolvedores de *software* ou programadores, como linguagens de programação, compiladores, e outras ferramentas para produzir *software*. Ex. Java, C#, Python, etc.



Fonte: <http://br.freepik.com>

- **Software de Aplicação**

Esses programas, também chamados de aplicativos, podem ser programas de negócios, como editores de texto, apresentação, planilhas eletrônicas. Ou



ainda, podem ser programas educacionais, jogos, aplicações de multimídia e telefonia. Ex.: Ms Office, Open Office, Skype, CounterStrike, etc.

Podemos concluir que um *software* de sistema controla o *hardware*, e o *software* de sistema mais importante de um computador é o sistema operacional.



Fonte: <http://br.freepik.com>

O sistema operacional é uma camada de *software* colocada entre o *hardware* e os *aplicativos*, aqueles programas que fazem as tarefas dos usuários do computador (Oliveira, 2010).



Um computador ou, de maneira geral, um Sistema Computacional pode ser representado basicamente como um dispositivo composto de dois componentes: o *hardware* e o *software* (Fig. 1.3).



Fig. 1.3 – O Computador

Fonte: sxc.hu

O *hardware* fornece os recursos necessários de computação, como: processador, memória e dispositivos de entrada e saída. O *software* são os aplicativos e o sistema operacional que utilizam os recursos do *hardware*. Você pode imaginar o computador como competidor de fórmula 1, composto por um carro e um piloto (Fig. 1.4)





Fig. 1.4 – Competidor de fórmula 1.

Fonte: sxc.hu

Fazendo uma comparação entre o competidor de formula 1 e o computador, o *hardware* corresponde ao carro que, como o computador, não pode andar sozinho, necessitando do piloto, o *software*. Seu objetivo é ganhar a corrida.

Quanto melhor for o desempenho do motor, da suspensão e da aerodinâmica do carro (*hardware*), aliadas à perícia do piloto (*software*), maiores serão as chances de o competidor alcançar o objetivo.



Tempo! Hora da reflexão. Reúna os conceitos que estão na sua cuca para relacionar sistema operacional, *software* e *hardware*.



Curta esta charge genial do
Maurício Ricardo sobre sistemas
operacionais.
[http://charges.uol.com.
br/2011/10/06/especial-steve-
do-outro-lado-2](http://charges.uol.com.br/2011/10/06/especial-steve-do-outro-lado-2)

No computador, o processo ocorre de maneira similar. Quanto mais rápido for o processador, quanto maior for a capacidade de armazenamento da memória e quanto mais rápido forem os dispositivos de entrada e saída (carro), aliadas a eficiência do sistema operacional e aplicativos (piloto), melhor será o desempenho do computador, ou seja, as tarefas serão executadas com maior eficiência e rapidez.

Portanto, um computador com processadores ultrarrápidos, com grande capacidade de armazenamento de dados e dispositivos de entrada e saída de última geração pode ter seu desempenho comprometido se o sistema operacional e os aplicativos não controlarem os recursos de *hardware* adequadamente.

Ficou clara a relação entre *software* e *hardware*? Não? Uma volta de carro resolve o problema. O importante é que agora você sabe que o desempenho do computador é função do *hardware* e do *software*. Outra situação que fica clara com essa comparação é que quando o sistema operacional trava, todo o sistema computacional fica inoperante, ou seja, o carro não anda porque fica sem piloto!





Pé no freio! Uma parada no box para uma revisão geral. Faça a atividade para fixar os conceitos.

Atividade 1.2

Descreva com as suas palavras o significado de *software* e de *hardware* e como os dois se relacionam com o sistema operacional.



Resumo

Na primeira seção, você estudou os fundamentos do sistema operacional e descobriu que o conceito de sistema operacional está diretamente ligado ao significado de *software* e *hardware*. Avaliou também o papel de mediador do sistema operacional que, além de proteger o sistema de acesso indevido ao *hardware*, apresenta ao usuário uma interface amigável sem a complexidade da máquina. Na seção 1.2, enumeramos os diversos tipos de *software* e fizemos uma relação entre um competidor de fórmula 1 e o computador, o que permitiu a você concluir que um computador sem *software* é um carro sem piloto, ou seja, não tem nenhuma utilidade.

Atividades de aprendizagem

1. Um sistema operacional de última geração é instalado em um computador com um processador lento e com uma memória reduzida. Comente sobre o desempenho desse sistema computacional.

2. Complete o quadro informativo sobre *software* abaixo, descrevendo as características de cada tipo e informando, pelo menos, três exemplos de cada. Use as referências, pesquise na Internet ou consulte os seus colegas para enriquecer as suas respostas.



Tipos de Software	Características	Exemplos
Sistema		
Programação		
Aplicação		





Agora você tem a visão geral da natureza do sistema operacional e sua função básica no computador. Vamos acelerar, então, e explorar, na próxima aula, as funções e tipos de sistemas operacionais.



Aula 2. Funções e tipos de sistemas operacionais

Objetivos:

- conceituar sistema operacional através das tarefas;
- distinguir os diversos tipos de sistema operacional; e
- identificar as funções e serviços que o sistema operacional desempenha nos sistemas computacionais.

Bem vindo(a) à segunda aula da nossa disciplina. Começaremos tratando das funções do Sistema Operacional. Preparado(a)?

2.1 Funções do sistema operacional

Estudamos, na seção anterior, que o sistema operacional não permite aos aplicativos do usuário acesso direto ao *hardware*. Vimos que existem duas razões para isso. A primeira é esconder a complexidade da máquina para possibilitar ao usuário operá-la sem conhecimento da sua estrutura interna. A segunda razão é evitar instabilidade do sistema devido a problemas de comunicação entre aplicativos do usuário e dispositivos de E/S.

O sistema operacional cria, na verdade, um ambiente virtual mais seguro, mais fácil de entender, operar e programar (Tanenbaum, 2008).



A outra função básica do sistema operacional está relacionada com o controle do acesso aos recursos do sistema. Considere a seguinte situação: Você abre um programa editor de texto e um programa de planilha eletrônica e começa a trabalhar alternando entre um programa e outro. Você decide, então, imprimir os dois arquivos ao mesmo tempo na mesma impressora. O que acontece? Os arquivos serão impressos corretamente? Ufa! Ainda bem! Graças ao sistema operacional que organizou o acesso à impressora, o seu trabalho foi impresso perfeitamente.

Como você deve ter percebido, o gerenciamento de recursos de forma jus-



ta, organizada e protegida é função do sistema operacional. Essa função não se restringe somente aos recursos de *hardware*, mas também a informações, como arquivos e pastas.



Em resumo, controlar quem está usando qual recurso, garantir os pedidos de recursos, medir a utilização e resolver conflitos de diferentes programas é uma tarefa básica do sistema operacional.

Você agora conhece as duas funções básicas de um sistema operacional. Vamos usar esse conhecimento para subir mais um degrau e estudar funções mais específicas dos sistemas operacionais.

Você é daqueles usuários que abrem várias janelas e só trabalha com uma? Não? Ainda bem! Abrir janelas em sistemas operacionais, como o da Fig. 2.1, significa executar programas no computador.

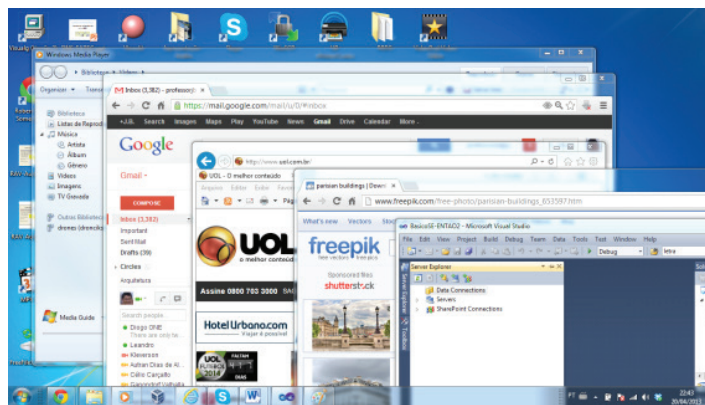


Fig. 2.1 – Abrindo janelas

Cada janela aberta é um programa em execução, que chamaremos de *processo*. Quanto maior o número de janelas abertas, maior é o trabalho do sistema operacional.



Para que um programa seja executado, ele deve ser carregado em um espaço exclusivo na memória, protegido, e deve ter o direito de controle da UCP. Cabe ao sistema operacional oferecer meios para que um programa seja carregado, executado e finalizado.

Uma das principais funções do computador é a sua utilização como ferramenta de criação, armazenamento e gerenciamento da informação. A variedade de mídias de armazenamento - como discos rígidos, fitas magnéticas, DVDs, discos de estado sólido (SSDs) e memórias USB existentes nos



computadores - é um problema, porque cada dispositivo tem uma forma de armazenamento diferente.

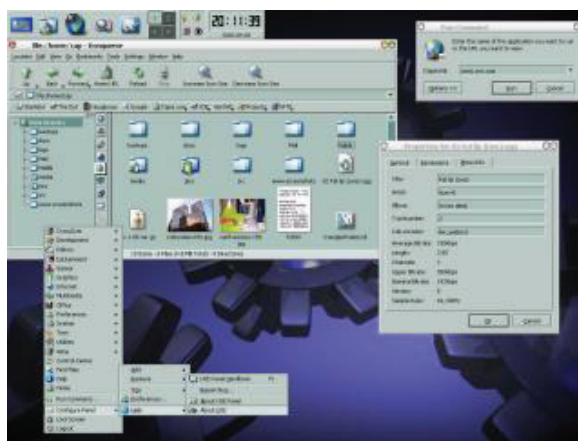


Fonte: <http://br.freepik.com>

O sistema operacional deve fornecer uma visão estruturada e organizada das informações, independente dos diferentes dispositivos de armazenamento.



Por exemplo, o sistema de arquivos para CD-ROM é o ISO 9660, enquanto que em disco rígido pode ser FAT32 ou NTFS, no caso do sistema operacional Windows, e EXT3, no caso do Linux.



Fonte: <http://www.kde.org>

Fig. 2.2 – Interface KDE

A Figura 2.2 mostra a interface gráfica KDE, comum nas edições do sistema operacional Linux, onde você pode ter uma visão lógica e uniforme dos recursos de armazenamento do computador implementada por intermédio de diretórios contendo arquivos e pastas.

Essa visão é apresentada para todas as unidades de armazenamento do sistema, independente do sistema de arquivos, desde que ele seja suportado pelo sistema operacional.

Tempo! Vamos fazer uma pausa para você refletir e reorganizar as ideias.





Faça a atividade para fixar os conceitos.



Atividade 2.1

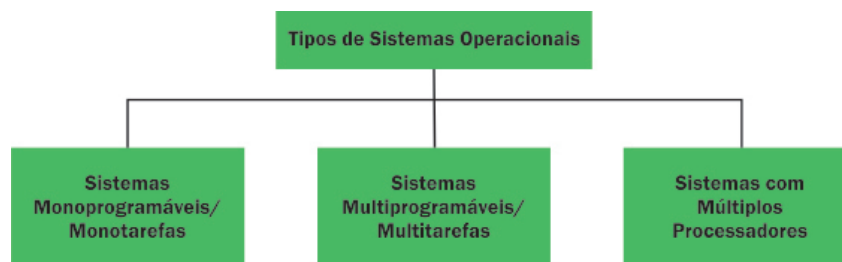
Faça uma lista das quatro funções do sistema operacional, explicando a finalidade de cada uma delas.

2.2 Tipos de sistemas operacionais

A maneira como os sistemas operacionais realizam tarefas pode servir para determinar sua classificação. Um sistema operacional pode ser classificado de acordo com o número de programas executados simultaneamente e pelo número de processadores por ele controlado (Fig. 2.3).



Acesse o site abaixo e veja um desenho animado sobre sistema operacional muito didático e ao mesmo tempo divertido. Confira!
<http://www.youtube.com/watch?v=ntOP8ZAYuUo>



Fonte: Machado (2011)

Fig. 2.3 – Tipos de sistemas operacionais

Historicamente, os tipos de sistemas operacionais estão relacionados com o *hardware*. Os primeiros sistemas operacionais surgiram com os *mainframes*, computadores de grande porte com capacidade de processamento de um volume grande de informações. Com a chegada dos microprocessadores, as opções de *hardware* se multiplicaram, gerando também uma grande variedade de sistemas operacionais.

2.2.1 Sistemas monoprogamáveis/monotarefas

O vovô MS-DOS foi um dos primeiros sistemas operacionais comerciais monoprogamáveis. Embora com todas as limitações inerentes ao sistema, foi um grande sucesso da época.



Nos sistemas monoprogamáveis todos os recursos do *hardware* estão a serviço de um único processo.

A restrição de executar um único programa causa uma utilização ineficiente dos recursos da máquina. Uma tarefa simples pode significar uma perda de tempo considerável para o usuário. Por exemplo, a impressão de um relatório com muitas páginas vai bloquear o sistema até que a última página seja





impressa e o processo de impressão seja finalizado.

2.2.2 Sistemas multiprogramáveis/multitarefas

A evolução dos sistemas operacionais monotarefas introduziu o conceito de multiprogramação, um ambiente onde uma única UCP é compartilhada juntamente com uma área de memória limitada. Nesse ambiente, o sistema operacional alterna a execução dos programas, dando uma falsa impressão ao usuário de que os programas estão sendo executados simultaneamente.

Um sistema multiprogramável tem uma taxa de utilização dos recursos da máquina acima de 90% em relação ao seu antecessor monotarefa.



Grande parte dessa taxa é atribuída ao tratamento de eventos nos quais os aplicativos de usuário fazem acesso a periféricos. Por exemplo, enquanto um programa faz um acesso a um dispositivo de E/S lento, como, por exemplo, um disco rígido, o sistema operacional transfere o controle da UCP para outro processo que está aguardando na fila, reduzindo o tempo ocioso do sistema e, em consequência, aumentando a eficiência do sistema.

Tempo é dinheiro! Esse velho bordão continua valendo e muito para os sistemas multitarefas. A redução do tempo de processamento e a execução de vários programas concorrentes proporcionam ao usuário ganho de produção e aumento da eficiência do computador.

Os sistemas operacionais multitarefas são encontrados na maioria dos computadores pessoais atuais. Versões recentes do Windows, MacOs e as principais edições do Linux são baseadas nessa tecnologia.

2.2.3 Sistemas com múltiplos processadores

Os sistemas com múltiplos processadores são semelhantes aos sistemas multitarefas. O que os diferencia é que os primeiros possuem mais dois processadores. Essa característica permite a execução paralela de tantos programas quantos forem o número de processadores do sistema computacional. Sistemas com múltiplos processadores foram desenvolvidos para atender a aplicações que exigem alto poder de processamento.

A maneira como é feito o acesso aos dispositivos de E/S e como a memória principal é conectada ao sistema computacional determinam o tipo de sistema operacional adequado para cada configuração.



Além das classificações dos sistemas operacionais abordadas nesta seção, o Prof. Silvestri apresenta a classificação segundo a arquitetura interna do sistema operacional e a classificação quanto a utilização do tempo do processador. Você pode ampliar seu conhecimento em sistemas operacionais acessando suas notas de aula. Confira! Disponível em < <http://gredes.iftto.edu.br/wp-content/uploads/SO-Aula004.pdf>> Acesso em 29 jan. 2013





Podemos enumerar também outras vantagens dos sistemas com múltiplos processadores como, por exemplo, a redundância que permite que o sistema continue a funcionar mesmo quando um dos processadores falha e o balanceamento de carga que permite dividir as tarefas.

Tempo! Vamos fazer uma pausa para você refletir e reorganizar as ideias. Faça a atividade para fixar os conceitos.



Atividade 2.2

Pesquise nas indicações das referências ou na Internet para encontrar uma aplicação para um sistema operacional com múltiplos processadores.

Muito bem! Mais uma aula completada com sucesso. Vamos agora relembrar os principais tópicos.

Resumo

Nesta aula, estudamos as funções e serviços do sistema operacional. Vimos que esconder a complexidade do *hardware* e organizar o acesso aos periféricos são duas funções básicas de um sistema operacional. Foi destacado também o gerenciamento de processos e o gerenciamento da informação em que você analisou como o sistema operacional mostra diferentes sistemas de arquivos em uma mesma interface. Conhecemos os tipos de sistemas operacionais, o que permitiu a você distinguir sistemas monotarefas de sistemas multitarefas e diferenciar um sistema multiprogramável de um sistema com múltiplos processadores. Você aprendeu também que a arquitetura do sistema computacional influencia significativamente a função dos sistemas operacionais.



Atividades de aprendizagem

1. Faça um quadro com os tipos de sistemas operacionais classificados nesta aula, contendo as seguintes colunas:

- tipo de sistema operacional;
- número de processadores admitidos;
- número de programas executados;

2. No sistema operacional Windows 8, a medida que cada janela é aberta o desempenho do computador fica comprometido. Por quê?



Caro(a) estudante, esta aula, juntamente com a anterior, fornecem os conceitos básicos necessários para o estudo mais profundo dos sistemas operacionais. Vamos começar na próxima aula um conceito fundamental dos sistemas operacionais: o processo.





Aula 3. Processos

Objetivos:

- distinguir a diferença entre processo e programa;
- identificar o contexto de um processo;
- identificar as funções do bloco de controle de processo;
- conceituar multiprogramação.

Seja muito bem vindo(a) a nossa terceira aula da disciplina Sistemas Operacionais. Vamos começar esta aula fazendo um paralelo entre do nosso assunto com a cozinha. Vamos por a mão na massa?

Colocando a mão na massa...

Uma nova receita surge como resultado de nossas experiências acumuladas a todo instante. Nada é mais legal na cozinha do que aprender a enxergar/compreender as transformações químicas que ocorrem o tempo todo. Receitas são fórmulas químicas. Acho absolutamente fascinante a diferença que a simples regulação de intensidade da chama pode causar. Da mesma forma, sutilezas, como cortar ingredientes em pedaços maiores ou menores, ou procedimentos podem fazer toda diferença no resultado final.

Um bom exemplo é o pão. Imagine que a imensa maioria dos pães que você já comeu foi feita com os mesmíssimos ingredientes: farinha de trigo, fermento biológico (em pó ou tabletes), água e sal! O que muda é o procedimento, as quantidades, a temperatura e o tipo de forno (Adaptado de Gradim, 2012).

Você deve estar se perguntando o que a receita de pão tem a ver com processo. Tem muito a ver! Um programa de computador é um texto com uma série de instruções, ou seja, a receita do pão. Assim como o padeiro faz o



pão segundo uma receita, o sistema operacional executa o processo segundo as instruções de um programa.



Um processo é a execução de um programa dentro de um ambiente fornecido pelo sistema operacional. Os ingredientes que o sistema operacional usa para executar o programa são os recursos disponíveis, como memória, tempo de processador e espaço em disco.

Portanto, assim como os pães não saem iguais, mesmo sendo feitos com a mesma receita, a execução de um mesmo programa pelo sistema operacional pode ser diferente, pois dependerá da oferta dos recursos existentes no momento em que o processo for criado. Construiu uma ideia do que é um processo? Tenho certeza que sim. Você vai se lembrar desse conceito quando tomar o seu café da manhã. Agora que você está conectado com o assunto, vamos conceituar formalmente um processo.

3.1 O que é um processo?

Você já aprendeu que todo sistema operacional oferece meios para que um programa seja carregado na memória principal e executado. Essa é uma das principais funções de um sistema operacional.

Um programa de computador pode ser escrito em uma grande variedade de linguagens de programação. Analise o programa abaixo escrito na linguagem C++.

```
#include <iostream>
Voidmain()
{
    Cout<< "Meu primeiro Programa em C++ !!!" <<endl;
}
```

Esse programa simples de apenas cinco linhas, quando executado, abre uma janela e mostra o texto que está entre aspas no código. Mas, para que ele seja executado, uma série de etapas precisa ser vencida. A Figura 3.1 mostra um diagrama com o caminho a ser percorrido.

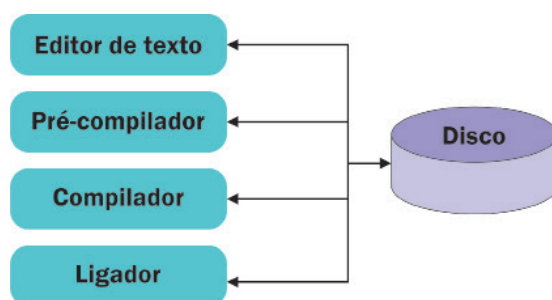


Fig. 3.1 – Compilação de um programa

O primeiro passo é escolher uma linguagem de programação, escrever o programa em um editor de texto e salvar no disco rígido. A seguir ele deve ser processado por um **compilador**, um programa que vai ler o arquivo original (código fonte), verificar se os comandos estão corretos (pré-compilação), gerar e salvar um texto codificado em *linguagem de máquina*, mostrado na Figura 3.2, que contém as instruções que o processador vai executar.



Fig. 3.2 – Código em linguagem de máquina

Observe que o programa original foi transformado em uma série de linhas contendo somente zeros e uns. Cada linha é uma instrução em linguagem de máquina ou código binário que será decodificada e executada pelo processador.

O último passo é feito pelo *ligador* que vai adicionar as funções declaradas no código fonte e as chamadas de sistema e “ligar” esses blocos de código, gerando o programa executável. Essas chamadas de sistema são rotinas do sistema operacional para acesso ao *hardware*, como, por exemplo, abrir uma janela no monitor ou escrever um caractere na tela, necessárias para a execução do nosso primeiro programa em C++ que agora está formatado na linguagem da UCP.

A-Z

Compilador

Que ou aquele que compila. Programa que converte as instruções de uma linguagem qualquer de programação (p. ex., Cobol, Pascal) para linguagem de máquina, passível de ser entendida e processada pelo computador (Aurélio, 2013).



Poucas linguagens são adequadas para implementação de sistemas operacionais como a linguagem C e a C++. Martins et. al mostram nessa primeira aula do curso de Programação Estruturada o desenvolvimento de programas utilizando compiladores gratuitos. Confira! http://www.metrooledigital.ufrn.br/aulas/disciplinas/prog_est/aula_01.html



Os programas são normalmente armazenados na memória secundária, ou memória de massa. Para que um programa seja executado, ele deve estar na memória principal.

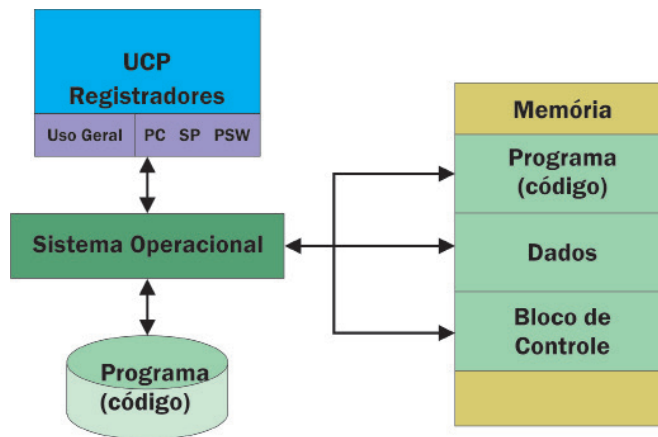


Fig. 3.3 – Cenário de execução de um processo

Então, vamos executar o nosso programa! É exatamente agora que o conceito de processo entra em cena. A Figura 3.3 mostra o cenário onde pode ser visto o diagrama de relacionamento dos atores. O sistema operacional reserva uma área na memória principal para o programa executável e seus dados e reserva, também, uma área para carregar o *Bloco de Controle do Processo* (BCP), uma estrutura de onde serão armazenados todos os dados necessários para o controle da execução do processo.

O programa, então, é transferido do disco para a memória principal, onde ficará aguardando o comando do sistema operacional para ser executado. O sistema operacional transfere, então, o controle para a Unidade Central de Processamento (UCP) que carrega o registrador *Contador de Programas* (PC) com o endereço de memória da primeira instrução e inicia a execução do processo.

Se, durante a execução, por algum motivo, tiver que ser interrompida, o sistema operacional salva no BCP o conteúdo de todos os registradores da UCP e das demais informações do processo e suspende a sua execução. Esta operação permite que o processo retorne ao estado de execução exatamente onde parou, sem perda de dados.



Um processo é, portanto, uma entidade dinâmica que só existe enquanto está em execução, é algo em atividade que pode receber processar e produzir dados.





Tempo! Vamos fazer uma pausa para você refletir e reorganizar as ideias. Faça a atividade para fixar os conceitos.

Atividade 3.1

Enumere as etapas necessárias para criação de um programa executável.



Chegamos ao final de mais uma aula. Você vai agora refrescar sua memória lendo o resumo a seguir.

Resumo

Nesta aula, você constatou que o processo é um dos mais importantes conceitos dos sistemas operacionais. Você estudou os fundamentos dos sistemas operacionais e identificou, nesta aula, que a principal função de um sistema operacional é o gerenciamento de processos. Construímos também o conceito de que um processo é uma entidade dinâmica, com uma estrutura muito bem definida, que permite ao sistema operacional controlar sua execução mesmo em um ambiente com vários processos disputando uma única UCP. Vimos também o conceito de comunicação entre processos.

Atividades de aprendizagem

1. Descreva os passos para que um programa se torne um processo.
2. Qual a função do BCP na interrupção de um processo?



Caro(a) estudante, você conferiu que processo é o motivo da existência do sistema operacional. Vamos à próxima aula explorar a multiprogramação. Um ambiente com dois ou mais processos disputando uma única UCP.





Aula 4. Multiprogramação

Objetivos:

- identificar filas e descrever os estados de um processo;
- descrever as mudanças de contexto de um processo;
- classificar os tipos de processo.

Que bom tê-lo(a) aqui novamente. Vamos para a nossa quarta aula? Já estamos chegando na metade da disciplina! Bem, para iniciarmos essa nova etapa, vamos trabalhar o gerenciamento de processos. Bons estudos!

4.1 Gerenciando processos

Considere um computador com um único processador (UCP) executando somente um processo. O sistema operacional prepara a área de trabalho do processo em um espaço reservado, correspondente ao total de memória disponível, transfere a execução para o processador e vai dormir! Sim, porque não há o que fazer a não ser que o processo acesse algum periférico ou termine.



Fig. 4.1– Multiprogramação



Considere agora a execução, no mesmo computador, de três processos simultâneos (Fig. 4.1).

O sistema operacional deve, então, reservar um terço do espaço disponível na memória para cada processo e alternar a execução pela UCP de forma justa, de modo que cada processo tenha direito a mesma fatia de tempo.

Esse procedimento deve ser feito com precisão e eficiência, protegendo as áreas de memória de cada processo e alternando sua execução de forma segura. O sistema operacional divide os recursos disponíveis da máquina física entre três máquinas virtuais, uma para cada processo. Essa técnica é chamada de multiprogramação.

Quando um processo termina ou um novo processo é criado, é preciso ajustar o uso dos recursos, que são limitados, para atender a nova configuração.

Você concorda que o gerenciamento de processos é uma das mais importantes tarefas de um sistema operacional?



Nos sistemas multiprogramáveis, o sistema operacional se preocupa em gerenciar o acesso concorrente aos seus diversos recursos, como memória, processador e periféricos, de forma ordenada e protegida, entre os diversos programas (Machado, 2011).

O gerenciamento de processos é também uma pedra no sapato dos projetistas de sistema operacionais. A Figura 4.1 mostra uma situação ideal, perfeita. No caso da memória, a área de trabalho de cada processo cabe exatamente no segmento alocado. Na prática isso não acontece, porque os processos ocupam espaços de tamanho variável e, à medida que o seu número aumenta, fica impossível acomodar todos na memória. As técnicas para resolver esses problemas serão estudadas mais adiante.

Da mesma forma, no caso do processador, a situação não é diferente, porque os processos executam uma grande variedade de tarefas. Processos que realizam muitos cálculos e raramente acessam periféricos ocupam mais tempo da UCP.

Em contrapartida, os processos que acessam periféricos com frequência passam grande parte do tempo aguardando, ocupando pouco tempo do processador. Esse é um problema clássico que tira o sono dos projetistas de



sistemas operacionais. Mas não vamos nos preocupar com isso agora. Por enquanto, vamos reforçar o conceito de multiprogramação.



Fig. 4.2 – 523 partidas simultâneas de xadrez

Em 2010, o israelense Alik Gershon venceu 454, empatou 58 e perdeu 11 das 523 partidas simultâneas, disputas em 19 horas na praça Yithzak Rabin, em TelAviv (Fig. 4.2). Alik está no livro Guinness de recordes. Você conhece o jogo de xadrez?

Em partidas simultâneas, o mestre joga com os desafiantes dispostos em um circuito fechado. Ele dedica um tempo para cada competidor, que só pode fazer o lance quando o mestre chega à sua frente. O mestre, então, analisa a jogada, faz a sua, zera o relógio e vai para o próximo desafiante, repetindo o procedimento até que todos os jogos sejam finalizados.

Bom, agora você pergunta: Qual a relação entre multiprogramação e o desafio do mestre Alik? Tem tudo a ver. Imagine que o mestre é a nossa UCP e que cada participante, juntamente com o seu tabuleiro, é um programa. À medida que os participantes são acomodados nos seus lugares, eles se tornam processos. Inicialmente, todos estão em estado de espera, aguardando a passagem do mestre. A passagem do mestre por um tabuleiro corresponde atribuir ao processo o estado de execução, e o tempo que o participante faz o lance e o mestre responde corresponde à fatia de tempo em que o controle da UCP é transferido para o processo. Ficou claro agora o conceito de multiprogramação? Não? Então, jogue uma partida de xadrez e releia a seção.

Tempo! Vamos fazer uma pausa para você refletir e reorganizar as ideias. Faça a atividade para fixar os conceitos.



O xadrez é um dos jogos mais antigos e populares do mundo. Ele é jogado em um tabuleiro com 16 peças para cada jogador e o objetivo é imobilizar o rei adversário (cheque mate).



Confira mais informações sobre o xadrez acessando a página da WEB:
<http://pt.wikipedia.org/wiki/Xadrez>



Atividade 4.1

É possível criar múltiplos processos de um mesmo programa? Justifique.

4.2 Executando processos

A limitação de uma única UCP permite somente um processo no estado de execução. E os outros? Os outros aguardam na fila. Quando o processo que está em execução termina, esgota a sua cota de tempo ou acessa um periférico, o sistema operacional suspende sua execução e passa o comando do processador para o primeiro da fila. Essa operação de alteração do estado do processo é chamada *mudança de contexto*. As transições de estado dependem do estado em que o processo se encontra (Fig. 4.3). Vamos conhecer os detalhes das mudanças de contexto de um processo:

- **Novo** → **Pronto** – O processo foi criado e é enviado para a fila de execução;
- **Pronto** → **Execução** – O processo sai do estado pronto e entra em execução segundo um **algoritmo** de escalonamento do sistema operacional;
- **Execução** → **Espera** – O processo em execução vai para o estado de espera quando o sistema operacional suspende sua execução por algum motivo ou ele executa uma operação de entrada e saída;
- **Espera** → **Pronto** – Depois de atendida a requisição, o processo sai do estado de espera e vai para o estado pronto. O processo em espera para retornar a execução vai obrigatoriamente para o estado pronto.

A-Z

Algoritmo

s.m. Matemática. Sequência de raciocínios ou operações que oferece a solução de certos problemas.



Fonte: Adaptado de Silberschatz(2004)

Fig. 4.3 – Diagrama de estados de um processo



- **Execução** → **Pronto** – Esta mudança ocorre normalmente quando o processo esgotou sua cota de tempo de UCP;
- **Execução** → **Terminado** – O processo encerrou sua execução normalmente ou foi encerrado por acesso indevido.

Nos sistemas operacionais modernos, o sistema operacional tem domínio completo sobre a UCP. Ele pode a qualquer tempo suspender a execução de qualquer processo através de um mecanismo chamado *interrupção*.

Tempo! Vamos fazer uma pausa para você refletir e reorganizar as ideias. Faça a atividade para fixar os conceitos.

Atividade 4.2

Qual o motivo da mudança de contexto dos processos em um sistema operacional multiprogramável?



4.3 Tipos de processos

Você sabe quantos programas estão sendo executados em seu computador? Aposto que não. Nem eu sei. São tantos tipos de processos. Mas vamos descobrir o que eles têm em comum. Pronto para conferir como os processos podem ser classificados? Então, vamos lá!

4.3.1 Processos classificados pelo tipo de tarefa

As tarefas realizadas pelos processos consomem recursos do sistema computacional. Podemos associar processos quanto à utilização da UCP e quanto ao acesso a periféricos:

- **Processos limitados pela UCP**

O algoritmo *raytracing* é usado em computação gráfica para **renderizar** imagens 3D com alto grau de realismo (Fig. 4.4).





Fonte: <http://www.nvidia.com/object/advanced-rendering.html>

Fig. 4.4 – Imagem renderizada com Ray Tracing

A-Z

Renderizar

É processo pelo qual pode-se obter o produto final de um processamento digital.

A cor de cada pixel da imagem é calculada levando em conta a luz refletida diretamente no ponto de vista do observador. Processos que envolvem cálculos matemáticos intensivos como o do algoritmo *raytracing* ocupam praticamente 100% da UCP. O tempo que ele leva para completar uma tarefa, como, por exemplo, renderizar a imagem acima, é dependente da velocidade da UCP.



Os processos limitados pela UCP fazem poucas requisições de entrada e saída. Em consequência disso, não são vistos na fila *espera* com frequência, mas, em contrapartida, são fregueses assíduos da fila *pronto*.

• Processos limitados por Entrada e Saída (E/S)

Se não for o mais usado, certamente está entre os 10 mais. Estou falando do seu navegador! Você o abre e ele já envia uma solicitação da sua página padrão para um servidor WEB através de uma conexão de rede para Internet. Alguns segundos mais tarde a carga da página é iniciada.



Fonte: <http://portal.mec.gov.br>

Fig. 4.5 – Navegando na WEB



Processos que realizam requisições de E/S como um navegador WEB (Fig. 4.5), mesmo com uma conexão de alta velocidade, ocupam muito pouco a UCP, pois passam a maior parte do tempo em comunicação com periféricos. O tempo que ele leva para completar uma tarefa, como, por exemplo, receber uma página completa de um portal, é dependente da velocidade do dispositivo de E/S, no caso, a conexão com a Internet.

Os processos limitados por E/S fazem muitas requisições de entrada e saída. Em consequência disso, são vistos raramente na fila *pronto*, mas, em contrapartida, são fregueses assíduos da fila *espera*.



4.3.2 Processos classificados pela forma de comunicação

Processos recebem informações, processam e apresentam resultados. Podemos caracterizar um processo associando-o aos canais de E/S usados para comunicação, como, por exemplo:

- **Processos em primeiro plano**

Também chamados de processos interativos, os processos em primeiro plano exigem a comunicação direta com o usuário.



Fonte: <http://www.alientrap.org/games/nexuiz>

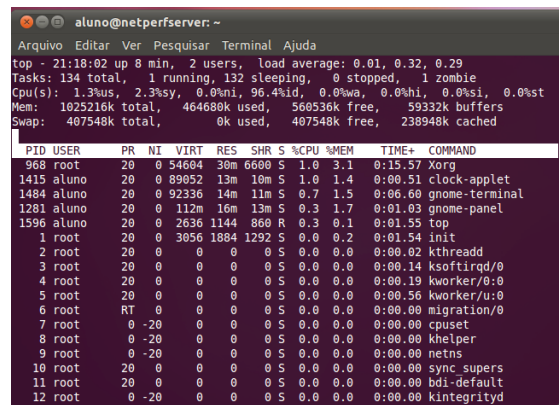
Fig. 4.6 – Cena do jogo Nexuiz

Jogos, como o Nexuiz (Fig. 4.6), executam processos que recebem comandos de entrada do usuário através do mouse, teclado ou um joystick. Esses processos mostram os resultados dos comandos através do monitor de vídeo (saída).



- **Processos em segundo plano**

Diferentemente dos processos interativos, os processos em segundo plano não se comunicam diretamente com o usuário. Normalmente, a comunicação de um processo em segundo plano é feita através de arquivos. Os serviços de um sistema operacional Ubuntu (Fig.4.7), também chamados de *daemons* (pronuncia-se dimons), são exemplos de processos executados em segundo plano.



```
aluno@netperfserver: ~
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
top - 21:18:02 up 8 min, 2 users, load average: 0.01, 0.32, 0.29
Tasks: 134 total, 1 running, 132 sleeping, 0 stopped, 1 zombie
Cpu(s): 1.3%us, 2.3%sy, 0.0%ni, 96.4%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1025216k total, 464688k used, 560536k free, 59332k buffers
Swap: 407548k total, 0k used, 407548k free, 238948k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
  968 root        20   0 54604 30m 6680 S   1.0   3.1   0:15.57 Xorg
1415 aluno      20   0 89052 13m 10m S   1.0   1.4   0:00.51 clock-applet
1484 aluno      20   0 92336 14m 11m S   0.7   1.5   0:00.00 gnome-terminal
1281 aluno      20   0 112m 16m 13m S   0.3   1.7   0:01.03 gnome-panel
1596 aluno      20   0 2636 114k 860 R   0.3   0.1   0:01.55 top
    1 root        20   0 3056 1884 1292 S   0.0   0.2   0:01.54 init
    2 root        20   0      0      0 S   0.0   0.0   0:00.02 kthreadd
    3 root        20   0      0      0 S   0.0   0.0   0:00.14 ksoftirqd/0
    4 root        20   0      0      0 S   0.0   0.0   0:00.19 kworker/0:0
    5 root        20   0      0      0 S   0.0   0.0   0:00.56 kworker/u:0
    6 root        RT   0      0      0 S   0.0   0.0   0:00.00 migration/0
    7 root        0 -20      0      0 S   0.0   0.0   0:00.00 cpuset
    8 root        0 -20      0      0 S   0.0   0.0   0:00.00 khelper
    9 root        0 -20      0      0 S   0.0   0.0   0:00.00 netns
   10 root        20   0      0      0 S   0.0   0.0   0:00.00 sync supers
   11 root        20   0      0      0 S   0.0   0.0   0:00.00 bdi-default
   12 root        0 -20      0      0 S   0.0   0.0   0:00.00 kintegrityd
```

Fig. 4.7– Processos em segundo plano no Ubuntu.

Tempo! Vamos fazer uma pausa para você refletir e reorganizar as ideias. Faça a atividade para fixar os conceitos.



Atividade 4.3

Como você classifica o processo de um editor de texto responsável por capturar a digitação.

Hora da retrospectiva. Vamos relembrar os tópicos estudados nesta aula.

Resumo

Estudamos, nesta aula, o conceito de multiprogramação, em que você constatou que o processo é a figura central dessa técnica que torna possível a execução de múltiplos processos em sequência, aparentando execuções simultâneas. Vimos que o sistema operacional trabalha muito para compartilhar os recursos, como o tempo da UCP e o espaço de memória, de modo a atender com eficiência a execução de todos os processos. Vimos também como o sistema operacional organiza as filas de processos baseadas em seus estados e como ele gerencia a mudança de contexto.



Atividades de aprendizagem



1. Compare o trabalho do processador com um sistema monoprogramável e com um sistema multiprogramável.

2. Um processo A é limitado por UCP, enquanto um processo B é limitado por E/S. Sabendo que os dois processos tem o mesmo número de instruções, qual deles será executado primeiro? Justifique.

Parabéns! Mais uma aula concluída. Prepare-se agora para conhecer como o sistema operacional controla a execução de processos.





Aula 5. Escalonamento da UCP

Objetivos:

- conceituar escalonamento;
- distinguir modo usuário de modo núcleo na execução de processos;
- reconhecer as ações dos módulos *escalonador* e *despachante*; e
- identificar as formas básicas de escalonamento.

Seja bem-vindo(a) à nossa sétima aula da disciplina Sistemas Operacionais. Estamos entrando na segunda metade da nossa disciplina. A metade do caminho já foi percorrida! Vamos começar a nossa aula revendo conceitos sobre filas e aplicando-os aos sistemas operacionais.



Fonte: sxc.hu

Furando a fila...

Nós aprendemos desde pequenos no jardim de infância: não devemos furar a fila! Existem consequências para quem desobedece a uma das regras mais básicas da sociedade: que você aguarde pacientemente na fila atrás da pessoa que chegou antes de você, não importa quanto tempo demore. Mas, será que esse é sempre o caso? Com certeza, você consegue identificar uma



situação em que as pessoas normalmente furam a fila DEPOIS de pedir para fazer isso, por exemplo, quando o ato de furar a fila é feito com o consentimento da outra pessoa, ainda que ofereçam uma desculpa para furar a fila.

Essa é uma prática comum em filas de segurança nos aeroportos, quando as pessoas pedem para furar a fila para evitar perder o voo...

... Até mesmo na Inglaterra, onde impera a regra “primeiro que entra, primeiro que sai”, comportamento semelhante é observado na fila para compra de passagens de trens: alguém que está quase perdendo seu trem consegue passar na frente dos outros, se pedir e informar esse fato (Adaptado de Coelho, 2012).



Uma das razões para a formação das filas é quando o tempo de atendimento dedicado a uma pessoa é maior que o tempo de chegada da outra. É exatamente o que acontece no ambiente de multiprogramação! Enquanto a UCP atende um processo, uma fila se forma devido a grande quantidade de processos de um computador.



Fonte: sxc.hu

Pessoas têm necessidades diferentes, que resultam em diferentes tempos de atendimento, assim como os processos que têm tamanhos variáveis, resultando em tempos de execução variáveis. Pessoas também não são iguais e devem receber tratamento diferenciado na hora do atendimento, como um idoso ou um portador de necessidades especiais, ou ainda uma mulher grávida. Assim como as pessoas, os processos têm características diferentes e, se forem tratados da mesma maneira, alguns terão ótimo desempenho enquanto outros serão prejudicados.





O responsável pela organização, controle e atribuição de prioridades é o sistema operacional. É ele que controla o tempo que a UCP dedica a cada processo, que depende do seu tipo, da sua função e do seu tamanho. E com ele não tem conversa, ninguém fura a fila!



Agora que você já está por dentro do assunto que vamos explorar nesta aula, vamos conceituar formalmente o gerenciamento da UCP.

5.1 Modo usuário e modo núcleo

Você estudou, na aula 4, que no ambiente da multiprogramação com uma única UCP, é necessário fazer uma escala de execução para poder atender um a um os processos prontos para a execução. Esse procedimento é chamado de *Escalação da UCP*.

O escalonamento da UCP é a base dos sistemas operacionais multiprogramados. A partir da redistribuição da UCP entre os processos, o sistema operacional pode tornar o computador mais produtivo (Silberschatz, 2004).



Para realizar o escalonamento, o sistema operacional usa processos próprios, que são carregados em uma área de memória protegida chamada de núcleo do sistema operacional, mostrado na Figura 5.1.

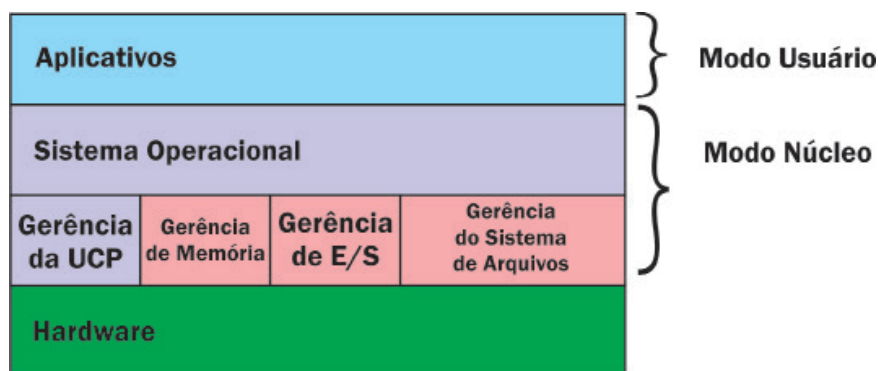


Fig. 5.1 – Núcleo do sistema operacional

O núcleo do sistema operacional tem quatro módulos básicos:

- **Gerência da UCP** – responsável pelo escalonamento da UCP e pela mudança de contexto dos processos;
- **Gerência de Memória** – responsável pela alocação de memória para os processos;





- **Gerência de E/S** – responsável pelo controle de acesso aos periféricos, atendendo requisições de processos e controlando interrupções da UCP;
- **Gerência do Sistema de Arquivos** – responsável pela organização dos dados em arquivos e pastas, controle de acesso e segurança.



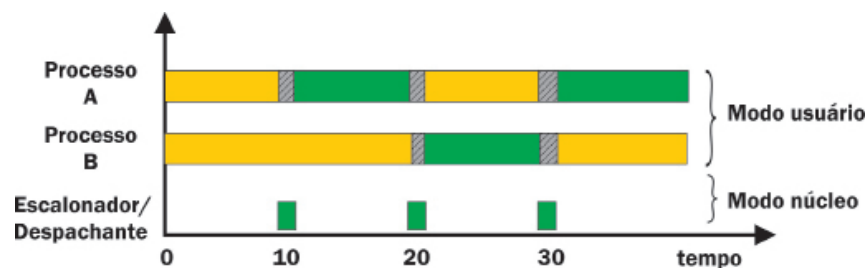
A-Z

Kernel

(ing.) é o núcleo de qualquer software, o código, programa ou conjunto de programas principais.

Os processos do núcleo ou **kernel** são processos de sistema que tem acesso completo ao *hardware* e à memória principal. São executados em modo protegido em espaços exclusivos na memória. Já os processos de usuário não têm acesso ao *hardware* e só podem endereçar memória dentro da sua área de trabalho.

Você pode deduzir, então, que a gerência da UCP é feita alternando a execução de processos em **modo núcleo** com processos em **modo usuário** (Fig. 5.1). Para clarear o seu juízo, analise o gráfico a seguir, que mostra a execução de dois processos de usuário.



Fonte: Adaptado de Machado (2011)

Fig. 5.2 – Escalonamento de UCP

Suponha que o seu computador está iniciando a execução de dois processos A e B (Fig. 5.2). No instante de tempo zero, o sistema operacional inicia um processo de sistema que vamos chamar de *escalonador*. Esse processo carrega na memória os dois processos e atualiza a fila *pronto*, com o processo A em primeiro e B em segundo.

As duas faixas horizontais contínuas do gráfico representam os processos A e B, e os blocos inferiores, os processos de sistema. As áreas amarelas mostram o tempo de permanência dos processos na fila *pronto*. As áreas verdes mostram o tempo de execução em que os processos detêm o controle da UCP. As áreas **hachuradas** mostram o tempo de transição de um estado para outro (Ex. *pronto* – executando). Entre o instante de tempo zero e o instante de tempo 10, os dois processos estão na fila *pronto* (área amarela) aguardando a execução.

A-Z

Hachura

s.f. Traços paralelos ou cruzados que se empregam nos desenhos, pinturas ou gravuras para marcar as sombras e meias-tintas.



No instante 10, o escalonador entra em ação chamando um processo de sistema denominado *despachante* para iniciar a execução do processo A.

O despachante é um módulo do núcleo do sistema operacional responsável por passar o controle da UCP para um processo. Ele muda o contexto do processo e alterna para o modo usuário.



Essa operação é executada em modo núcleo no menor espaço de tempo possível. Entre os instantes de tempo 10 e 20, o controle da UCP é do processo A (área verde), executado em modo usuário.

No instante 20, o despachante interrompe o processo A, o coloca na fila *pronto* e inicia a execução do processo B. A partir desse instante, quem tem o controle da UCP é o processo B, também executado em modo usuário. O procedimento se repete e o despachante vai alternar o controle da UCP entre os dois processos até que eles encerrem.

Ficou claro agora o conceito de modo núcleo e modo usuário? Tenho certeza que sim! Na prática, o sistema operacional distribui entre os processos o tempo da UCP e usa o menor tempo possível com os processos em modo núcleo.

Tempo! Vamos fazer uma pausa para você refletir e reorganizar as ideias. Faça a atividade para fixar os conceitos.

Atividade 5.1

Analise a Figura 5.2 e faça um gráfico incluindo mais um processo (C, por exemplo) na fila *pronto* no instante de tempo zero. Observe que nesse cenário os processos têm fatias de tempo de UCP iguais.



5.2 Introdução ao escalonamento da UCP

Você, como usuário de computador, já deve ter experimentado abrir aplicativos e amargar longos tempos de espera até que eles estejam disponíveis para uso. Vários fatores contribuem para que isso ocorra, e um deles é a ineficiência do sistema operacional.

Todos os sistemas operacionais têm políticas e critérios que orientam as ações de seus escalonadores para que distribuam o tempo da UCP da melhor maneira possível. Essas ações são procedimentos denominados *algoritmos de escalonamento*.



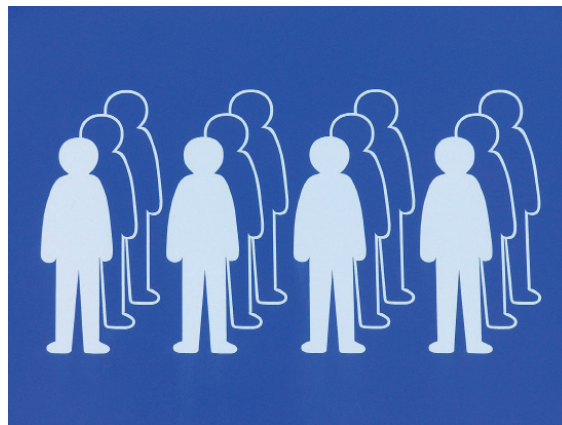


Para que você entenda a função dos algoritmos de escalonamento, vamos estudar dois algoritmos bastante simples usados nos primeiros sistemas operacionais. Para ajudar nessa tarefa, vamos estabelecer as seguintes medidas ou métricas para os critérios de escalonamento:

- **Produtividade (P_p)** – é o número de processos por unidade de tempo, medida em processos/segundo;
- **Tempo de espera (T_e)** – tempo total que o processo fica na fila pronto durante a sua execução, medido em milésimos de segundo (ms);
- **Tempo de Execução (T_x)** – tempo total compreendido do instante em que o processo é criado até o instante do término da sua execução, medido em milésimos de segundo (ms);
- **Utilização da UCP** – taxa de ocupação da UCP, mede a carga da CPU, medida em porcentagem (%).

Agora você está apto a conhecer e avaliar os algoritmos de escalonamento. Vamos começar com o algoritmo PEPS.

5.2.1 Algoritmo primeiro a entrar primeiro a sair



Fonte: sxc.hu

O algoritmo primeiro a entrar primeiro a sair (PEPS) é bastante simples. Exige somente uma fila em que o primeiro processo a entrar é o primeiro a sair. Se o processo que está sendo executado sofre bloqueio por uma requisição de E/S e vai para o estado de espera, o primeiro processo da fila *pronto* é executado. Quando o processo bloqueado é liberado, ele retorna para o final da fila. Este algoritmo foi muito usado nos primeiros sistemas operacionais





monoprogramáveis, com escalonamento sem preempção. Vamos avaliar o seu desempenho para o tempo de espera (T_e) e o tempo de execução (T_x).

Suponha que entrem três processos na fila *pronto* com os seguintes tempos de processamento:

Processo A = 20ms, Processo B = 2ms e Processo C = 5ms.

Para essa ordem, os tempos de espera e execução total estão mostrados na Tabela 5.1.

Tabela 5.1 – Algoritmo PEPS – ordem original

Ordem	Tempo de processamento	T_e	T_x
A	20	0	20
B	2	20	22
C	5	22	27
Média		14	23

Veja que, embora o processo B necessite de somente 2ms para ser executado, seu tempo de execução é de 22ms porque ele amarga 20ms na fila esperando o processo A (espaçoso!) ser executado. Isso não é justo, certo? Então, vamos mudar a ordem de entrada dos processos e refazer os cálculos (Tabela 5.2)

Tabela 5.2 – Algoritmo PEPS – ordem alterada

Ordem	Tempo de processamento	T_e	T_x
B	2	0	2
C	5	2	7
A	20	5	27
Média		2,3	12

Que diferença! Você viu? Bastou trocar a ordem de execução e os tempos ficaram excelentes. É exatamente o que faz o algoritmo a seguir.

5.2.2 Algoritmo menor primeiro

Este algoritmo seleciona para execução o processo que tiver menor tempo de processamento. No caso do exemplo anterior, o escalonador faria a escolha idêntica a da Tabela 5.2. O problema deste algoritmo é estimar o tempo de processamento de cada processo que entra na fila. Uma maneira de resolver o problema é calcular o tempo de processamento baseado nas execuções anteriores do processo.

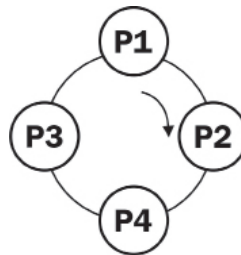


Você pode ter informações adicionais sobre algoritmos de escalonamento acessando as notas de aula do Professor Marcelo Johann no site abaixo. Confira!

<http://www.inf.ufrgs.br/~johann/sisop1/aula10.scheduling.pdf>

5.2.3 Algoritmo por alternância circular

Neste algoritmo, também chamado de Round Robin, o escalonador dedica um tempo fixo denominado *quantum* para cada processo da fila *pronto* sequencialmente até o final, retornando ao primeiro e reiniciando o processo continuamente.



Fonte: Adaptado de Machado (2011)

Fig. 5.3 – Escalonamento por alternância circular

O problema deste algoritmo é o tamanho do quantum. Se for muito pequeno fica ineficiente, causa muitas mudanças de contexto. Se for muito grande torna o tempo de resposta inaceitável, principalmente para processos interativos.

Você conferiu que os algoritmos de escalonamento são fundamentais para o desempenho do sistema operacional. Técnicas como critérios de prioridade dos processos, usadas em conjunto com os algoritmos mostrados anteriormente, aumentam mais ainda a eficiência do sistema. Consulte a bibliografia básica e a dica a seguir para mais detalhes sobre os algoritmos de escalonamento da UCP.



Algoritmos de escalonamento são ainda objeto de pesquisa e a cada dia uma nova maneira de escalonar é proposta.

Tempo! Vamos fazer uma pausa para você refletir e reorganizar as ideias. Faça a atividade para fixar os conceitos.



Atividade 5.2

Dentre os três algoritmos de escalonamentos estudados, qual é o mais justo? Justifique.

Vamos relembrar os tópicos estudados nesta aula. Refresque a memória lendo o resumo da aula.



Resumo

Nesta aula, você teve uma visão geral da gerência da UCP. Conheceu o núcleo do sistema operacional, também chamado de *kernel*, que auxiliou a você distinguir a execução de processos em modo usuário da execução de processos em modo núcleo. Estudou também o papel do escalonador e do despachante no processo de escalonamento. Abordamos os principais critérios para um escalonamento justo e eficiente, e você experimentou a dificuldade de determinar uma política de escalonamento ideal, que atenda a todos os tipos de processo. Apresentamos, em seguida, as métricas para avaliar o desempenho dos algoritmos com relação a produtividade, tempo de execução, tempo total de espera entre outros. E, finalmente, estudamos dois algoritmos de escalonamento, destacando suas características e suas deficiências.

Atividades de aprendizagem



1. Complete o quadro abaixo com as informações sobre os eventos que o escalonador precisa para tomar decisões.

Evento	Descrição	Motivo
1		Início da execução do processo
2	O processo sai do estado de <i>execução</i> e vai para o estado <i>pronto</i>	
3		Retorno de uma requisição de E/S
4	O processo sai do estado de <i>execução</i> e vai para o estado <i>espera</i>	

2. Qual a influência dos algoritmos de escalonamento nos sistemas operacionais?

3. Suponha que quatro processos A, B, C e D devam ser executados. Sabendo que seus tempos de processamento são 4, 32, 7 e 25ms, respectivamente, faça uma comparação de desempenho entre o algoritmo PEPS e o algoritmo MP.

Parabéns, mais uma aula concluída. Os conhecimentos adquiridos nesta aula serão a base para o entendimento da gerência da memória, que abordaremos na próxima aula.



Aula 6. Gerenciando a memória

Objetivos:

- reconhecer o conceito de endereço lógico e físico;
- avaliar a gerência de memória usando partições;
- descrever paginação de memória;
- conceituar a memória virtual; e
- identificar as políticas de substituição de páginas.

Seja bem-vindo(a) a mais uma aula da nossa disciplina. Para dar início à essa nossa sexta aula, vamos começar tratando do assunto Memória Principal. Pronto(a)?

6.1 A memória principal

A memória principal ou memória física é um módulo do *hardware* composto de circuitos eletrônicos que armazenam dados na forma binária. Os dados na memória são organizados em *bytes* (8 bits), que podem ser acessados através de um *endereço físico*.

Cada posição de memória tem um endereço único correspondente a uma combinação de *bits* das linhas de endereços do circuito eletrônico. O número de linhas de endereço determina a capacidade máxima de memória do sistema computacional, assim, o computador, quando sai da fábrica, tem já estabelecida a quantidade máxima de dados que podem ser armazenados na memória principal.

A Figura 6.1 mostra uma memória com capacidade para 4 bytes. Para ter acesso a cada posição da memória, temos que ter 2 linhas de endereçamento.

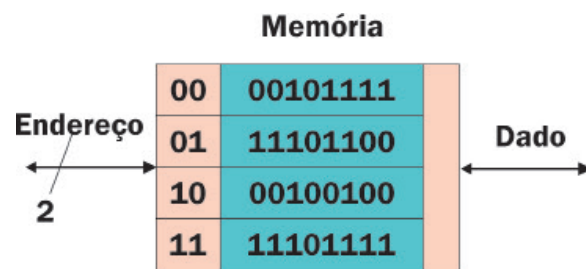


Fig. 6.1 – Endereçamento físico de memória

Para acessar a primeira posição de memória, o endereço deve ser 00, para a segunda, 01, e assim por diante. Assim, generalizando, para n linhas de endereço, a nossa memória terá capacidade para armazenar 2^n bytes.



Hora da reflexão. Analise a Figura 6.1 e calcule quantas linhas de endereço seriam necessárias para endereçar uma memória de 4GB

Conseguiu? Se você calculou 32 linhas, parabéns! Vamos usar esse conceito de endereçamento físico para entender como os processos endereçam dados na memória.

Quando um programa é criado, o compilador referencia os endereços de memória a partir do endereço onde o processo será carregado. Mas como saber? Impossível. Assim o endereço de referência para o processo é o endereço zero, a base da memória. O problema é que outros processos usam também o mesmo endereço de referência. Para resolver esse problema foi proposto o conceito de *endereço lógico*.



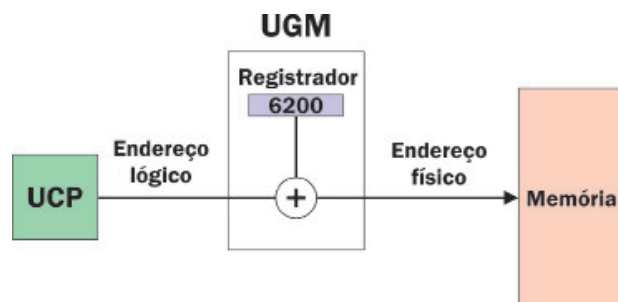
Unidade de Gerenciamento de Memória (UGM)

Do inglês *Memory management Unit* (MMU), é responsável por gerenciar a memória física, a memória do sistema operacional e a memória dos aplicativos. Normalmente está integrada à UCP, embora em alguns sistemas seja um módulo independente.

Os endereços gerados pelos processos quando estão no comando da UCP são endereços lógicos. Esses endereços são traduzidos para os endereços físicos pela **Unidade de Gerenciamento de Memória** (UGM).

No carregamento do processo na memória, um dispositivo de *hardware*, a Unidade de Gerenciamento de Memória, atualiza um registrador com o endereço físico correspondente à posição de memória da primeira instrução do processo. Esse valor, somado ao endereço lógico, gera o endereço físico correspondente (Fig. 6.2).





Fonte: Siberschatz (2004)

Fig. 6.2 – Tradução de endereços lógicos para físicos

Suponha que o processo foi carregado a partir do endereço físico 6200. Esse valor será somado a todos os endereços gerados pela UCP. Por exemplo, se o processo buscar uma instrução no endereço lógico 300, a UGM vai gerar o endereço real, físico, $6500 = 6200 + 300$.

Tempo! Vamos fazer uma pausa para você refletir e reorganizar as ideias. Faça a atividade para fixar os conceitos.

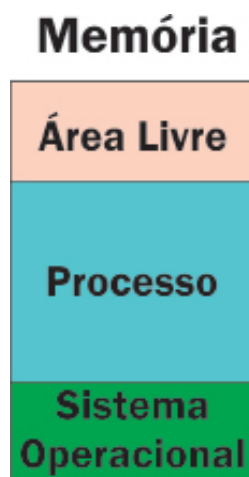
Atividade 6.1

Explique a função da Unidade de Gerenciamento da Memória (UGM).



6.2 Alocação de memória

O espaço da memória principal é dividido em duas partes. Uma para o sistema operacional e outra para os processos.



Fonte: Tanenbaum (2008)

Fig. 6.3 – Alocação de memória – Sistemas monoprogramáveis

Nos sistemas monoprogramáveis, a alocação de memória é bastante simples, porque temos somente um processo em execução (Fig. 6.3). O sistema



operacional é normalmente carregado na parte baixa da memória e o restante do espaço é ocupado pelo processo. Caso o tamanho do processo seja maior do que o espaço disponível, ele não será carregado.

Nos sistemas multiprogramáveis, a necessidade de alocar memória para vários processos obrigou os projetistas de sistemas operacionais a dividir a memória. Vamos conferir como eles resolveram o problema.

6.2.1 Particionando a memória

Os primeiros sistemas usavam partições fixas. Essas partições eram configuradas na inicialização do sistema. Os processos eram distribuídos de acordo com o tamanho na partição imediatamente maior, ou seja, processos pequenos em partições pequenas (Fig. 6.4). Como os processos têm tamanhos variáveis, esta solução gera **fragmentação**. As áreas livres dentro das partições não eram usadas, tornando o uso da memória ineficiente.

A-Z
Fragmentação
S.f. Ação de se partir em fragmentos. Quebrar em pequenos pedaços

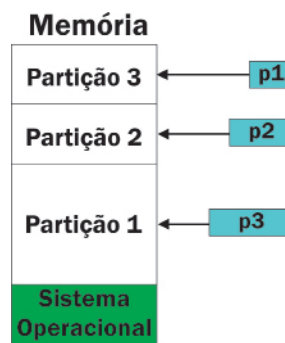


Fig. 6.4 – Alocação de memória – Partições fixas

Quando os processos eram retirados da memória, os espaços deixados não eram contíguos, formando “buracos”, obrigando a fazer manutenção no sistema periodicamente para restaurar o espaço original de memória. Para resolver este problema, uma solução foi proposta: a paginação de memória que estudaremos a seguir.

6.2.2 Paginação de memória

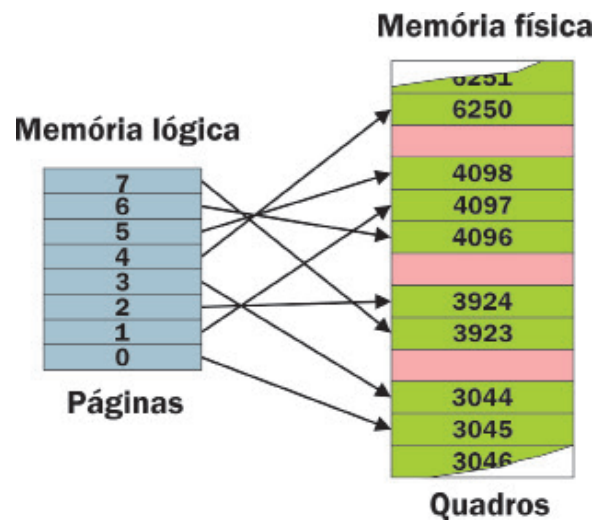
A paginação é uma técnica de gerenciamento de memória que não necessita de espaços **contíguos** de memória para carregar os processos. Na prática, um processo pode estar espalhado na memória, dividido em vários pedaços que não estão necessariamente juntos.

Para entender como funciona a paginação, vamos conceituar dois tipos de memória:

A-Z
Contíguo
Adj. Imediato, próximo, junto. Que toca em uma coisa: quarto contíguo a outro.

- Memória Lógica – espaço de endereçamento lógico composto pelos endereços gerados pela UCP;
- Memória física – espaço de endereçamento físico composto pelas posições reais de memória.

Na página, a memória física é dividida em blocos de tamanho fixo, denominados *quadros*, e a memória lógica é dividida em blocos de tamanho idêntico aos quadros, denominados *páginas*. Temos, então, uma correspondência direta entre páginas e quadros (Fig. 6.5).



Fonte: Tanenbaum (2010)

Fig. 6.5 – Correspondência entre página e quadro

Quando um programa é carregado na memória, ele é distribuído entre os quadros disponíveis, e a Unidade de Gerenciamento da Memória (UGM) é responsável por fazer a correspondência dos quadros da memória física com as páginas da memória lógica (Silberschatz, 2004).

Observe que, para o processo, ele está fazendo um endereçamento em um espaço contíguo, linear. Na realidade, ele está endereçando em um espaço totalmente disjuncto, espalhado. A vantagem é que o processo é carregado nos blocos disponíveis, não importando se eles estão juntos ou separados, evitando a fragmentação, comum no sistema de gerenciamento com partições.

Analise a Figura 6.5 e considere a seguinte situação: se um processo tem dez páginas, quantos quadros devem ter disponíveis na memória para que ele seja carregado?





Se você chegou à conclusão que, se o processo tem dez páginas, ele só poderá ser carregado na memória se existirem no mínimo dez quadros disponíveis, você está absolutamente certo!

Tempo! Vamos fazer uma pausa para você refletir e reorganizar as ideias. Faça a atividade para fixar os conceitos.



Atividade 6.2

Qual é o benefício da paginação de memória?

6.3 Memória virtual

Manter a maior quantidade de processos em execução é o objetivo principal da multiprogramação. Porém, à medida que o número de processos aumenta, o espaço de memória física diminui e, eventualmente, se esgota. A técnica de paginação cria uma imagem de uma memória contígua, que facilita o endereçamento feito pelo processo, mas exige que o processo inteiro esteja carregado na memória para que ele seja executado. Resumindo, não resolvem o problema de falta de espaço da memória física. A solução é a memória virtual que vamos explorar nesta aula.

6.3.1 Conceituando a memória virtual

Você já sabe que um programa precisa estar carregado na memória para poder ser executado, mas será que é preciso que o programa inteiro, incluindo o código principal, as sub-rotinas, as estruturas de dados e demais módulos estejam carregados? Não! Na realidade, somente a parte que está sendo executada precisa estar na memória. Assim, é possível manter as partes do processo que não estão sendo executadas na memória secundária (memória de massa) e carregar somente as partes necessárias para a execução.



Memória virtual é uma técnica sofisticada e poderosa de gerência de memória, em que as memórias principal e secundária são combinadas dando ao usuário uma ilusão de existir uma memória muito maior do que a capacidade real da memória principal (Machado, 2011).

Com essa técnica, é possível aumentar o número de processos concorrentes e ao mesmo tempo oferecer ao processo uma memória lógica teoricamente sem limitação de tamanho. A limitação seria o espaço disponível na memória física combinado com o espaço da memória secundária, que pode ser aumentado indefinidamente.

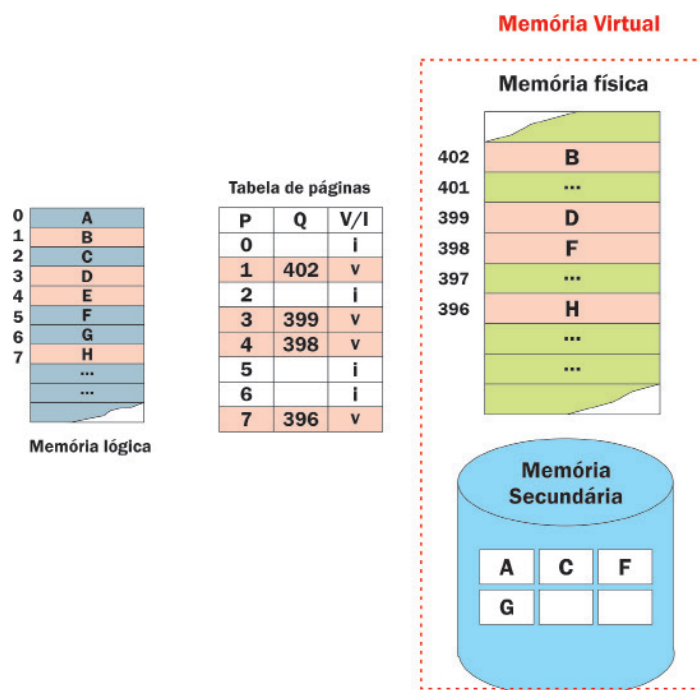


Esse é o conceito de memória virtual. Ficou claro para você? Muito bem! Então vamos em frente para você entender como é possível executar uma quantidade de processos que normalmente não caberiam na memória principal.

6.3.2 Memória virtual com paginação por demanda

Na paginação por demanda, somente algumas páginas da memória lógica são carregadas na memória física, embora para o processo toda a memória lógica esteja disponível a qualquer tempo. Para controlar quais as páginas que estão efetivamente carregadas, a UGM inclui mais uma coluna na tabela de páginas (Fig. 6.6) para informar se a página é válida (v) ou inválida (i). Caso seja válida, a página tem um quadro correspondente na memória física, caso contrário, a página está na memória secundária.

Quando o processo endereça uma instrução, a UGM verifica se a página contendo o endereço da instrução é válida. Se for válida, o endereço da página é traduzido apontando para o quadro vinculado a ele. Por exemplo, na Fig. 6.6, o módulo D ocupa a página 3. Se uma instrução for endereçada dentro deste módulo, a UGM vai fazer o endereçamento dentro do quadro 399 da memória física.



Fonte: Silberschatz (2004)

Fig. 6.6 – Memória Virtual – Paginação por demanda



A-Z

Falta de página

ou falha de página (*pagefault* em inglês), no contexto da tecnologia da memória dos computadores, é uma interrupção (ou exceção) disparada pelo hardware quando um programa acessa uma página mapeada no espaço de memória virtual, mas que não foi carregada na memória física do computador.

Até aqui tudo bem, certo? Ok. Vamos agora analisar o caso em que a instrução endereçada está na página A. Neste caso a UGM não vai encontrar o quadro correspondente porque a página é inválida. Imediatamente, a UCP inicia uma *exceção de falta de página*. Essa exceção é tratada pelo sistema operacional como uma operação de E/S e o processo entra em estado de espera. Nesse intervalo, a página A, que se encontra na memória secundária, é transferida para um quadro disponível na memória física. Após a transferência, o escalonador posiciona o processo na fila *pronto* onde ele aguardará a oportunidade de obter o controle da UCP.

Uma variação dessa técnica é a memória virtual usando paginação antecipada. O sistema operacional, em vez de carregar somente a página inválida, carrega também páginas subsequentes. Como os programas são executados sequencialmente, as páginas na sequência da página inválida têm grande possibilidade de serem solicitadas.

Tempo! Vamos fazer uma pausa para você refletir e reorganizar as ideias. Faça a atividade para fixar os conceitos.



Atividade 6.3

Qual é o benefício da memória virtual com paginação por demanda para o sistema operacional?

Chegamos ao final de mais uma aula. Você vai agora refrescar sua memória lendo o resumo a seguir.

Resumo

Nesta aula, vimos como é feito o endereçamento dos dados da memória principal. Você aprendeu também a distinguir endereço lógico de endereço físico. Conhecemos a Unidade de Gerência da Memória (UGM), que fornece o suporte de *hardware* para a tradução de endereços entre a memória lógica e a memória física. Vimos as primeiras formas de alocação de memória usando partições e estudamos com detalhes a paginação que usa o espaço de memória com maior eficiência. Em seguida, você aprendeu o conceito central de memória virtual e as vantagens do uso dessa técnica. Verificamos como é realizado o endereçamento dos quadros a partir da requisição de páginas feita pelos processos. Estudamos também como é feita a substituição de páginas na memória virtual com paginação, destacando as políticas de substituição.



Atividades de aprendizagem



1. Quantas linhas de endereço são necessárias para endereçar uma memória física de 1GB?
2. Explique como a UGM realiza a tradução de páginas para quadros na memória virtual com paginação.
3. Analise a Figura 6.6 e determine qual seria a ação da UGM se o processo requisitasse as páginas D, E, F, G, e H?

Muito bem! Você concluiu mais uma etapa. Na próxima aula você terá oportunidade de conhecer como os periféricos se comunicam com o sistema operacional. Não deixe de realizar as atividades de aprendizagem.



Aula 7. Gerência de Entrada e Saída

Objetivos:

- reconhecer os dispositivos de entrada e saída;
- identificar os mecanismos de interrupções; e
- conceituar os subsistemas de E/S.

Estamos nos aproximando da última aula da disciplina de Sistemas Operacionais. Que pena que já está acabando, mas, por outro lado, vai ficando o sabor do dever cumprido e de mais um ciclo que se completa. Aproveite os últimos encontros para avançar em seu processo de aprendizagem.

Você, no dia-a-dia, interage com o computador fornecendo informações por meio dos dispositivos de entrada e saída. A maioria dos benefícios que o computador lhe oferece é disponibilizada através de dispositivos como monitores, impressoras, scanners, teclado, mouse, entre outros. Portanto, você deve ter percebido que o gerenciamento dos sistemas de entrada e saída é uma das funções mais importantes do sistema operacional.

7.1 Dispositivos de Entrada e Saída

A base da estrutura de gerência de E/S é o *hardware*, a comunicação com o mundo real. Uma grande quantidade de dispositivos pode ser conectada a um computador, e o sistema operacional deve ter suporte para permitir os processos de usuário e de sistema para cada um deles. Em um sistema computacional, temos basicamente dois tipos de dispositivos de E/S:

- Dispositivos de bloco – armazenam informações em blocos de tamanho fixo, cada um com seu próprio endereço. O sistema operacional faz operações de busca e gravação através dos endereços. Os discos rígidos estão nessa classe de dispositivo.
- Dispositivos de caractere – operam com fluxos de caracteres. Não são



endereçáveis e não têm operações de busca. O teclado, o mouse e as interfaces de rede estão nessa classe.



Nem todos os dispositivos de E/S se encaixam nessas duas classes. Cada dispositivo se comunica com o sistema operacional de uma forma diferente. Alguns através de portas seriais, outros usando barramentos USB, outros usando conectores internos, como adaptadores de vídeo, dispositivos de captura de áudio e vídeo, entre outros.

7.1.1 Controladores de dispositivo

Todos os dispositivos de E/S têm um *controlador*, um conjunto de componentes eletrônicos que fazem a comunicação do dispositivo com o sistema computacional (Fig. 7.1).

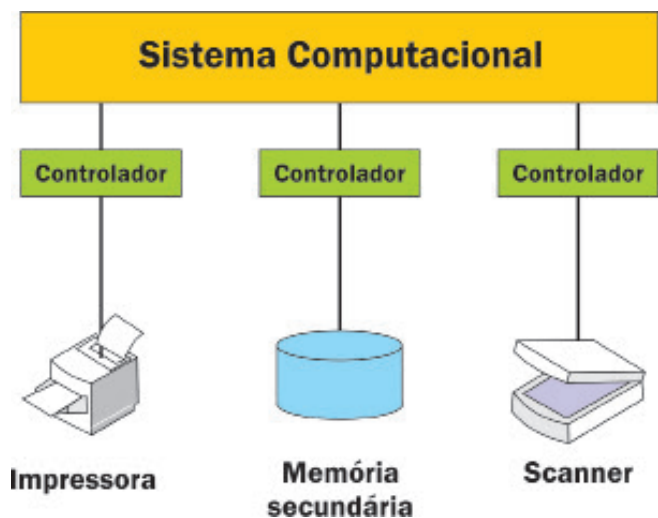


Fig. 7.1 – Controladores de dispositivo

Os controladores têm registradores próprios e alguns deles possuem memória, que são usadas para transferência de dados.

A responsabilidade por essa comunicação é do sistema operacional, através de um componente chamado driver de dispositivo, que vai traduzir os comandos do sistema operacional para o dispositivo e receber os dados enviados por eles.



Buffer

É uma região da memória física utilizada para armazenar dados temporariamente enquanto eles estão sendo movidos de um local para outro.

A transferência de dados pode ser feita em bloco ou em caractere, dependendo do tipo de dispositivo. O sistema operacional reserva para cada dispositivo uma área de memória denominada **buffer** para armazenar os dados de leitura e escrita. Quando o controlador recebe um bloco ou uma sequ-

ência de caracteres do dispositivo, é iniciada a transferência para o *buffer* interno pela UCP ou diretamente do dispositivo para a memória.

7.1.2 Drivers de dispositivo

Segundo Priscila Faccioli:

os *drivers* têm como função receber comandos gerais sobre acessos aos dispositivos e traduzi-los para comandos específicos, que poderão ser executados pelos controladores. Disponível em: < <http://pt.scribd.com/doc/68893986/SIS013-Aula1-Subsist-Em-A-E-S-Kernel> > Acesso em: 22 nov.2013.

Cada *driver* manipula somente um tipo de dispositivo ou grupo de dispositivos semelhantes (Fig. 7.2).

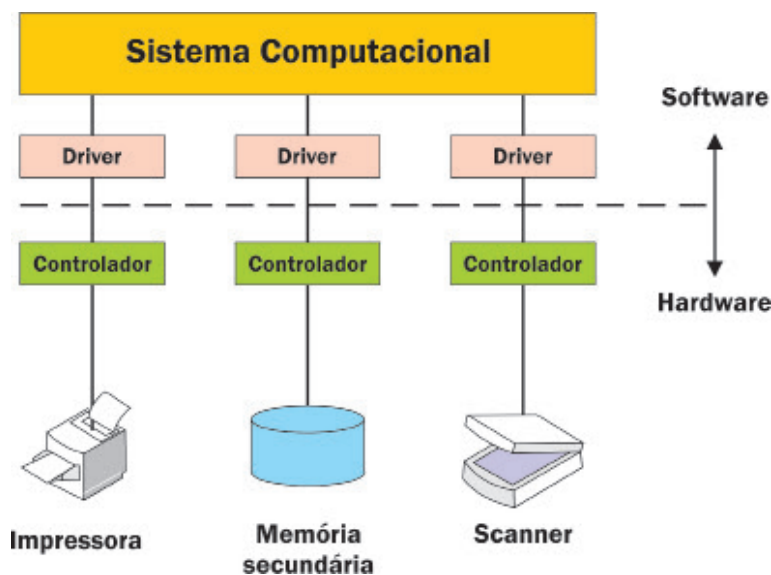


Fig. 7.2 – Drivers de dispositivo

A principal função do *driver* é proporcionar uma interface entre o controlador e o gerenciador de E/S independente do dispositivo conectado.



Um *driver* é um programa escrito em linguagem de máquina para um dispositivo específico.



Confira este vídeo montado por KolKsiano, que apresenta de uma forma bem radical os principais periféricos de um computador moderno.

<http://www.youtube.com/watch?v=8lyKAKz8CIA>

Quando o dispositivo é substituído ou o dispositivo é conectado em outro computador com sistema operacional diferente, o driver precisa ser atualizado para funcionar corretamente.

Tempo! Vamos fazer uma pausa para você refletir e reorganizar as ideias. Faça a atividade para fixar os conceitos.



Atividade 7.1

Faça uma pesquisa para encontrar dois dispositivos que se comunicam com o sistema operacional por blocos e dois por caracteres.

7.2 Interrupções

O mecanismo de interrupções é um conjunto formado por rotinas e componentes do *hardware* usado pelo sistema operacional para controlar as requisições de E/S.



Todo processador tem uma linha física usada pelo dispositivo de E/S ou pelo processo para solicitar acesso a um periférico. Quando essa linha é acionada, o processador interrompe a tarefa em curso, salva os dados referentes a ela e entrega o controle ao sistema operacional que executa uma rotina chamada *manipulador de interrupção*.

Essa rotina que manipula a interrupção é responsável por identificar o requisitante, o atendente, e realizar a tarefa requisitada, que pode ser a leitura de um caractere do teclado, um bloco de dados da memória secundária ou ainda enviar uma sequência de caracteres para a interface de rede.

Quando você está digitando um texto em um editor, cada tecla acionada corresponde uma interrupção da UCP. Da mesma forma, quando você está navegando na Internet, cada sequência de caracteres recebida do servidor WEB provoca uma interrupção.

Imagine o professor em uma sala de aula explicando um determinado assunto e, de repente, um aluno levanta a mão.





Fonte: sxc.hu

Fig. 7.3 – Interrompendo o professor

Imediatamente, o professor interrompe a explicação, atende ao aluno, respondendo ao seu questionamento e retorna ao ponto onde parou, dando prosseguimento à aula (Fig. 7.3).

O mecanismo de interrupção em um computador funciona da mesma forma. O processador é o professor e os alunos são os periféricos (mouse, teclado, disco rígido, câmera, etc). Quando um periférico levanta uma linha de interrupção, o processador para sua atividade, salva os registros em uso e atende ao periférico. Após o processamento da interrupção, ele recupera os registros salvos e retoma a atividade do ponto onde parou.

Ficou claro agora a função das interrupções em um sistema operacional? Não? Então, levante a mão e interrompa seu tutor para que ele tire suas dúvidas!

Hora da reflexão. Quando você movimenta o *mouse* do seu computador, quais os eventos que acontecem até que o ponteiro se mova na tela? Tente fazer uma sequência de passos.

Sua sequência pode ser diferente. Quando você movimenta o *mouse*, ele envia uma sequência de dados indicando movimento em uma determinada direção. O controlador do *mouse* detecta o movimento e envia os dados para o *driver* do mouse. O *driver* aciona a linha de interrupção e assume o controle da UCP para tratar a interrupção. O *mouse* começa a se movimentar na tela.



As notas de aula do Prof. Luis, do grupo GTA da Universidade Federal do Rio de Janeiro, são uma excelente referência para o estudo dos sistemas operacionais. Selecionei a parte 5 – Entrada e Saída, que compreende o assunto abordado na nossa aula. Confira!
<http://www.gta.ufrj.br/ensino/EEL580/apresentacoes/Parte5.pdf>





Tempo! Vamos fazer uma pausa para você refletir e reorganizar as ideias. Faça a atividade para fixar os conceitos.



Atividade 7.2

Você substituiu uma impressora por outra de modelo diferente. Qual componente deve ser atualizado no sistema operacional?

Chegamos ao final de mais uma aula. Você vai agora refrescar sua memória lendo o resumo a seguir.

Resumo

Exploramos, nesta aula, a gerência de Entrada e Saída. Começamos estudando os dispositivos de E/S, base da estrutura de gerência de E/S. Em seguida, estudamos os controladores de dispositivos que estão acoplados aos periféricos. Mostramos como é feita comunicação entre o sistema operacional e os dispositivos através dos *drivers*. Estudamos as interrupções responsáveis por iniciar e terminar as requisições de entrada e saída dos processos, assim como as requisições dos dispositivos.



Atividades de aprendizagem

1. Explique a função dos controladores de dispositivos.
2. Você está trabalhando com um aplicativo de edição de texto. Descreva, passo a passo, identificando os componentes envolvidos, os eventos que acontecem quando você digita uma letra no teclado.

Caro(a) estudante,

Parabéns, você concluiu mais uma aula. Na próxima aula, vamos abordar um tema muito interessante para a sua formação: o sistema de arquivos, item fundamental do estudo dos sistemas operacionais.

Aula 8. Sistema de arquivos

Objetivos:

- reconhecer o conceito de biblioteca;
- identificar o que são arquivos;
- distinguir o que são diretórios; e
- diferenciar um arquivo de um diretório.

Estamos no nosso último encontro. Nesta aula, vamos trabalhar os sistemas de arquivos, assunto tão importante quanto os que já foram estudados até aqui. Para iniciarmos, imagine que você está em uma biblioteca...

Visitando a biblioteca...

Biblioteca é uma palavra de origem grega. Trata-se de um espaço físico em que se guardam livros. Em geral, biblioteca é todo espaço, seja ele concreto ou virtual que reúne uma coleção de informações de qualquer tipo, sejam elas escritas em folhas de papel, como enciclopédias, dicionário, monografias, etc, ou digitalizadas e armazenadas em produtos como CD, DVD e banco de dados.

O bibliotecário organiza os itens do acervo conforme um padrão, cuja busca, convenientemente, procura deixar mais fácil, sem ocupar muitas prateleiras e assegurando a integridade deste. Ainda, certamente, os organiza segundo suas características (assunto, tipo de documento, data, etc.). Depois de organizados, ou durante a organização, o bibliotecário cria uma lista com todos os livros da biblioteca, com seus assuntos, localizações e códigos respectivos (Adaptado de <http://pt.wikipedia.org>).

Mas qual a relação entre uma biblioteca e um sistema de arquivos? Você deve estar se perguntando! Eu vou dizer. A biblioteca é o dispositivo de



armazenamento para o computador e o bibliotecário é o sistema de arquivos. Da mesma forma que o bibliotecário organiza o acervo da biblioteca, o sistema de arquivos guarda os dados através de arquivos, identificados com um nome, com a posição onde ele foi gravado, com o tamanho e outras informações necessárias para o gerenciamento pelo sistema operacional.

Ficou clara a função do sistema de arquivos? Muito bem! Vamos, então, conceituar formalmente o sistema de arquivos, os arquivos e diretórios.

8.1 O que é um arquivo?

Para o usuário, não há possibilidade de armazenar dados a não ser através de arquivos, ou seja, os arquivos são as unidades de armazenamento de informação fechadas como uma caixa com fichas (Fig. 8.1). A caixa representa o arquivo e as fichas os blocos de dados.



Fonte: sxc.hu

Fig. 8.1 – Visão de um arquivo

Os arquivos podem representar dados numéricos, alfanuméricos ou binários. Podem ser arquivos de texto, de música, de vídeo, programas contendo instruções em linguagem de máquina, enfim, um arquivo é uma sequência de bits, bytes, linhas ou registros, cujo significado é definido por quem o criou.



Um arquivo é um conjunto identificado de informações relacionadas com dados armazenados em memória secundária (disco rígido, SSD, DVD, pen-drive, etc.)

Os arquivos são identificados por nomes formados por um conjunto de caracteres **alfanuméricos**. A atribuição de nomes depende do sistema operacional, porém, normalmente um nome de arquivo tem duas partes separa-





das por um ponto. A primeira parte é a identificação definida pelo criador do arquivo e a segunda, denominada extensão, mostra o tipo de arquivo. Por exemplo, *agenda.txt* é um arquivo com os endereços e telefones dos seus amigos e é um arquivo de texto identificado pela extensão *txt*. As extensões mais comuns são:

- **.avi** - arquivo de vídeo (é o formato dos arquivos DivX);
- **.bmp** - arquivo de imagem;
- **.bin** - arquivo binário executável;
- **.dll** – biblioteca de sistema;
- **.exe** - arquivo executável;
- **.html** - linguagem de programação para Web;
- **.jpg** - arquivo de imagem comprimido;
- **.mp3** – arquivo de áudio;
- **.pdf** - documento do Adobe Acrobat Reader;
- **.png** - arquivo de imagem;
- **.txt** - arquivo de texto;
- **.wav** - arquivo de áudio sem compressão;
- **.zip** – Arquivo comprimido.

Hora da reflexão. Os dados em um disco rígido são armazenados em setores na forma binária (zeros e uns). Como salvar e como recuperar os dados nesse dispositivo?



Vamos conferir o seu raciocínio. Se você respondeu que para salvar e recuperar dados em um disco rígido basta usar a estrutura de um arquivo, parabéns! Você está absolutamente certo!





Tempo! Vamos fazer uma pausa para você refletir e reorganizar as ideias. Faça a atividade para fixar os conceitos.



Atividade 8.1

Defina arquivo, o que ele pode representar e onde ele é armazenado no computador.

8.2 Sistemas de arquivos

A sala de visitas do sistema operacional, a interface de maior visibilidade e onde existe mais interação com o usuário é o sistema de arquivos. Os processos armazenam e recuperam dados a partir de dispositivos de *hardware* diferentes e devem mostrar ao usuário uma interface amigável e de fácil manipulação.

A-Z

Dispositivos de armazenamento em massa.

São memórias secundárias que armazenam grande volume de dados. Discos rígidos, disco de estado sólido e fitas magnéticas são alguns exemplos desses dispositivos.

Um sistema de arquivos é um conjunto de estruturas lógicas e de rotinas, que permitem ao sistema operacional controlar o acesso aos **dispositivos de armazenamento em massa**. Os sistemas podem suportar mais de um sistema de arquivo, desde que não estejam nas mesmas partições. Por exemplo, algumas versões do Windows suportam o sistema de arquivos FAT32 e o NTFS.

O sistema de arquivos estabelece, para cada dispositivo, uma organização para leitura e gravação. Por exemplo: um dispositivo de armazenamento em fita é organizado como uma sequência de blocos. A leitura é feita sequencialmente e a gravação, sempre após o último registro. Dependendo de como está organizado o sistema de arquivos, podemos recuperar registros ou blocos de arquivos de várias maneiras. Essas formas de leitura e gravação dos arquivos na memória secundária são chamadas de *métodos de acesso* e os principais você pode conferir no box abaixo:

- **Acesso sequencial** - o acesso é restrito à ordem em que os registros foram gravados, sendo que a gravação de novos registros só é possível no final do arquivo;
- **Acesso indexado ou por chave** - o arquivo deve possuir uma área de índice onde existam ponteiros para os diversos registros e, a partir desta informação, realiza-se um acesso direto;
- **Acesso direto** - permite a leitura/gravação de um registro direta-





mente na sua posição. A forma mais simples de acesso.

Disponível em: <http://www2.icmc.usp.br/~luisc/download/fundamentos/sistemas_de_arquivos.ppt> Acesso em: 8 out. 2013

Tempo! Vamos fazer uma pausa para você refletir e reorganizar as ideias. Faça a atividade para fixar os conceitos.

Atividade 8.2

Cite uma função do sistema de arquivos em um sistema operacional?



8.3 Diretórios

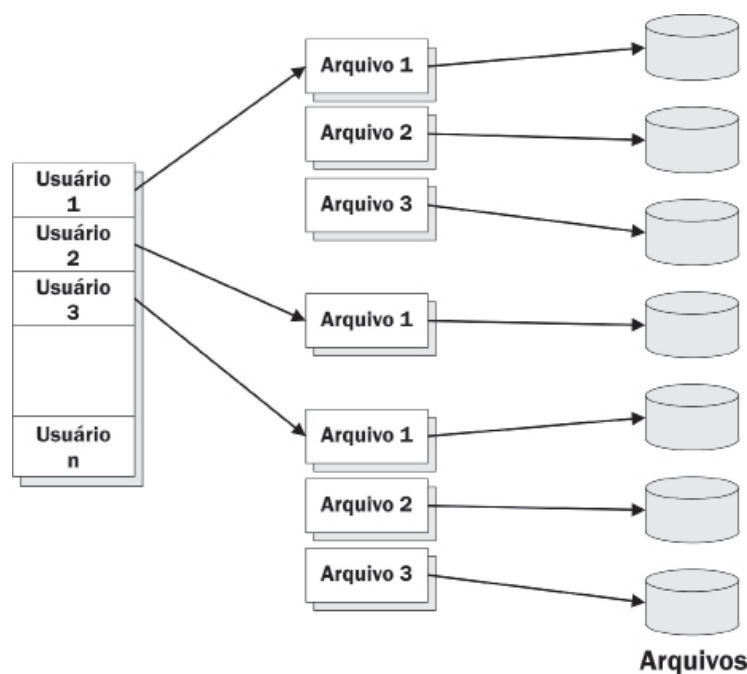
Sistemas operacionais podem trabalhar com vários sistemas de arquivos, que podem ser bastante extensos. Alguns sistemas armazenam milhões de arquivos ocupando dezenas e até centenas de terabytes em disco. Para gerenciar todos esses dados, precisamos organizá-los. Essa organização envolve o uso de diretórios.

Os diretórios são a forma como um sistema de arquivos organiza os arquivos em diferentes dispositivos de armazenamento. É a estrutura de dados que contém entradas associadas aos arquivos onde estão informações como localização física, nome, organização, etc.

O sistema de arquivos organiza logicamente os arquivos em uma estrutura de diretórios. São elas:

- Estrutura de diretório em nível único - todos os arquivos fazem parte do mesmo diretório, sendo necessário nome único para cada arquivo;
- Estrutura de diretórios em dois níveis - diretórios separados para cada usuário e no diretório principal mantem-se apenas os endereços dos diretórios dos usuários (Fig. 8.2).

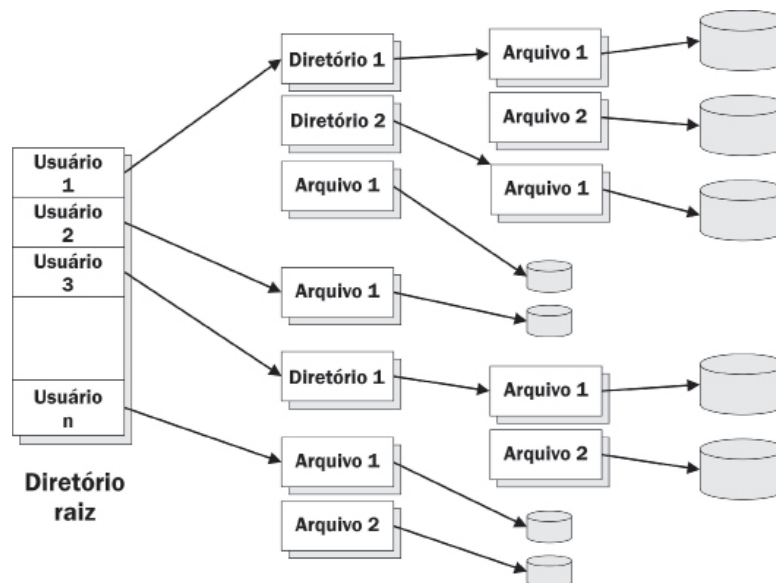




Fonte: Machado (2011)

Fig. 8.2 – Estrutura de diretórios em dois níveis

- **Estrutura de diretórios em árvore** – segundo Toscani os usuários podem criar seus próprios sub-diretórios para organizar seus arquivos (Fig. 8.3). A árvore possui um diretório raiz. Cada arquivo possui um *caminho* (nome do arquivo composto pelos nomes de diretórios que formam o caminho da raiz até ele) Disponível em <http://www.inf.pucrs.br/~ramos/sop_sistarquivos.ppt> Acesso em 22 nov.2013.



Fonte: Machado (2011)

Fig. 8.3 – Estrutura de diretórios em árvore



Um problema que o sistema operacional precisa resolver é quando há exclusão de arquivos ou diretórios que provocam a criação de “buracos” na memória secundária. É preciso reutilizar, sempre que possível, o espaço de arquivos excluídos para novos arquivos. Esse processo é chamado *alocação de arquivos*. Uma técnica usada para eliminar os “buracos” no disco rígido é a desfragmentação, um utilitário de sistema presente em todos os sistemas operacionais, que reagrupa os arquivos e diretórios de forma contígua.

Tempo! Vamos fazer uma pausa para você refletir e reorganizar as ideias. Faça a atividade para fixar os conceitos.



Acesse o site abaixo para conhecer mais detalhes sobre as principais técnicas de alocação de arquivos.

http://www.dcc.ufrj.br/~valeriab/SO-unid04_arquivos_2012.pdf

Atividade 8.3

Qual a vantagem de usar uma estrutura de diretórios em árvore?

Chegamos ao final de mais uma aula. Você vai agora refrescar sua memória lendo o resumo a seguir.

Resumo

Nesta aula, você conheceu o sistema de arquivos de um sistema operacional. Aprendeu o conceito de arquivo e como ele é armazenado em uma memória de massa. Estudou os métodos de acesso aos blocos dos arquivos e sabe que eles dependem do tipo de dispositivo de armazenamento. Pôde conhecer também os diretórios e suas estruturas básicas e como distinguir os diversos métodos de alocação de arquivos de um sistema operacional.

Atividades de aprendizagem

1. Qual o tipo de acesso em um dispositivo de armazenamento com fita magnética?
2. Apresente uma solução para o problema da criação de “buracos” na memória secundária, resultado da exclusão e gravação de arquivos.

Prezado(a) estudante,

Chegamos ao fim da nossa disciplina. Espero que tenha aproveitado ao máximo o conteúdo apresentado. Tenho, ainda, algumas palavras para você...





Palavras Finais

Prezado(a) estudante,

Parabéns pela sua dedicação e esforço para chegar com sucesso ao final desta disciplina. Na realidade, este final significa um início de uma nova fase na sua vida, pois com os conhecimentos adquiridos aqui, complementados pelas atividades da tutoria, você vai olhar para um computador com uma nova visão. Uma visão profissional, crítica e, sobretudo, investigativa, pois você estará sempre pronto a assimilar novos conhecimentos e vivenciar novas experiências. A partir desse ponto você deve se perguntar: Que caminhos devo seguir? Vou trabalhar com os sistemas operacionais tradicionais? Qual classe de clientes vou atender? Usuários domésticos? Autônomos? Ou empresas? Comece esses questionamentos agora, que ao longo da sua trajetória no curso as respostas virão naturalmente.

Desejo a você sucesso!





Guia de Soluções

Aula 1

Atividade 1.1

O sistema operacional proporciona uma interface conveniente para o usuário e, ao mesmo tempo, gerencia os dispositivos de hardware evitando falhas e controlando o uso dos recursos da máquina.

Atividade 1.2

O hardware fornece os recursos necessários de computação, como: processador, memória, e dispositivos de entrada e saída. O software são os aplicativos que utilizam os recursos do hardware e o sistema operacional os gerenciam.

Atividades de Aprendizagem

1. Seu comentário pode ser diferente. O sistema computacional terá baixo desempenho porque o hardware ultrapassado não consegue atender aos recursos exigidos pelo sistema operacional de última geração.

2. Sua tabela pode ser diferente.

Tipos de Software	Características	Exemplos
Sistema	São programas responsáveis por controlar o hardware	Windows 8, Ubuntu 10.4 e Android
Programação	São programas para criar outros programas	Java, Pearl e C#
Aplicação	São programas do usuário	Office, Skype e Counter Strike

Aula 2

Atividade 2.1

As quatro funções do sistema operacional são:

1. fornecer ao usuário um ambiente virtual mais seguro, mais fácil de entender, operar e programar;



2. controlar e gerenciar o acesso aos recursos de hardware;
3. oferecer meios para que um programa seja carregado, executado e finalizado;
4. fornecer uma visão estruturada e organizada das informações, independente dos diferentes dispositivos de armazenamento.

Atividade 2.2

- Sua resposta pode ser diferente. As simulações numéricas com informações tridimensionais para mapeamento das correntes marinhas nas áreas de exploração de petróleo é uma aplicação da Petrobrás que roda no supercomputador Netuno. Ele reduz de algumas semanas para poucos dias o tempo de realização das simulações.

Atividades de Aprendizagem

1. Sua tabela pode ser diferente.

Tipo de SO	Numero de processadores admitidos	Número de programas executados
Monotarefa	1	1
Multitarefa	1	Vários
Multiprocessadores	Dois ou mais	Vários

2. Cada janela aberta no sistema operacional Windows 8 corresponde a um programa em execução. Portanto, quanto maior o número de programas em execução maior é o trabalho do processador, acarretando queda de desempenho.

Aula 3

Atividade 3.1

As etapas são:

1. edição do texto do programa;
2. pré-compilação;



3. compilação;

4. ligação.

Atividades de Aprendizagem

1. Os passos para execução de um programa são:

- o código do programa em linguagem de máquina é transferido do disco para a memória principal;
- o Bloco de Controle do processo com os dados do programa é carregado na memória.
- o sistema operacional transfere o controle para a Unidade Central de Processamento (UCP) que carrega o registrador Contador de Programas (PC) com o endereço de memória da primeira instrução e inicia a execução do processo.

2. A função do BCP na interrupção do processo é armazenar todas as variáveis do processo para que a próxima execução inicie exatamente onde parou.

Aula 4

Atividade 4.1

Sim, é possível. Cada processo tem um identificador único, portanto o sistema operacional pode usar o mesmo código para executar processos diferentes. Por exemplo: é possível abrir dois ou mais documentos de texto ao mesmo tempo.

Atividade 4.2

A mudança de contexto é a base da multiprogramação. Ela acontece para que haja rodízio na execução dos processos que estão na fila pronto.

Atividade 4.3

O processo que captura os dados digitados pelo usuário é um processo exe-



cutado em primeiro plano.

Atividades de Aprendizagem

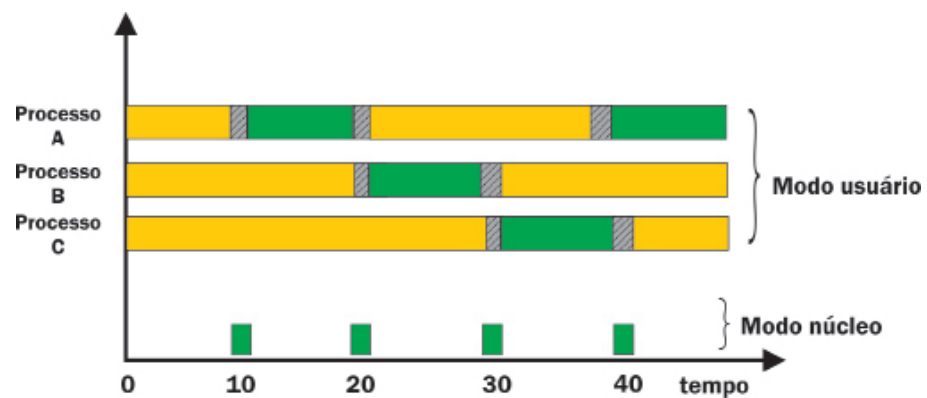
1. Em um sistema monoprogramável, o processador passa a maior parte do tempo ocioso, principalmente quando executa o processo em primeiro plano. Ao passo que em um sistema multiprogramável, o processador tem uma carga de trabalho intensa devido aos vários processos em execução.

2. O processo a ser executado primeiro será o limitado por UCP, que não depende da velocidade dos periféricos.

Aula 5

Atividade 5.1

Seu gráfico pode ser diferente.



Atividade 5.2

O mais justo é o Algoritmo por Alternância circular por dedicar o mesmo tempo de UCP para todos os processos, independentemente de tamanho ou tipo.



Atividades de Aprendizagem

1. O seu quadro pode ser diferente.

Evento	Descrição	Motivo
1	O processo sai do estado pronto e vai para o estado de execução	Início da execução
2	O processo sai do estado de execução e vai para o estado pronto	Execução interrompida
3	O processo sai do estado de espera e vai para o estado pronto	Retorno de uma requisição de E/S
4	O processo sai do estado de execução e vai para o estado espera	Inicia de uma requisição de E/S

2. Os algoritmos de escalonamento determinam a maneira como a UCP trabalha. Portanto, influenciam diretamente no desempenho do sistema operacional.

3. Comparando PEPS com MP

PEPS				MP			
Ordem	Tempo de processamento	Te	Tx	Ordem	Tempo de processamento	Te	Tx
A	4	0	4	A	4	0	4
B	32	4	36	B	7	4	11
C	7	36	43	C	25	11	36
D	25	43	68	D	32	36	68
Média		20,7	37,5	Média		12,7	29,7

Aula 6

Atividade 6.1

Sua resposta pode ser diferente.

Cabe à UGM gerar a faixa de endereços físicos usados pelos processos. Essa função permite o carregamento de vários processos simultaneamente na memória sem que haja interferência entre eles.



Atividade 6.2

O benefício da paginação de memória é que ela evita a fragmentação quando carrega um processo nos blocos disponíveis, não importando se eles estão juntos ou separados.

Atividade 6.3

O benefício da memória virtual com paginação por demanda é permitir ao sistema operacional aumentar o número de processos, em execução independente da limitação da memória física.

Atividades de Aprendizagem

1. São necessárias 30 linhas de endereço. Pois $2^{30} = 1,073,741,824$ posições de memória.
2. Sua resposta pode ser diferente.

A UGM mantém uma tabela com todas as páginas do processo. Quando o processo endereça uma determinada página, a UGM verifica na tabela se a página é válida, ou seja, se ela está associada a um quadro na memória física. Se não estiver, a UGM transfere a página do disco rígido para o primeiro quadro livre na memória física.

3. Para as páginas D, E e H, a UGM não executaria nenhuma, porque as páginas já estão carregadas na memória física. Porém, para as páginas F e G a UGM, vai transferir as páginas da memória secundária para quadros livres na memória física.

Aula 7

Atividade 7.1

Sua resposta pode ser diferente.

- Dispositivos que se comunicam por blocos: Disco Rígido, Memória USB.
- Dispositivos que se comunicam por caracteres: Impressora, mouse.



Atividade 7.2

Você deve atualizar somente o driver.

Atividades de Aprendizagem

1. A função dos controladores de dispositivos é estabelecer a comunicação do periférico com o sistema operacional.

2. Sua resposta pode ser diferente.

Quando você pressiona uma tecla, o sistema operacional inicia uma interrupção e transfere o controle da UCP para o drive do teclado para atender a solicitação. Nesse instante, o controlador do teclado envia para o driver o código correspondente à tecla acionada. O driver copia para um espaço de memória e devolve o comando do processador para o sistema operacional que direciona o código para a aplicação.

Aula 8

Atividade 8.1

Um arquivo é uma sequência de bits, bytes, linhas ou registros, cujo significado é definido por quem o criou. Um arquivo pode representar dados numéricos, alfanuméricos ou binários armazenados em memória secundária (disco rígido, SSD, DVD, pen-drive, etc.)

Atividade 8.2

Sua resposta pode ser diferente.

Uma das funções do sistema de arquivos é estabelecer para cada dispositivo de armazenamento em massa (Disco rígido, CD, DVD ou SSD) uma organização para leitura e gravação.

Atividade 8.3

Os usuários podem criar seus próprios subdiretórios para organizar seus arquivos.



Atividades de Aprendizagem

1. Acesso sequencial.
2. Usar o utilitário desfragmentador regularmente.





Referências

BAIXAKI MARANHÃO. **Android 2.2**. Disponível em: <<http://baixakimaranhao.blogspot.com.br/>> Acesso em: 22 nov. 2013.

FRACCIOLLI, P.. SIS013 - **Aula1 Subsist Em A E S Kernel**. Disponível em: < <http://pt.scribd.com/doc/68893986/SIS013-Aula1-Subsist-Em-A-E-S-Kernel>> Acesso em: 22 nov.2013.

MACHADO, Francis B. **Arquitetura de Sistemas Operacionais**. Rio de Janeiro, 4ª Ed., LTC, 2011.

OLIVEIRA, Romulo S.; Carissimi, A.; Toscani, S. **Sistemas Operacionais**. Porto Alegre, 4ª Ed., Bookman, 2010.

SILBERSCHATZ, Abraham; Galvin, Peter B.; Gagne, G. **Fundamentos de Sistemas Operacionais**. Rio de Janeiro, 6ª Ed., LTC, 2004.

_____; Galvin, Peter B.; Gagne, G. **Fundamentos de Sistemas Operacionais**. Rio de Janeiro, 8ª Ed., LTC, 2010.

SILVESTRI, Eduardo. **Sistemas Operacionais, Aula 04**. Notas de aula. Disponível em :<<http://grades.ifto.edu.br/wp-content/uploads/SO-Aula004.pdf>> Acesso em: 29 jan.2013.

TANENBAUM, Andrew S. **Sistemas operacionais, projeto e implementação**. Porto Alegre, 3 Ed., Bookman, 2008.

Toscani, S. S.. **Sistema de arquivos – ambientes operacionais**. Disponível em: <http://www.inf.pucrs.br/~ramos/sop_sistarquivos.ppt> Acesso em: 22 nov.2013.

UNIVERSIDADE DE SÃO PAULO.ICMC São Carlos. **Sistema de informação**. Disponível em: < http://www2.icmc.usp.br/~luisc/download/fundamentos/sistemas_de_arquivos.ppt> Acesso em: 8 out.2013.

YOURDICTIONARY, William Henry Gates III Biography, Disponível em: <<http://biography.yourdictionary.com/william-henry-gates-iii>> Acesso em: 26 jan.2013.



Obras Consultadas

BASTOS, Valéria M. **Sistemas Operacionais. Unidade.** IV. Sistema de Arquivos. Notas de aula, 2012, Disponível em: < http://www.dcc.ufrj.br/~valeriab/SO-unid04_arquivos_2012.pdf> Acesso em: 15 mar.2013.

Flynn, Ida M. **Introdução aos Sistemas Operacionais.** São Paulo, Editora Thomson, 2002.

GUGIK, Gabriel. **A história dos sistemas operacionais, artigo publicado na Internet, 2009.** Disponível em: <<http://www.tecmundo.com.br/linux/2031-a-historia-dos-sistemas-operacionais.htm>> Acesso em: 05 jan.2013.

TECHTUDO, **Sistemas Operacionais.** Android 2.2, Disponível em: < <http://www.techtudo.com.br/tudo-sobre/s/android.html>> Acesso em: 01 mar.2013.

Bibliografia Básica

MACHADO, Francis B. **Arquitetura de Sistemas Operacionais.** Rio de Janeiro, 4ª Ed., LTC, 2011.

TANENBAUM, Andrew S. **Sistemas operacionais, projeto e implementação.** Porto Alegre, 3 Ed., Bookman, 2008.

Currículo do Professor-autor



João Batista Pinto Neto

Graduado em Engenharia Eletrônica pela Universidade Federal do Rio de Janeiro (1979), com especialização Lato Sensu em Docência Superior também pela UFRJ (2004). Mestre em Redes de Computadores pela Universidade Federal do Amazonas (2011), onde atua como pesquisador no Instituto de Computação na área de Redes Tolerantes a Atrasos e Desconexões (DTN). Atu-

almente, trabalha como Analista de Redes de Computadores no Ministério Público do Estado de Rondônia e leciona no curso de Redes de Computadores da Faculdade Interamericana de Porto Velho – UNIRON e nos cursos de Sistemas para Internet e Sistemas de Informação da Faculdade de Ciências Administrativas e de Tecnologia – FATEC, também em Porto Velho-RO.

