

PUCPR – Pontifícia Universidade Católica do Paraná
Bacharelado em Ciência da Computação

Andressa de Oliveira Barros

Fernanda Costa Moraes

Lissa Deguti

Sistemas Operacionais

Fedora Linux

Curitiba

2024

Andressa de Oliveira Barros

Fernanda Costa Moraes

Lissa Deguti

Sistemas Operacionais: Fedora Linux

Atividade apresentada como parte das exigências de obtenção de nota do curso de (Fundamentos de Sistemas Ciberfísicos) da Pontifícia Universidade Católica do Paraná, sob orientação do Prof. Andrey Cabral Meira

Curitiba

2024

Sumário

1. Introdução	1
2. Sistemas Operacionais.....	2
2. 1. Abstração de recursos.....	2
2. 2. Gerência de recursos.....	3
2. 3. Funcionalidades.....	4
2. 4. Categorias.....	5
2. 5. Interrupções e exceções.....	6
3. Sistema Operacional Linux: Fedora.....	9
3. 1. O que é?.....	9
3. 2. Qual sua utilidade?.....	10
3.3. Quais sistemas de arquivos utiliza?.....	10
3. 4. Requisitos mínimos para funcionar.....	11
3.5. Pontos fortes e fracos.....	11
3.6. Curiosidades.....	12
4. Referências bibliográficas.....	13

1. Introdução

A complexidade e a variedade dos dispositivos físicos se diferem dos softwares que operam neles. Essa diversidade pode ser uma fonte de complexidade para os desenvolvedores de software, o que exige uma abordagem uniforme para lidar com ela. Isso envolve a criação de uma interface padronizada que abstrai as peculiaridades individuais dos dispositivos físicos, o que permite que aplicativos interajam mesmo com diferenças tecnológicas.

Para lidar com essas complexidades e diversidades, se faz necessário oferecer aos aplicativos e programas de forma homogênea um acesso aos dispositivos físicos, o que significa a necessidade de abstrair a complexidade e suas diferenças entre os dispositivos, permitindo a interação mais consistente. Essa interface se chama sistema operacional.

“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, muitas vezes complexa, que incorpora aspectos de baixo nível (como drivers de dispositivos e gerência de memória física) e de alto nível (como programas utilitários e a própria interface gráfica).” (MAZIERO, cap. 1, pág. 16, 2019).

2. Sistemas Operacionais

2.1. Abstração de recursos

A abstração de recursos é um dos conceitos fundamentais dos sistemas operacionais modernos. Isso se refere à capacidade do sistema operacional de fornecer recursos de hardware, como CPU, memória e dispositivos de entrada/saída. Crie programas de software de forma simplificada e uniforme. Essa abstração permite escrever programas sem levar em conta os detalhes específicos do hardware subjacente.

O acesso aos recursos de hardware pode ser um trabalho complexo, por causa de suas características difundidas.

- “1. verificar se os parâmetros informados estão corretos (nome do arquivo, identificador do disco, buffer de leitura, etc.);
 2. verificar se o disco está disponível;
 3. ligar o motor do disco e aguardar atingir a velocidade de rotação correta;
 4. posicionar a cabeça de leitura sobre a trilha onde está a tabela de diretório;
 5. ler a tabela de diretório e localizar o arquivo ou subdiretório desejado;
 6. mover a cabeça de leitura para a posição do bloco inicial do arquivo;
 7. ler o bloco inicial do arquivo e depositá-lo em um buffer de memória.
- “(MAZIERO, cap. 1, pág. 16, 2019)

Dessa forma o sistema operacional deve definir interfaces abstratas para recursos de hardware projetados para atender aos seguintes objetivos:

interfaces mais simples para o acesso aos dispositivos, como por exemplo ao acessar um disco rígido, não haver a necessidade de usar a função read e close.

Tornar independentes do hardware, ou seja, a criação de uma interface abstrata de acesso ao hardware. Isso significa que os aplicativos podem funcionar de forma independente do tipo de hardware, permitindo a evolução autônoma, tanto do aplicativo quanto do hardware.

Fornecer uma maneira consistente de acessar diferentes dispositivos, mesmo que os dispositivos sejam muito diferentes, o sistema oferece uma interface uniforme para acessá-los, ou seja, os programas os programas podem usar os mesmos comandos para acessar diferentes dispositivos.

2.2. Gerencia de recursos

O sistema operacional, na ótica como um facilitador para os usuários, adere uma abordagem top-down, ele simplifica a complexidade por meio de uma visão abstrata e integrada. Em contrapartida uma perspectiva bottom-up, prioriza a coordenação e o controle eficiente dos vários componentes.

“Imagine o que aconteceria se três programas em execução em algum computador tentassem imprimir suas saídas simultaneamente na mesma impressora. As primeiras linhas poderiam ser do programa 1, as linhas seguintes seriam do programa 2, algumas outras do programa 3 e assim por diante. Resultado: uma confusão. O sistema operacional pode trazer ordem a essa confusão potencial, armazenando temporariamente no disco todas as saídas destinadas à impressora. Finalizando um programa e, o sistema operacional poderia então enviar sua saída, que estaria no arquivo em disco, para a impressora. Ao mesmo tempo, o outro programa poderia continuar gerando mais saídas, que não estariam, obviamente, indo para a impressora (ainda).” (TANEMBAUM, cap. 1, pág. 3, 2005)

Normalmente competem muitos processos simultâneos ao processador. Assim, o escalonamento de processos toma decisões com diversos critérios, como sua importância, tempo estimado de execução ou seu tempo de espera para ter acesso ao processador.

“Quando um computador (ou uma rede) tem múltiplos usuários, a necessidade de gerenciar e proteger a memória, dispositivos de E/S e outros recursos é muito maior, já que de outra maneira os usuários poderiam interferir uns nos outros. “(TANEMBAUM005, p. 3). Quando um processo necessita acessar os dispositivos de E/S, ele solicita ao sistema operacional, que encaminha aos drivers do dispositivo, ele interage diretamente com o hardware, que executa as operações específicas do dispositivo.

Além do gerenciamento do uso de hardware, também há a gerência de compartilhamento, sendo assim a principal função do sistema operacional nessa visão é supervisionar o acesso às informações, identificando quem está utilizando quais recursos, respondendo às solicitações de programas e usuários, administrando contas de usuários e resolvendo conflitos.

Muitas vezes, múltiplos usuários podem armazenar arquivos simultaneamente. Por sua vez, o sistema operacional tem a tarefa de alocar o espaço em disco e controlar a utilização

das porções do disco, que tem como objetivo garantir a equidade e segurança no acesso aos recursos compartilhados.

“O gerenciador de memória aloca memória para o sistema operacional e para os processos. Com o intuito de garantir que os processos não interfiram no sistema operacional ou uns nos outros, o gerenciador de memória impede que cada processador acesse memória que não lhe tenha sido alocada. Quase todos os sistemas operacionais de hoje suportam memória virtual.” (DEITEL, DEITEL, CHOFFNES, cap, 1 pág. 17, 2005)

2. 3. Funcionalidades

Segundo Maziero os sistemas operacionais devem agir em diferentes frentes para poderem realizar suas finalidades de abstração e gerência, na qual, quaisquer recursos de um sistema são necessárias exigências à gerência e abstração desses recursos. Assim, as principais funcionalidades de um sistema operacional que são implementadas pelo mesmo são a gerência do processador, de memória, de dispositivos, de arquivos e de proteção. Deve ressaltar que todas essas funcionalidades dependem uma da outra para garantir uma boa funcionalidade do sistema operacional. Então, cada uma possui sua particularidade, que são:

Gerência do processador: sua função visa garantir a separação de capacidade do processamento de forma equivalente entre as ações executadas, evitando que cada ação não se sobressaia e se comporte de acordo com o comando determinado pelo usuário. Contudo, é parte da gerência do processador prover, para tarefas individuais, abstrações para elas se comuniquem e sincronizem entre si.

Gerência de memória: Promove o uma área de memória isolada para cada aplicação, assim tendo um espaço individual e exclusivo de outras aplicações e do próprio sistema operacional, portanto, garante a segurança e estabilidade de todo o sistema e evita que aplicações mal-intencionadas interfiram em outras aplicações, e mesmo se na própria memória RAM faltar espaço para essas aplicações o sistema irá utilizar um processo chamado de memória virtual, onde o sistema operacional irá utilizar um espaço disponível por meio de uma memória de uso secundário.

Gerência de dispositivos: tem o objetivo de fazer a interação de dispositivos de entrada/ saída (*pendrives*, mouse, teclado etc), por meio de *drivers* e modelos abstratos que adotam diversos

dispositivos equivalentes, e que estes dispositivos possam ser reconhecidos pelo sistema operacional e executados corretamente.

Gerência de arquivos: esta funcionalidade tem como base de criação a gerência de dispositivos, pois através do uso de dispositivos de armazenamento a gerência de arquivos fornece uma interface funcional para o usuário que permite a organização, a criação, a modificação e exclusão de arquivos e diretórios.

Gerência de proteção: esta função tem como objetivo garantir a proteção do sistema de acessos não permitidos, pois pela presença de diversas redes e computadores, é necessário assegurar a segurança de quem pode acessar esses recursos.

“Para proteger os recursos do sistema contra acessos indevidos, é necessário: a) definir usuários e grupos de usuários; b) identificar os usuários que se conectam ao sistema, através de procedimentos de autenticação; c) definir e aplicar regras de controle de acesso aos recursos, relacionando todos os usuários, recursos e formas de acesso e aplicando essas regras através de procedimentos de autorização; e finalmente d) registrar o uso dos recursos pelos usuários, para fins de auditoria e contabilização” (MAZIERO, cap. 1, pág. 6, 2013) .

2. 4. Categorias

“Os sistemas operacionais podem ser classificados segundo diversos parâmetros e aspectos, como tamanho de código, velocidade, suporte a recursos específicos, acesso à rede etc” (MAZIERO, cap. 1, pág. 7, 2013).

De acordo com Maziero podemos citar os seguintes tipos de sistemas operacionais:

Batch (de lote): São programas executados em lote presentes em antigos sistemas operacionais, ou seja, eram executados em ordem pelo computador sem a intervenção do usuário, assim sendo mais eficiente sem precisar esperar e gastar tempo com um comando direto do usuário. Podemos ver atualmente esse tipo de sistema que executam tarefas automaticamente sem um comando do usuário.

De rede: Esse tipo de sistema permite que dispositivos de rede compartilhem recursos entre si, contudo, podendo oferecer recursos locais, como arquivos e impressoras, e conseguindo acessar de outros computadores, de forma moderada, assim garantindo que só usuários permitidos utilizem tais recursos.

Distribuído: um sistema operacional distribuído permite que os recursos no computador de cada usuário estejam compartilhados para todos na rede de forma transparente com o usuário. Dessa maneira, o sistema que irá decidir de maneira automatizada onde as aplicações feitas pelo usuário estão sendo executadas e armazenadas, assim o usuário não se preocupa onde essas aplicações estão sendo armazenadas. Um exemplo que podemos citar é o Google Drive, onde se tem um serviço de nuvem na qual se tem o manuseio e edição do arquivo que pode ser executado em diferentes dispositivos de forma transparente com o usuário.

Multiusuário: neste tipo de sistema é necessário de uma identificação de usuário para acessar cada um dos recursos, isso garante uma maior segurança e controle para que não haja acessos indesejados. Contudo, esse tipo de sistema é essencial para sistemas distribuídos e de redes.

Servidor: no sistema operacional de servidor irá permitir que o usuário utilize uma grande porção de recursos, tendo como suporte sistemas de multiusuário e rede para que possam garantir a gestão eficiente desses recursos, e nesse sistema se permite limitações e definições de preferências do manuseio dos recursos.

Desktop: é um sistema operacional que se traduz como um sistema de mesa, como próprio nome já diz é voltado para atividades usuais rotineiras para usuários domésticos e corporativos, onde se pode por exemplo reproduzir mídias, editar textos, navegar pela internet. Contudo, é uma interface visual, ou seja, o usuário consegue de forma simples com o uso de ícones dar os comandos desejados para o computador.

Móvel: é usado para dispositivos de uso móvel, como celulares e *tablets*, seus principais objetivos são garantir a duração de energia do dispositivo dure o máximo possível, assegurar

conexões diretas com outras redes, por exemplo o *wifi*, e a interação com sensores dos dispositivos, como trazer a sensibilidade com o *touch* do dispositivo.

Embarcado: construído para ser executado através de hardwares é um sistema embarcado dispositivos que utilizam pouco armazenamento, processo e bateria, como eletrodomésticos que usamos usualmente (micro-ondas, airfryer etc), na qual são implementados durante sua fabricação, dessa maneira os dispositivos já vem programados e prontos para o usuário.

Tempo real: esse tipo de sistema tem o objetivo principal que é o tempo, não precisam ser muito rápidos, só é necessário que o tempo de execução seja previsível. Existem os sistemas de tempo críticos, onde se o processamento não ocorre de acordo com o tempo determinado, gera consequências no sistema físico e no ambiente em voltas severas, e temos também os não críticos que a perda desse tempo determinado não irá gerar grandes consequências, porém será notória.

2. 5. Interrupções e exceções

De acordo com o artigo de Carlos Maziero, quando um processador precisa se comunicar com um dispositivo periférico, ele realiza a leitura ou escrita de dados nesses dispositivos por meio de acesso de portas de entrada e saída. Em casos de necessidade de notificações rápidas, como a conclusão de uma operação de disco, ou um clique de mouse, os dispositivos tem a opção de requisitar uma interrupção (IRQ, Interrupt ReQuest). Ao recebê-la, o processador ele suspende o programa em execução e redireciona o foco para uma rotina de tratamento de interrupção específica (interrupt handler). Em alguns casos, o processador gera exceções em resposta a exemplos de software ou outros erros internos.

- **Exceções:**

Ao ocorrer uma exceção, a execução normal do programa é interrompida e passa seu controle para um exception handler (controlador de exceção), que já tem o código para resolver a situação.

Essas situações são as que alteram a execução normal durante um programa. São várias as causas das exceções, como erros e situações de necessidade imediata de solução.

Caso um programa solicite o sistema operacional, isso é chamado de System Call, que essencialmente é a chamada do sistema.

Alguns erros que também podem acontecer no software que causam essas exceções são:

- a) Instruções ilegais, que tentam ser executadas por um processador que não as permite;
- b) Overflow e Underflow são erros de operações aritméticas, como: valor acima ou abaixo da capacidade e divisões por 0.

Outros erros que podem acontecer, são no hardware, como quando o programa tenta acessar página(s) que não estão na memória física, ou seja, falha de página (page fault).

- **Interrupções:**

Eventos que requerem atenção imediata do processador enviam sinais pelo hardware ou software. Esses sinais são as interrupções. Elas servem para que a interação entre CPU e os dispositivos de hardware seja eficiente.

Entre as interrupções, existem as de hardware. Alguns exemplos são a interrupção de dispositivos (teclados, discos rígidos, mouses, etc) que sinalizam que dados podem ser processados ou de operações concluídas, e a interrupção de timer, onde no interno, um temporizador gera essas interrupções periodicamente para ter controle do tempo e multitarefas.

Além dessas interrupções de hardware, existem as de software, que são geralmente ocasionadas por System calls.

Interrupções são geralmente causadas por eventos externos do programa, enquanto exceções, por erros internos. Por esse motivo, exceções são mais simples de resolver, isso porque são mais previsíveis, então possuem códigos já preparados para lidar com a maioria dos erros. Já as interrupções, são menos previsíveis e consequentemente mais complicadas de resolvê-las.

3. SISTEMA OPERACIONAL LINUX: FEDORA

3. 1. O que é?

Segundo Petersen (2018) o Fedora Linux é uma distribuição do sistema operacional Linux mantida pelo Fedora Project. É composta inteiramente por software de código aberto e desenvolvida com contribuições de desenvolvedores de Linux, permitindo-lhes total liberdade para promover aprimoramentos e novos recursos. Essa distribuição é disponibilizada gratuitamente sob a GNU General Public License.

O Fedora é lançado aproximadamente a cada 6 meses e oferece uma plataforma estável e poderosa para uma variedade de casos de uso, desde uso pessoal até ambientes de servidor.

3. 2. Qual sua utilidade?

De acordo com PETERSEN (2018), o Fedora Linux é útil como uma plataforma de código aberto, na qual oferece uma variedade de aplicativos de software para administração, servidores e uso desktop. Ele inclui uma ampla gama de software, desde servidores de rede e programas de segurança até aplicativos de escritório e ferramentas de desenvolvimento. Além disso, o Fedora é acompanhado por documentação detalhada, como guias de instalação e guias de usuário de desktop, cobrindo uma variedade de tópicos específicos.

Além disso, é popular entre desenvolvedores, sysadmins e entusiastas de tecnologia que apreciam sua abordagem de estar na vanguarda da inovação em software livre.

PETERSEN (2018) afirma que o objetivo do Fedora é ter um rápido progresso do software e conteúdos livres. Fóruns Públicos. Processos abertos. Inovação rápida. Transparência. Tudo com o objetivo de alcançar o melhor sistema operacional e plataforma que o software livre pode prover.

3. 3. Quais sistemas de arquivos ele utiliza?

Segundo Ball e Duff em seu livro traduzido por Tortello, em um sistema operacional podemos separar os sistemas de arquivos em dois tipos, o de rede e disco. No sistema operacional Fedora o usuário tem todos os aplicativos para utilizar ambos os sistemas de arquivos. Onde o sistema de arquivo de rede irá armazenar seus arquivos em algum ambiente sem estar no seu computador, mas se tendo a impressão de que está no computador do usuário, podemos citar exemplos de tipos de sistemas de arquivos de rede que são utilizados pelo Fedora:

- Network File System - com a sigla NFS é um sistema de arquivo que foi projetado para ser executado em redes seguras, então não tem um sistema de segurança interna, porém tem uma fácil implementação.
- Coda - tem um suporte de operação offline e com um cache de segurança, assim mantém os arquivos caso tenha uma perda de conexão com a rede local.

Todavia o sistema de arquivo de disco o arquivo irá ser encontrado de forma física, ou seja, no componente disco rígido do computador, temos alguns exemplos desse tipo de sistema de arquivo que são usados no Fedora Linux:

- JFS (Journaled File System), da IBM;
- XFS, da Silicon Graphics;
- Ext4, ext3 e ext2.

3. 4. Requisitos mínimos e recomendados (se houverem) para funcionar.

Segundo Ball e Duff em seu livro traduzido por Tortello, o sistema operacional Fedora pode ser baixado pela internet e gravar os CD's. O primeiro passo para instalação do sistema operacional é entender basicamente como o *hardware* do seu computador funciona, logo após é necessário escolher um tipo de instalação, o Fedora oferece vários tipos, como para o uso pessoal, trabalho ou servidor. Em seguida, é necessário que o usuário escolha uma opção de instalação de *software*. Contudo, o Fedora Linux irá funcionar em diversos tipos de hardware com processadores baseados em Intel.

“A Red Hat, Inc. publica os requisitos de hardware mínimos gerais para a instalação e uso de sua distribuição-base. Para a versão atual, seu PC deve ter pelo menos um processador Pentium de 200 MHz, 650 MB de espaço na unidade de disco rígido e 64 MB de memória RAM para uso (e instalação) do Linux, sem uma interface gráfica” (BALL, DUFF, pág. 19, 2004).

3. 5. Pontos fortes e fracos.

Pontos Fortes do Fedora Linux:

1. Inovações e atualizações:

- Frequentemente incorpora novas tecnologias antes de outros sistemas operacionais. Ou seja, propício para usuários com interesse em testar novidades tecnológicas.
- Software atualizado a cada 6 meses para manutenção de segurança.

2. Segurança:

- Security-Enhanced Linux (SELinux) é um sistema de segurança. Foi desenvolvido pela segurança nacional dos EUA, com intuito de melhorar a segurança Linux por meio de um maior controle dos processos e recursos. Entre as características do SELinux, está o controle granular, ou seja, um controle de definição do que cada processo pode realizar ou não. Além disso, esse sistema de segurança possibilita o isolamento de serviços, o que reduz o risco de caso haja uma vulnerabilidade, isso não afete o sistema por completo.
- O Fedora Linux rapidamente corrige vulnerabilidades encontradas. Isso porque atualizações frequentes permitem a solução rápida.

3. Software (Desenvolvimento):

- O sistema operacional Fedora abrange várias ferramentas de auxílio para os desenvolvedores. Além disso, oferece suporte para várias linguagens de programação: C, C++, Java, Python, JavaScript, Ruby, PHP, Lua, entre outras. Isso tudo o torna um ambiente altamente adaptável e preparado para atuação em projetos de desenvolvimento.
- Uma das edições especiais do Fedora Linux, Fedora Silverblue, foi projetada com intuito de fornecer um ambiente ideal para desenvolvimento de contêineres. Isso oferece várias vantagens para desenvolvimento de sistemas.

4. Comunidade:

- Possui uma comunidade ativa, composta de desenvolvedores do próprio Fedora, desenvolvedores em geral, e de usuários. Isso contribui com a melhoria constante desse sistema operacional.
- Dentro dessa comunidade, existem diversas documentações com resoluções de problemas e tutoriais que fortalecem a utilização do Fedora Linux.

5. Versatilidade:

- Fedora Linux possui várias edições (como Silverblue, IoT, Workstation e Server) que atendem diferentes necessidades e são suportadas em diversos tipos de dispositivos.

Pontos Fracos do Fedora Linux:

- 1. Manutenção:** Por possuir atualizações frequentes, isso pode se tornar um incômodo para usuários do Fedora.
- 2. Instabilidade:** Como a filosofia do Fedora inclui rapidamente novidades tecnológicas, é possível ocorrer erros de estabilidade. Isso não é ideal para a segurança de processos críticos.
- 3. Software (Instalação):** Alguns softwares não são de fácil instalação, ou até indisponíveis para instalação no Fedora Linux. Como por exemplo o Adobe Creative Cloud que não possui uma versão para Linux. Isso acaba exigindo soluções alternativas para a instalação desses programas.

3. 6. Curiosidades

O nome "Fedora" teve sua origem a partir de uma inspiração do logotipo da Red Hat, empresa que a patrocina. Foi usado o nome do clássico tipo de chapéu (fedora) do logo. Essa curiosidade reflete a relação entre Fedora e Red Hat.

Fedora iniciou como uma ramificação do Red Hat Linux (uma das distribuições de Linux). Até que em 2003, Red Hat Enterprise Linux (RHEL) e Fedora foram divididas em duas linhas com propósitos diferentes.

Fedora Linux agora tem o foco em inovações tecnológicas e por isso é considerado um lugar de "teste" para o sistema operacional empresarial, o RHEL.

4. Referências bibliográficas

BALL, T.; DUFF, H. *Fedora Linux*. Tradução por Tortello. 1. ed. [S.l.]: [Editora], 2004.

MAZIERO, C. *Sistemas Operacionais: Conceitos e Mecanismos*. Curitiba. DINF – UFPR, 2019.

PETERSEN, Richard. Beginning Fedora Desktop: Fedora 18 Edition. 1. Apress, 2018.

PETERSEN, Richard. Fedora Core 7 & Red Hat Enterprise Linux: The Complete Reference. 4. ed. New York. 2007.

SOYINKA, Wale. Linux Administration: A Beginner's Guide. 5. McGraw Hill. 2007.