

Sistemas Operacionais

Linux



Curitiba
2018

Sistemas Operacionais

O presente trabalho perfaz um requisito parcial para a aprovação na Disciplina de Sistemas Operacionais do curso de Análise de Sistemas oferecido pela Universidade Positivo - Campus Osório.

O objetivo é uma breve revisão da literatura atual sobre sistemas operacionais (SO), cabendo aos autores a seleção por uma das opções disponíveis no mercado. O estudo foi organizado na introdução sobre o assunto, o conceito de sistema operacional (SO), suas funções, comandos, bem como a abordagem específica sobre o sistema operacional Unix/Linux.

A opção de seleção do sistema operacional para esse estudo, Linux, é justificada por ser, atualmente, o sistema que mais aumenta em número de usuários no mundo, embora o Windows (Microsoft) seja o mais popular (aproximadamente 80% do mercado). Estabilidade, ótima performance, segurança e multiplicidade de recursos são os principais motivos que justificam essa preferência. Também por ser um sistema de livre distribuição.

Introdução

Um sistema de computação é constituído por hardware e software. O hardware é composto por circuitos eletrônicos (processador, memória, portas de entrada/saída) e periféricos eletro-óptico-mecânicos (teclados, mouses, discos rígidos, unidades de CD ou USB). O software de aplicação é representado por programas destinados ao usuário (editores de texto, navegadores Internet ou jogos) (MAZIERO, 2017).

O problema é que existe uma grande distância para comunicação entre o hardware e os programas aplicativos (softwares) utilizados pelo usuário, pois os circuitos eletrônicos são acessados por meio de interfaces de baixo nível (sinais elétricos interpretados como uma linguagem binária). Portanto, torna-se necessário oferecer aos programas (aplicativos do usuário) um modo de acesso homogêneo aos dispositivos físicos (hardware) que permita abstrair as diferenças tecnológicas entre eles (Figura 01).

O sistema operacional é uma camada de software que opera entre o hardware e os programas aplicativos voltados ao usuário final. Trata-se de uma estrutura de software ampla, complexa, que incorpora aspectos de baixo nível (drivers de dispositivos e gerência de memória física) e de alto nível (programas utilitários e a própria interface gráfica) (MAZIERO, 2017).

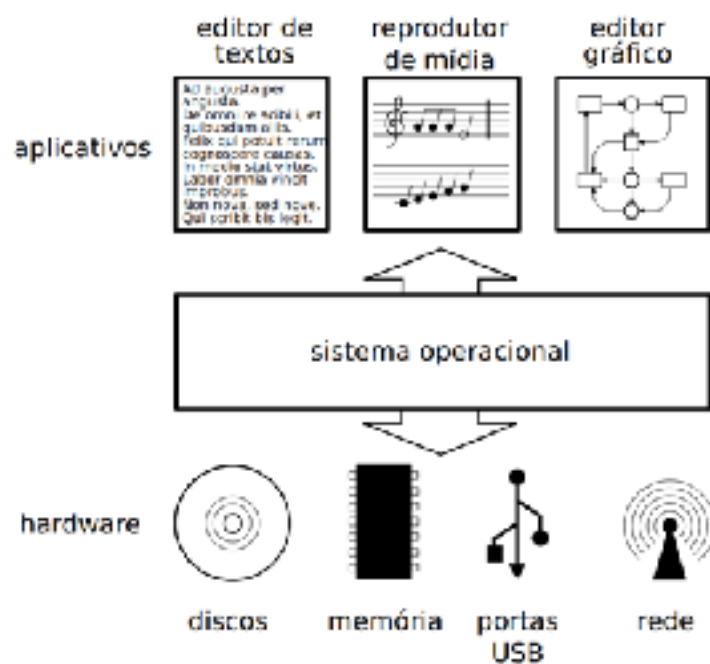


Figura 01 - Estrutura de um sistema de computação.

Sistemas operacionais são fundamentais para o funcionamento dos sistemas de computação (um telefone celular ou um complexo sistema corporativo). Apesar da grande diversidade de SOs, esses buscam resolver problemas de natureza similar e seguem, em termos gerais, os mesmos princípios.

A importância em se conhecer o assunto decorre do fato de que os mecanismos determinados pelo sistema operacional (SO) influem de modo direto no comportamento e o desempenho das aplicações, além de ser chave na configuração de serviços de rede e na segurança do sistema.

Classificação

De modo geral um sistema operacional é um programa que atua como uma camada entre o software (aplicativos) e o hardware do computador oferecendo uma interface prática e intuitiva para os usuários.

Em relação aos tipos, os SO podem ser classificados segundo diversos aspectos:

- **Batch** (de lote) - os sistemas operacionais mais antigos trabalhavam “por lote” (todos os programas a executar eram colocados em uma fila para a execução. O processador recebia os programas e os processava sem interagir com os usuários, com alto grau de utilização do sistema. Atualmente, este conceito se aplica a sistemas que processam

tarefas sem interação direta com os usuários (processamento de transações em bancos de dados).

- **Rede** - um sistema operacional de rede possui capacidade de oferecer às aplicações locais recursos que estejam localizados em outros computadores da rede, como arquivos e impressoras. A maioria dos sistemas operacionais atuais oferece esse tipo de funcionalidade.
- **Distribuído** - os recursos de cada máquina estão disponíveis globalmente aos usuários. Ao lançar uma aplicação, o usuário interage com sua janela, mas não sabe onde ela está executando ou armazenando seus arquivos: o sistema decide. Os SO distribuídos existem há tempos mas ainda não são uma realidade de mercado.
- **Multiusuário** - um SO multiusuário deve suportar a identificação do recurso de cada usuário dentro do sistema (arquivos, processos, áreas de memória, conexões de rede) e impor regras de controle de acesso para usuários não autorizados. Grande parte dos sistemas atuais são multiusuários.
- **Desktop** - um SO “de mesa” é voltado ao atendimento do usuário doméstico e corporativo. Suas principais características são a interface gráfica, o suporte à interatividade e a operação em rede (Windows XP, MacOS X e Linux).
- **Servidor** - permite a gestão eficiente de grandes quantidades de recursos (disco, memória, processadores), impondo prioridades e limites sobre o uso dos recursos pelos usuários e seus aplicativos. Normalmente um sistema operacional servidor também tem suporte a rede e multiusuários.
- **Embarcado** (embutido ou embedded) - é construído para operar sobre um hardware com poucos recursos de processamento, armazenamento e energia (telefones celulares, sistemas de automação industrial e controladores automotivos, equipamentos eletrônicos de uso doméstico). LynxOS, μ C/OS, Xylinx e VxWorks são exemplos de sistemas operacionais embarcados para controle e automação. Sistemas operacionais para telefones celulares inteligentes (smartphones) incluem o Symbian e o Android, entre outros.
- **Tempo real** - um SO de tempo real é necessariamente ultrarrápido. Sua característica essencial é ter um comportamento temporal previsível (o tempo de resposta deve ser conhecido). A estrutura interna de um SO de tempo real deve ser construída para minimizar esperas e latências imprevisíveis, como tempos de acesso a disco e

sincronizações excessivas. Existem duas classificações de sistemas de tempo real: 1) soft real-time systems, nos quais a perda de prazos implica na degradação do serviço prestado (Exemplo, suporte à gravação de CDs ou à reprodução de músicas). Nesses, caso o sistema atrase, pode ocorrer a perda da mídia em gravação. 2) hard real-time systems, a perda de prazos pelo sistema pode perturbar o objeto controlado, com graves consequências humanas, econômicas ou ambientais (Exemplos, controle de funcionamento de uma turbina de avião. Muitos sistemas embarcados têm características de tempo real e vice-versa.

Estrutura do SO

Um SO é composto de diversos componentes com objetivos e funcionalidades complementares (Figura 02). Os componentes mais relevantes são:

- **Núcleo** - é o cerne do sistema operacional, responsável pela gerência dos recursos do hardware usados pelas aplicações. Também implementa as principais abstrações utilizadas pelos programas aplicativos;
- **Drivers** - módulos de código específicos para acessar os dispositivos físicos. Existe um driver para cada tipo de dispositivo (discos rígidos, portas USB, placas de vídeo);
- **Código de inicialização** - a inicialização do hardware requer uma série de tarefas complexas como reconhecer dispositivos instalados. Outra função importante é carregar o núcleo do sistema operacional em memória e iniciar sua execução;
- **Programas utilitários** - programas que facilitam o uso do sistema computacional, fornecendo funcionalidades complementares ao núcleo (formatação de discos e mídias, configuração de dispositivos, manipulação de arquivos (mover, copiar, apagar), interpretador de comandos, terminal, interface gráfica, gerência de janelas).

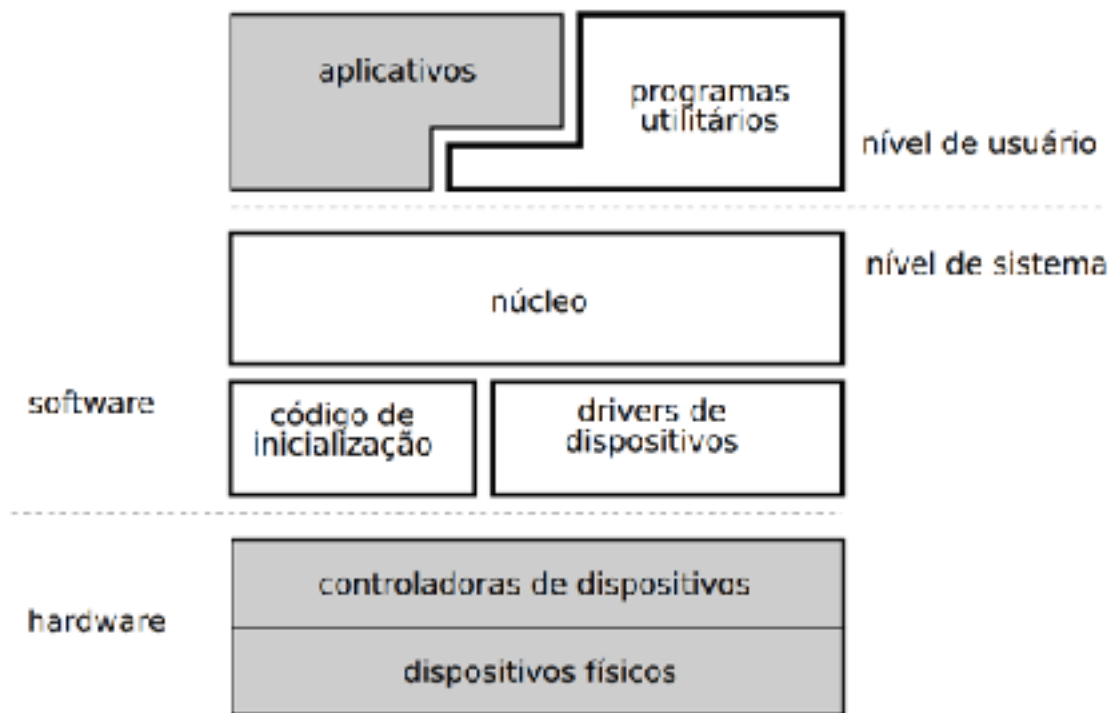


Figura 02 - Estrutura de um SO.

Funções

As funções básicas de um sistema operacional podem ser sintetizadas em duas palavras-chave: “abstração” e “gerência”.

- Abstração - diz respeito ao acesso dos recursos de hardware por meio de interfaces abstratas definidas pelo SO, proporcionando uma interface de alto nível para o usuário.
- Gerencia - refere-se às regras de uso dos recursos de hardware pelos aplicativos, solução de eventuais disputas e conflitos, haja vista que os computadores possuem menos processadores do que o número de processos em execução (o mesmo ocorre com a memória RAM).

Para cumprir os objetivos de abstração e gerência o SO deve desempenhar várias funções:

- Gerenciamento do processador (a preempção interrompe a execução de uma tarefa, oferecendo preferência à outra, para voltar depois à sua execução);

- Gerenciamento de memória (fornece a cada aplicação uma área de memória própria) (Figura 03);
- Gerenciamento de arquivos (cria arquivos e diretórios, definindo suas regras de uso);
- Gerenciamento de dispositivos (I/O controller gerencia periféricos por meio de drivers);
- Controle de tempo (cada processo recebe um quantum de tempo para uso da CPU, de modo a solucionar problemas de execução infinita);
- Controle de desempenho do sistema (escalonamento de processos);
- Interação com o usuário (interface);
- Coordenação entre aplicativos e o usuário;
- Segurança (controle de acesso por meio de login e senha);
- Tratamento e recuperação de erros.

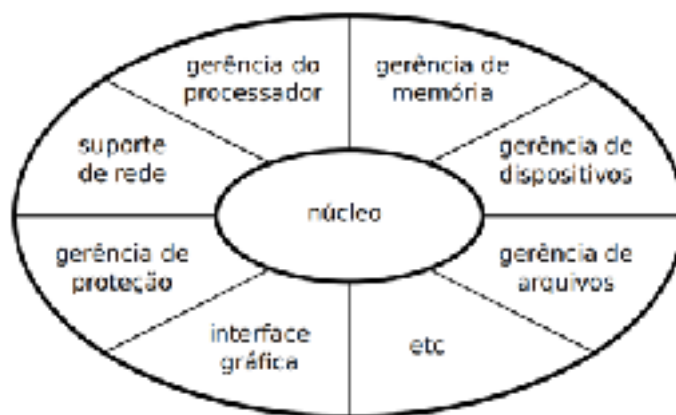


Figura 03 - Funcionalidades do SO (Fonte: MAZIERO, 2017).

Sistema GNU/Linux

GNU/Linux é um sistema operacional livre baseado no Unix. O GNU/Linux é multitarefa (permite ao processador executar várias tarefas simultaneamente), multiusuário (permite acesso simultâneo ao computador de vários usuários) e multiprocessado (capacidade do

SO executar simultaneamente 2 ou mais processos). Seu código fonte possui licença GPL (*General Public License*), dando liberdade para que qualquer pessoa possa baixá-lo e customizá-lo. Por este motivo, grandes corporações como Amazon e Google utilizam essa estrutura para os seus sistemas operacionais. Tanto o Android (SO mais popular no segmento mobile), quanto o Chrome OS são desenvolvidos com base nessa tecnologia.

Para os usuários domésticos o GNU/Linux pode não ser o melhor SO, por não executar algumas aplicações populares como o iTunes ou Photoshop. Assim como o macOS, sua compatibilidade com hardwares periféricos também é limitada. Sua especificação é para prover navegação na internet e trabalhar com edição de textos.

Embora o termo Sistema Operacional Linux tenha se tornado bastante difundido, a rigor, **Linux é um kernel** (USP, sd).

Definição

- Sistema Operacional - software que serve de interface entre o computador e o usuário, gerenciando recursos (memória, processamento, arquivos, periféricos).
- Kernel - é o núcleo do sistema operacional (parte mais baixa do SO).

Quando um usuário instala o "Linux", está instalando o kernel e mais uma série de softwares (shell, utilitários, bibliotecas, compiladores e ferramentas). Grande parte desses aplicativos pertence a um projeto chamado GNU. Logo, o SO é formado pelo kernel mais os utilitários do GNU.

No ano de 1984 Richard Stallman iniciou o Projeto GNU com o objetivo criar um SO livre. Esse sistema operacional deveria ser compatível com outro sistema operacional - o UNIX (daí o nome GNU - *GNU is Not Unix*). (USP, sd).

Unix

O Unix é um sistema operacional multitarefa (executa várias tarefas ao mesmo tempo) multiusuário (onde várias pessoas usam ao mesmo tempo). A maioria dos servidores na Internet utilizam esse sistema. O unix, como os "unix like" é montado basicamente em três níveis:

- *Kernel*. - é o núcleo do sistema, invisível ao usuário, responsável pelas funções internas do sistema;
- *Shell* - é a interface entre sistema operacional, o usuário e o núcleo do sistema(Kernel). O primeiro processo, executado automaticamente ao entrar no sistema (login) é o shell;
- *Aplicativos* - nessa classe se encaixam a maioria dos comandos do unix

Por volta de 1991, o sistema GNU estava quase pronto, exceto pelo kernel. Ao mesmo tempo, o finlandês Linus Torvalds havia criado um kernel compatível com as aplicações do projeto GNU. A esse kernel foi dado o nome de Linux. (USP, sd).

Linus Torvalds

Em agosto de 1991, na Finlândia, Linus Torvalds, um jovem estudante de 21 anos de idade, da Universidade de Helsinki, iniciou o projeto do Linux. A intenção era desenvolver uma versão do Minix (um sistema baseado em Unix) que rodasse em micros PC AT e compatíveis, mas diferente dos sistemas Unix existentes, cujo preço era exorbitante. Linus programou a primeira versão do kernel do Linux. A nova criação foi batizada com o nome de Linux, vocábulo que resultou da fusão de Unix com o primeiro nome de seu criador, Linus. Depois de finalizar o kernel, Linus passou a distribuir o código fonte do kernel pela Internet, para que outros programadores pudessem aprimorar o sistema. Esse foi o segredo, trabalho cooperativo e voluntário.

Atualmente, Linux tornou-se um termo genérico para se referir a sistemas operacionais “Unix-like” baseados no kernel Linux; entretanto, Richard Stallman sugere aos usuários que se referiam ao sistema completo como GNU/Linux.

Portabilidade

Embora não fosse um objetivo inicial (Linus Torvalds desejava um sistema operacional portátil no sentido de compilar de modo fácil aplicativos de várias de fontes), o Linux funciona com eficiência em plataformas como x64 da Intel (EM64T e

AMD64) PowerPC, Alpha, SPARC, além de ser facilmente instalado nos sistemas embarcados como PVR, celulares, TV's e Handhelds. Suporta também as extensões RRIP e ZISOFS, Reizer4, ReiserFS, Ext2, Ext3, Ext4, NFS, SWAP, UnionFS, SquashFS e Tmpfs.

Com o desenvolvimento da internet, esse SO tornou-se bastante popular, em particular no segmento de servidor e desktop, para web e banco de dados. Atualmente o GNU/Linux está entre os sistemas mais portados, rodando desde computadores portáteis (iPaq) até mainframes (IBM S/390).

Distribuição

Quando Linus Torvalds passou a disponibilizar o Linux, era apenas o Kernel com alguns comandos básicos, de modo que o usuário devia obter os demais programas, compilá-los e configurá-los. Para evitar esse trabalho, surgiu a MCC (*Manchester Computer Center*), a primeira distribuição Linux, feita pela Universidade de Manchester, disponibilizando programas pré-compilados para o usuário apenas instalar.

Distribuições GNU/Linux (distrôs) são conjuntos do sistema operacional Unix-like incluindo o kernel Linux e os outros softwares de aplicação (GNU), mantidas por organizações comerciais, como a Red Hat, Ubuntu, SUSE e Mandriva, bem como projetos comunitários como Debian e Gentoo montam e testam seus conjuntos de software antes de disponibilizá-los ao público.

É possível obter o GNU/Linux de diversas origens, porém a recomendação é evitar versões antigas para não enfrentar problemas já resolvidos. Os principais distribuidores lançam atualizações entre 4 e 6 meses.

O modo mais fácil de obter uma distribuição sem custo é por meio de download de imagens ISO, que são arquivos, em geral, por volta de 650MB cada um. O site linuxiso.org aponta links para imagens linux de distribuições do mundo todo.

Características do Sistema

- **Startup e shutdown** - correspondem respectivamente aos procedimentos de ligar e desligar o computador. O primeiro diz respeito ao fornecimento de energia para o funcionamento dos circuitos eletrônicos e não requer grandes precauções. O segundo

(shutdown) requer cuidado, uma vez que o desligamento inadequado pode causar danos ao SO destruindo as tabelas internas que ele necessita para funcionar. Quando o computador "trava", portanto, desligá-lo deve ser a última escolha. Uma opção é iniciar outra sessão, verificar o problema e encerrar a sessão travada com o comando apropriado.

- **LogIn e LogOut** - no início da sessão o nome e senha do usuário são lidos e verificados pelo programa login. O programa login criptografa a senha e a compara com a senha do arquivo de senhas para dar permissão ao usuário. Esse arquivo com as informações dos usuários normalmente é encontrado no arquivo: `/etc/passwd`. As senhas criptografadas no arquivo `/etc/shadow`. O logout da sessão é realizada pelo comando "exit".
- **Segurança** - o princípio da segurança do sistema de arquivo UNIX está baseado em usuário (dono do arquivo), grupos (reunião de vários usuários) e outros usuários (categoria de usuários que não correspondem às duas primeiras). As permissões de acesso para donos, grupos e outros usuários são independentes, permitindo níveis de acessos diferenciados. A conta root (utilizada para administração do sistema) é também chamada de super usuário (login sem restrições de segurança).
- **Consoles virtuais** - terminal (console) é o conjunto de teclado e tela conectados no computador. O GNU/Linux faz uso da característica multiusuário usando "terminais virtuais". Trata-se de uma segunda sessão de trabalho independente de outras que pode ser acessada no computador local ou remotamente. No GNU/Linux, em modo texto, é possível acessar outros terminais virtuais segurando a tecla `<ALTY>` e pressionando `<F1>` a `<F6>`.
- **Sistema de arquivos** - arquivos são centrais para o UNIX de uma maneira distinta de outros SOs. Isto porque os comandos são arquivos executáveis localizados na árvore de diretórios. Logo, privilégios do sistema, permissões e a comunicação entre processos são controlados, em grande parte por meio de arquivos.
- **Nomeclatura de arquivos** - o GNU/Linux é *case sensitive* (diferencia letras maiúsculas e minúsculas nos nomes de arquivos). Distinto de outros sistemas, os nomes dos arquivos podem ser formados por várias extensões, como: `lista.ord.txt`, `nomes.maius.classe`, `livros.meu.ord.txt`.
- **Árvore e diretórios** - o UNIX tem uma organização de diretórios hierárquica em formato *filesystem* (técnica pela qual os arquivos são nomeados e posicionados de modo lógico

para armazenamento e recuperação). A base desta árvore é um diretório chamado ``root directory``. Todo espaço em disco é combinado em uma única árvore de diretório abaixo do `/`. Três tipos de arquivos são reconhecidos pelo sistema. 1) Arquivos comuns (usados para armazenar dados do usuário e programas executáveis). 2) Arquivos de diretório (contém uma lista de arquivos com o nome do arquivo e um ponteiro para o arquivo real em disco). 3) Arquivos especiais (usados para fazer referência a dispositivos físicos como os terminais, impressoras e discos).

- **Permissões de acesso aos arquivos** - para controle de acesso aos três tipos de arquivos, o sistema identifica três tipos de usuários. 1) Proprietário (designado por "u", de *user*. é quem criou o arquivo e tem controle total sobre a restrição ou permissão de acesso e esse). 2) Grupo (designado por "g", de *group*) é formado por um ou mais usuários que podem ter acesso ao arquivo. 3) Outros (designado por "o", de *others*, refere-se a qualquer outro usuário do sistema).
- **Modos de acesso aos arquivos** - existem três modos de acesso aos arquivos: leitura, escrita e execução. Um usuário deve ter permissão de execução em um diretório para ter acesso aos arquivos ali alocados. Se um usuário tem permissão para leitura e escrita em um arquivo mas não tem permissão de execução para aquele diretório, o sistema não o deixa ler nem gravar o conteúdo daquele arquivo.
- **Processos** - é um programa rodando em seu espaço de endereçamento virtual. Comandos simples como ``ls`` são executados como processos. Para acompanhar o status de cada processo o sistema cria um PID (Process ID) para cada processo aberto.

Principais Comandos do Terminal Linux

O modo texto do Linux, utilizando o terminal, permite maior agilidade no uso do sistema. É possível usar a maior parte dos comandos GNU/Linux apresentados em outros SO como o Windows 10, o Mac OS X e os sistemas BSD.

- **ssh** - permite logar o usuário num servidor de protocolo ssh (conexão segura).
- **exit** - encerra o Shell (como o "esc"), para que outro usuário não entre no sistema.
- **apropos** - mostra todos os comando e teclas com a devida descrição.
- **cal** - mostra o calendário do mês corrente.
- **cd** - acessar e mudar de diretório corrente.
- **cp** - copia o arquivo (como o CTRL+C).
- **date** - mostra na tela a data e horário atualizados.

- **ls** - lista arquivos que estão na pasta corrente.
- **mkdir** - cria diretórios.
- **mv** - move arquivos de um local para outro. Também utilizado para renomear arquivos.
- **pwd** - mostra a pasta atual que o usuário está no momento
- **rm** - remove arquivos (irreversível).
- **rmdir** - remove diretórios vazios.
- **cat** - mostra o que tem dentro do arquivo.
- **file** - informa qual é o tipo de arquivo digitado como parâmetro (texto, imagem).
- **clear** - limpa a tela do terminal (como CTRL+L).
- **man** - mostra função de determinado comando.
- **who** - mostra os usuários logados no sistema naquele momento.
- **wc** - mostra a quantidade de linhas, palavras e caracteres de um arquivo.
- **whoami** - mostra o nome do usuário que está logado no sistema.

Existem outros comandos mais avançados que são exigidos nas provas para a certificação LPI (Linux Professional Institute).

Linux e Windows: são compatíveis?

Considerando que as duas principais famílias de sistemas operacionais atuais são o GNU/Linux e Windows, a questão que segue é a compatibilidade entre esses.

GNU/Linux e Windows podem ser instalados no mesmo computador, desde que em partições diferentes. O Windows não "enxergará" nenhuma das partições do Linux, porém o Linux "enxergará" todas as partições do Windows, inclusive com permissão de escrita.

Não é possível executar diretamente em GNU/Linux uma aplicação escrita para um sistema Windows, pois as chamadas de sistema emitidas pelo programa Windows não serão compreendidas pelo sistema SO GNU/Linux. Todavia, é possível contornar esses problemas de compatibilidade por meio de uma camada de virtualização construída em software. Essa camada de software permitirá o acoplamento entre interfaces distintas, de modo que o programa desenvolvido para a plataforma A possa executar sobre a plataforma distinta B.

Referências Bibliográficas

MAZIERO, C. A. Sistemas Operacionais: Conceitos e Mecanismos. Disponível em: <http://wiki.inf.ufpr.br/maziero/lib/exe/fetch.php?media=so:so-livro.pdf>. Acesso em: 26/10/2018.

USP. Curso de Linux - Módulo Básico. Disponível em: <https://linux.ime.usp.br/arquivos/apostila.pdf>. Acesso em 26/10/2018.

DEV MEDIA. Comandos Importantes Linux. Disponível em: <https://www.devmedia.com.br/comandos-importantes-linux/23893>. Acesso em 27/10/2018.