

**UNIVERSIDADE FEDERAL FLUMINENSE**  
**ANTONIO LUIZ FERREIRA**

**O KERNEL LINUX E AS DISTRIBUIÇÕES GNU/LINUX**

**Niterói**  
**2016**

**ANTONIO LUIZ FERREIRA**

**O KERNEL LINUX E AS DISTRIBUIÇÕES GNU/LINUX**

Trabalho de Conclusão de Curso submetido ao Curso de Tecnologia em Sistemas de Computação da Universidade Federal Fluminense como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Computação.

**Orientador(a):**

**Marden Braga Pasinato**

**NITERÓI**

**2016**

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da UFF

F383 Ferreira, Antonio Luiz  
O *Kernel Linux* e as distribuições *GNU/Linux* / Antonio Luiz  
Ferreira. – Niterói, RJ : [s.n.], 2016.  
47 f.

Projeto Final (Tecnólogo em Sistemas de Computação) –  
Universidade Federal Fluminense, 2016.  
Orientador: Marden Braga Pasinato.

1. Linux (Sistema operacional de computador). 2. Software livre  
3. Arquitetura de software. I. Título.

CDD 005.43

**ANTONIO LUIZ FERREIRA**

**O KERNEL LINUX E AS DISTRIBUIÇÕES GNU/LINUX**

Trabalho de Conclusão de Curso submetido ao Curso de Tecnologia em Sistemas de Computação da Universidade Federal Fluminense como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Computação.

Niterói, \_\_\_\_ de \_\_\_\_\_ de 2016.

Banca Examinadora:

---

Prof. Marden Braga Pasinato, M. Sc. – Orientador  
Universidade Federal do Rio de Janeiro - UFRJ

---

Bruno José Dembogurski, D. Sc. – Avaliador  
Universidade Federal Rural do Rio de Janeiro - UFRRJ

Dedico este trabalho a minha mãe e a minha esposa. Ambas sempre acreditaram em mim, outorgando-me motivação nos momentos em que esta se fez necessária. E ao pequeno Oscar, meu abençoado e querido filho, nascido durante a elaboração deste trabalho.

## **AGRADECIMENTOS**

A Deus, pela vida que me deu, por permitir que eu chegasse até aqui e esse trabalho fosse feito.

A meu Orientador Marden Braga Pasinato por todo auxílio prestado, dedicação para com a minha pessoa e por compartilhar conhecimentos comigo na confecção deste trabalho.

Ao meu Tio Oswaldo, por ter visto em mim um futuro que hoje é construído graças à oportunidade que ele me concedeu.

A minha Tia Joana, por toda orientação, carinho e ajuda durante minha criação.

Ao meu Sogro Marcos, por ter me ajudado como um professor, amigo e conselheiro. Acompanhado de minha esposa, me motivaram a não desistir deste curso.

Aos meus familiares, amigos e colegas de trabalho, por toda ajuda e compreensão, também contribuíram para que este trabalho se tornasse possível.

“Nem todos que sonharam conseguiram, mas pra conseguir é preciso sonhar”.

Gabriel, o Pensador

## RESUMO

O presente trabalho tem como intuito guiar e cativar interesses ao redor do *Linux* e seu funcionamento. Para tanto, conceitos básicos e orientações serão fornecidos para tratar de problemas conhecidos. Um deles é a falta de difusão de conhecimento sobre o *kernel* e os Sistemas Operacionais *Linux*. Em comparação a grande massa de usuários de computadores pessoais, apenas um nicho específico de pessoas é usuária deste sistema e apenas algumas são detentoras de clareza a respeito do que é o *Linux* e como fazer uso de uma distribuição. Outros apesar de não serem entendidos no assunto, retêm um certo interesse e é esse público que essa temática deseja atingir. Contudo, há um desencorajamento de que um Sistema *Linux* é sinônimo de dificuldade. Definitivamente, hoje temos um leque de opções e distribuições tão fáceis (ou até mais) do que outros Sistemas Operacionais, quebrando esse paradigma. Veremos ao longo do trabalho o que é o núcleo do sistema, o famoso *kernel* Linux, o que são as distribuições e os benefícios que podem ser extraídos de cada uma delas.

**Palavras-chaves:** *Linux*, *kernel*, distro e distribuição.



## ABSTRACT

The present work is intended to guide and captivate interests around Linux and its distributions. To do so, basic concepts and guidelines will be provided in order to address known issues. One of them is the lack of diffusion of knowledge about the kernel and the Linux Operating Systems. In comparison to the large mass of personal computer users, only a specific niche of people are users of this system, and only a few have a clear understanding about what Linux is and how to make use of a distribution. Others retain a certain interest and it is this audience that this work is aiming at. However, there is a disenchantment that a Linux system is synonymous of difficulty. We definitely have a range of options and distributions as easy (or even more so) as other Operational Systems today, breaking this paradigm. We will see throughout the work what is the core of the system, the famous Linux kernel, what distributions and benefits can be extracted from each of them.

**Key words:** *Linux, kernel, distro and distribution.*

## LISTA DE ILUSTRAÇÕES

Figura 1: Processo de Inicialização de Sistemas <i>Linux</i> . .....	19
Figura 2: <i>GNU/Linux</i> , a arquitetura básica do sistema. ....	19
Figura 3: <i>Comparativo entre Distribuições Linux em 2011</i> .....	19

## LISTA DE ABREVIATURAS E SIGLAS

GNU – Um acrônimo para “*GNU Não é Unix*”, do inglês: “*GNU is Not Unix*”.

DOS – *Disk Operating System* - Sistema Operacional em Disco

MS-DOS – *Microsoft Disk Operating System* - Sistema Operacional em Disco da *Microsoft*

PC's – Personal Computers – Computadores de uso Pessoais

GPL – “*GNU General Public License*” - Licença Pública Geral

GCC – Um conjunto de compiladores de linguagem de programação – do inglês *GNU Compiler Collection*

BASH – *Bourne Again SHell* – “Nasça de novo *Shell*”

CentOS – Abreviatura de “*Community ENTerprise Operating System*” - do inglês “Sistema Operacional Empresarial da Comunidade”

RPM – Outrora “*Red Hat Package Manager*” – “Gerenciador de pacotes *Red Hat*”, atualmente “*RPM Package Manager*” – “Gerenciador de Pacotes *RPM*”

YUM – *Yellowdog Update, Modified* – Em uma tradução literal: “Atualização do *Yellowdog* Modificada”

LMDE – *Linux Mint Debian Edition* – Edição do *Linux Mint* baseada no Debian

BSD – *Berkeley Software Distribution* – do original em tradução literal: “Distribuição de *Software* de Berkeley”.

RAM – do inglês *Random Access Memory* – Memória de acesso aleatório

BIOS – Sistema Básico de Entrada/Saída – do inglês um acrônimo para “*Basic Input/Output System*”

UEFI – do inglês *Unified Extensible Firmware Interface* – Interface Unificada de *Firmware* Extensível

ROM – Acrônimo para *Read-Only Memory* – Memória somente de leitura

POST – Acrônimo para *Power-On Self Test* – de forma literal seria algo como “Auto Teste ao Ligar”

HD – *Hard Disk* – Disco Rígido

SSD – *Solid-State Drive* – Unidade de Estado Sólido, tecnologia sucessora aos discos rígidos.

USB – Universal Serial Bus – Conhecidos dispositivos "*plug and play*", ou seja, do tipo "li-gue e use", um exemplo comum são dispositivos de armazenamento externo, como *pen-drives*.

CD – Do inglês *Compact Disc* – Disco compacto, uma mídia utilizada para armazenar dados.

DVD – Disco Digital de Vídeo – do inglês *Digital Video Disc* – Tecnologia que sucedeu o CD.

SO – Sistema Operacional

ID – Termo comum que na computação é referente a identificação, habitualmente utilizado na forma de um número único.

PID – ID de Processo

ISO – Extensão de arquivos contendo imagens brutas de CD

# SUMÁRIO

RESUMO .....	7
ABSTRACT (opcional) .....	8
LISTA DE ILUSTRAÇÕES .....	9
LISTA DE ABREVIATURAS E SIGLAS .....	12
1 INTRODUÇÃO .....	15
2 O QUE É <i>LINUX</i> .....	17
2.1 NASCIMENTO DO <i>LINUX</i> .....	17
2.2 DISTRIBUIÇÕES <i>LINUX</i> .....	20
2.2.1 <i>Slackware</i> .....	21
2.2.1 <i>Debian</i> .....	22
2.2.1 <i>Fedora</i> .....	23
2.2.1 <i>Ubuntu</i> .....	23
2.2.1 <i>Linux Mint</i> .....	24
2.2.1 <i>OpenSUSE</i> .....	24
2.2.1 <i>Arch Linux</i> .....	25
2.2.1 <i>Gentoo</i> .....	26
2.3 OUTROS SISTEMAS <i>LINUX</i> .....	26
3 O <i>KERNEL LINUX</i> .....	28
3.1 A ARQUITETURA DO <i>KERNEL LINUX</i> .....	29
3.2 O TIPO DE <i>KERNEL</i> DO <i>LINUX</i> .....	31
3.3 VERSÕES DO <i>KERNEL</i> .....	31
3.4 O <i>KERNEL</i> E AS DISTRIBUIÇÕES.....	33
3.5 O <i>SHELL LINUX</i> .....	34
4 QUAL DISTRIBUIÇÃO ESCOLHER? .....	35
4.1 ADQUIRINDO UMA DISTRO .....	36
4.2 INTERFACE GRÁFICA .....	36
4.2.1 Interfaces minimalistas com foco em leveza .....	38
4.2.2 Interfaces completas com foco em aparência e efeitos visuais.....	39

4.2.3 Interfaces intermediárias com foco em balancear aparência e leveza .....	39
4.3 ESTILOS DE USO EM SISTEMAS <i>LINUX</i> .....	40
5 CONCLUSÕES E TRABALHOS FUTUROS .....	42
REFERÊNCIAS BIBLIOGRÁFICAS.....	46
ANEXOS .....	22

# 1 INTRODUÇÃO

Este trabalho se dedica a trazer conhecimento sobre o *Linux*, esse importante universo que permeia a comunidade de *Software*<sup>1</sup> Livre. Apresentaremos um estudo sobre o *kernel*, relatando o seu funcionamento interno, as distribuições *GNU/Linux* e, então, orientaremos a escolha deste Sistema Operacional que contempla uma pluralidade de opções, de acordo com o contexto onde será empregado e também a adaptação pessoal do usuário.

Um Sistema *Linux* é comumente conhecido por ser gratuito, seguro, rápido e complexo. A verdade é que a sua famigerada complexidade, nos dias de hoje, foi quase que extinta, com boa parte das distros (distribuições) sendo cada vez mais amigáveis ao usuário. Customizável e altamente configurável, é uma excelente ferramenta para profissionais da área de Informática. Fatores que regem a segurança do sistema são também pontos fortes encontrados no *Linux*, tendo em vista seu uso em máquinas como servidores e outras em que a proteção contra vulnerabilidades é algo imprescindível. Essas e outras características acabam por culminar na decisão de quem o usa.

As motivações para este trabalho surgiram quando o autor teve sua primeira experiência com uma distro. O conhecimento em torno do *Linux*, ainda é pouco difundido, sendo desencorajado aos novatos. Apesar da reputação que o precede, de ser um sistema difícil, existem distribuições *Linux* que podem atender a qualquer tipo de público interessado. Buscando contribuir com um conteúdo brasileiro, este material se compromete a conceder um estímulo a quem deseja dar os primeiros passos e, sendo ainda mais pertinente, desmitificar este sistema e como as distros funcionam por dentro.

Em sequência a esta introdução, temos o capítulo 2 que aborda, de forma sucinta, o que exatamente é o *Linux*, salientando as diferenças entre o que é o *kernel Linux* e o que é um Sistema Operacional *GNU/Linux*. Uma explanação do *kernel*

---

<sup>1</sup> *Software* – De forma sucinta, pode ser entendido como um programa de computador.

é feita no capítulo 3, descrevendo suas principais características e seu funcionamento. O capítulo 4 trata de orientações a respeito das distribuições *Linux*, conhecendo as características mais marcantes que comumente levam a escolha de uma determinada distro. Finalmente, apresentamos nossas conclusões que nos levam a uma sondagem do *kernel* (elemento primordial e característico aos sistemas *Linux*) ao produto final (a distro).



## 2 O QUE É *LINUX*?

*Linux* é o nome do núcleo (*kernel*) de Sistema Operacional criado por Linus Torvalds em 1991, no entanto o termo *Linux* faz referência também aos Sistemas Operacionais que fazem uso do *kernel Linux*. Apesar de comumente fazer alusão a ambos, sistema operacional e núcleo do sistema, temos que o Sistema Operacional *Linux* (ou *GNU/Linux*) é na verdade o conjunto de *softwares* que, acompanhado do *kernel*, formam uma distribuição *Linux*. Em outras palavras, o *kernel Linux* não é um sistema operacional, já que são necessários outros *softwares* (utilitários e aplicativos), funcionando em conjunto com o núcleo num composto [1].

Tais *softwares*, que comumente fazem parte de uma distribuição *Linux*, são *softwares* livres e dentre eles estão os desenvolvidos pelo projeto *GNU*. Por isso o nome *GNU/Linux*, que compõe um extenso debate, ainda que sem um veredito, quanto ao emprego desta nomenclatura e não somente “*Linux*”, como convencionalmente é usado em referência ao Sistema Operacional. Para entender melhor esses e outros conceitos que permeiam o universo *Linux*, abordaremos um pouco da história desse sistema operacional [1].

### 2.1 NASCIMENTO DO *LINUX*

É necessário tomar conhecimento de que o *Linux* adveio do *Unix*, um sistema operacional que teve a sua história iniciada na década de 60. Devido à escassez de recursos computacionais disponíveis naquela época, a tarefa de criar um sistema operacional exigia trabalhar com poucos recursos (pouca memória e poder de processamento, máquinas que hoje são consideradas rudimentares) e aproveitá-los ao máximo (o sistema não poderia exaurir o pouco de recurso disponível). Somando essas privações e o potencial dos envolvidos no projeto, surgiu o sistema *Unix*, ali-

cerçado em conceitos minimalistas que até hoje continuam sendo empregados em outros projetos [2].

No início da década de 80, o *Unix* já havia amadurecido, até então ele era trabalhado em áreas de pesquisa e universidades que contribuíam cooperativamente para o seu desenvolvimento, contudo o sistema agora passava a ser um *software* proprietário. Versões do *Unix* eram comercializadas, de forma concorrente entre as empresas que possuíam os direitos sobre o sistema, findando assim com o seu desenvolvimento colaborativo, desmembrado em versões conflitantes [2].

A *Microsoft*<sup>2</sup> trabalhou com um sistema *Unix* chamado *Xenix*. Embora a empresa tenha pouco tempo depois perdido o interesse nele, a experiência com o *Xenix* influenciou o trabalho da *Microsoft* com o sistema *DOS* que desde sua segunda versão, possuía ideologias oriundas dos sistemas *Unix*. No ano 1981, a *Microsoft* lançou o sistema *MS-DOS* em sua primeira versão. A empresa estava ganhando mercado com os primeiros computadores de uso pessoal, o *DOS* e posteriormente o *Windows*<sup>3</sup>, tinham a sua popularidade notável entre os microcomputadores. O Sistema *Unix* sempre fora utilizado em bancos, organizações militares e em outras aplicações de porte empresarial. Não houve um investimento para estender o foco aos microcomputadores, pois os mesmos possuíam um potencial computacional fraco em comparação aos computadores de porte [2].

Um estudante finlandês da Universidade de Helsinque chamado Linus Torvalds estava trabalhando numa proposta de sistema tipo *Unix* para rodar num computador *Intel 80386*<sup>4</sup>. O *Minix*, desenvolvido por Andrew S. Tanenbaum, tratava-se de um sistema *Unix* feito para o meio acadêmico, era o que mais se aproximava do objetivo de Torvalds à época. Após dedicar um tempo de trabalho sozinho, Linus enviou uma mensagem que expunha seu trabalho num grupo de discussão, um passo importante que possibilitou o desenvolvimento conjunto do núcleo que hoje é conhecido como *kernel Linux*.

---

<sup>2</sup> *Microsoft* – Famosa empresa detentora do Sistema Operacional Windows.

<sup>3</sup> *Windows* – Principal produto da *Microsoft*, sendo o sistema operacional mais usado em *PC*'s.

<sup>4</sup> *Intel 80386* – *Intel* é uma empresa muito conhecida, produtora de elementos de *hardware* como microprocessadores, cujo o *80386* foi um dos seus primeiros processadores utilizado em computadores de uso pessoal.

O *kernel* era o elemento principal, mas não era o único necessário para um sistema operacional. Richard Stallman, cientista do Instituto de Tecnologia de Massachusetts, havia iniciado, em 1983, o projeto *GNU*, isso ocorreu antes de Linus Torvalds, em 1989, anunciar sobre o núcleo de Sistema que estava trabalhando. Assim como Stallman, outros programadores e colaboradores que haviam contribuído para o *Unix* achavam injusta a monopolização das empresas detentoras do direito de propriedade sobre o sistema.

O projeto tinha o intuito de criar um sistema *Unix* gratuito, o *GNU*. Em sequência, Stallman junto com outros envolvidos no projeto criaram a *Free Software Foundation* (Fundação *Software* Livre) em 1985. Uma fundação sem fins lucrativos e a partir dela foi criada a *GPL* com conceitos que podem ser sintetizados em: o *software* pode ser utilizado por qualquer um e para qualquer desígnio; o programa pode ser explorado e assim ter uma interpretação a respeito de seu funcionamento, bem como pode ser moldado conforme for conveniente; o *software* pode ser redistribuído sem restrições; melhorias previstas ao programa podem ser feitas e elas devem ser disponibilizadas para todos. Em virtude disto, o acesso ao código fonte é um pré-requisito a ser atendido por um *software* disponibilizado sob a licença *GPL* [2].

Analogamente a um dos órgãos principais de um corpo, o *kernel* não funciona sozinho, ele é como um coração ou um motor, precisa de outros elementos (como editores, compiladores, sistema de janelas, gerenciadores de janelas, interface gráfica, utilitários, bibliotecas, etc) trabalhando harmonicamente para se ter um sistema. Esses programas, oriundos do projeto criado por Richard Stallman, eram as ferramentas necessárias de que o núcleo *Linux* necessitava. A recíproca era verdadeira, havia a necessidade de um núcleo para completar os *softwares* do projeto *GNU*. Diferente dos seus antecessores, os sistemas *Unix* que possuíam licença de código fechado (*software* proprietário), o núcleo criado por Linus Torvalds era de código aberto, acessível para qualquer um que quisesse distribuir e contribuir com aprimoramentos e correções. Foi possível que o sistema crescesse gradativamente, sendo constantemente ampliado com melhorias. Tudo o que foi reunido ao longo dos anos, na espera de um *Unix* voltado para os microcomputadores, foi direcionado e aplicado no *Linux* [2].

## 2.2 DISTRIBUIÇÕES *LINUX*

Distribuições *Linux* são diferentes sistemas que empregam o *kernel Linux*, ou seja, são Sistemas Operacionais do tipo *Unix*, comumente utilizando também os *softwares* do projeto *GNU*, por isto o termo Distribuições *GNU/Linux*. O *kernel* somado a programas como o servidor gráfico (o mais utilizado é *X.org*, também conhecido como *X Window System*), o *GCC*<sup>5</sup>, o *Shell*<sup>6</sup>, o *bootloader*<sup>7</sup>, dentre muitos outros *softwares*, que funcionam como bibliotecas, ferramentas, utilitários, etc. [5].

Uma distribuição, por vezes chamada apenas de distro, diferencia-se das outras, conforme a escolha, combinação e utilização desses programas com o *kernel* (o núcleo também sofre variações, podendo ser melhor adaptado e customizado de acordo com seu emprego numa distribuição). A forma como as distros manipulam os programas no sistema pode ser analisada a luz de duas ferramentas principais: repositório de pacotes e gerenciamento de pacotes. Pacotes são todos os aplicativos, ou seja, todos os *softwares* que são instalados compondo o sistema. Usualmente, estes pacotes são programas que foram previamente compilados para uma posterior instalação, referenciados como arquivos binários. Entretanto, existem distribuições que compilam os pacotes durante a execução do sistema, trabalhando diretamente com arquivos de código fonte do programa. Na verdade, esta última é uma opção comum a todas as distros, uma vez que dispendo do código fonte para uma arquitetura de processador e para um sistema *linux*, basta seguir um processo de compilação e posterior instalação, obtendo o programa desejado.

Um repositório de pacotes de uma distro é o local onde esses *softwares* estão armazenados e, posteriormente, são buscados e requisitados pelos usuários para instalação. Esse processo é feito pelo gerenciador de pacotes, ferramenta nati-

---

<sup>5</sup> *GCC* – Compilador aprimorado para diversas linguagens de programação, especialmente a linguagem C.

<sup>6</sup> *Shell* – Em tradução literal significa casca, mas no *Linux* é conhecido como um interpretador de comandos, sendo o mais comum o *Bash*.

<sup>7</sup> *Bootloader* – Trata-se de um pequeno programa, como exemplo temos o *grub2*, que é um gerenciador de inicialização, responsável por carregar o sistema operacional na memória, processo conhecido como *boot*.

va nas distros. Cada distro costuma possuir um repositório e um gerenciador de pacotes próprio, salvo quando uma distro é filha de uma outra e herda seu gerenciador e/ou repositório.

Outro ponto importante na diferenciação de uma distro é a sua filosofia de uso. Se há distros com os mesmos *softwares*, o que vai diferenciá-las, além do que tange os gerenciadores e repositórios de pacotes, é a forma como esses *softwares* são empregados e utilizados. Isso significa que as distros costumam possuir um foco que determina para que tipo de utilização ela é voltada. Existem distros populares que tem sua atenção direcionada para serem amigáveis ao usuário, tornando seu uso cotidiano facilitado, com ferramentas cada vez mais poderosas. Todavia, temos distros notáveis no universo *Linux* que mantêm o conceito de simplicidade latente, ou seja, não quer dizer que o modo de utilizar o sistema seja mais fácil, nem o menos trabalhoso de se fazer, e sim o modo mais limpo, por vezes mesclando-se ao minimalismo e leveza, herança dos ideais *Unix*.

As distros são mantidas por projetos comunitários ou por organizações comerciais. Graças a maior parte, se não em sua totalidade, dos *softwares* de uma distro serem livres, uma pessoa ou organização tem a liberdade de oferecer, seja de forma comercial ou não, distribuições de sua autoria. Por conta disso, hoje temos mais de 300 distros que são mantidas ativas, em contrapartida, somente algumas dezenas de distros são popularmente conhecidas. Citaremos algumas das mais famosas em atividade, frisando que muitas distribuições são, na verdade, distros filhas de outra. Graças à licença *GPL* existe essa possibilidade: modificar as distros, seja com melhorias ou com adaptações, bem como disponibilizar tais alterações no *software*.

### 2.2.1 *Slackware*

*Slackware* é a distro mais antiga que permanece em atividade. A fidelidade aos ideais *Unix* ainda vive nessa distribuição, com conceitos mantidos como a leveza, simplicidade, estabilidade e segurança. *Slackware* é uma distro poderosa que corriqueiramente é associada à dificuldade de utilização, já que não possui muitas

ferramentas gráficas nativas comparando-se com outras distribuições populares. Inclusive seu processo de instalação permanece em modo texto. Porém, para usuários avançados e/ou para quem busca conhecimento em torno do *Linux*, essa distro funciona aos moldes mais corretos, por sempre prezar pelo jeito mais simples de lidar com o sistema (princípio K.I.S.S - “*Keep It Simple, Stupid*” – um acrônimo para “mantenha isso simples, estúpido”). O *software* que comumente gerencia seus pacotes é o *pkgtool* (sendo o *slackpkg* também utilizado oficialmente, vindo com a distro em sua instalação) que faz uso de pacotes “.txz” [3].

### 2.2.2 *Debian*

*Debian* possui um dos maiores repositórios de pacotes e faz uso de arquivos empacotados que possuem a extensão “.deb”. Seu sistema de gerenciamento de pacotes é feito com o *APT (Advanced Package Tool* – “Ferramenta de Pacote Avançada”) que é um dos mais práticos. O *Debian* é uma das distribuições mais estáveis já que seus pacotes são testados inúmeras vezes. A versão convencional do *Debian* possui o repositório de pacotes *Stable* (estável), mas temos outros dois repositórios: o *Testing* (teste) e o *Unstable* (instável). *Debian Unstable*, também chamado de *Debian Sid*<sup>8</sup>, trata-se de um repositório com os pacotes mais recentes disponíveis, porém sacrificando estabilidade já que os programas ainda estão em fase de teste. O *Debian Testing* faz uso de um repositório de pacotes mais atuais que os presentes no *Stable*, que passaram por testes um certo tempo no repositório *Sid*, mas ainda ficarão no repositório *Testing*, por um longo período, até serem considerados estáveis e finalmente passarem para o repositório *Stable* [3].

---

<sup>8</sup> *Debian Sid* – *Sid* é o nome de um vilão destruidor de brinquedos do filme infantil *Toy Store*, a versão instável sempre terá esse codinome. A versão estável e de teste recebem nomes de personagens do filme, na corrente data temos o *Debian Jessie* em sua versão estável e a próxima versão estável (atualmente versão teste) *Debian Stretch*.

### 2.2.3 Fedora

*Fedora* é uma distro filha da distribuição *Red Hat*, tendo sua versão comercial hoje chamada de *Red Hat Enterprise Linux (RHEL)*. Red Hat é também o nome da empresa mantenedora desta distribuição, ela testa seus projetos no *Fedora* antes de serem considerados estáveis para sua versão comercial. Desta forma, a distro *Fedora* tem seu foco em inovação e é sempre servida por uma tecnologia recente. Outra distribuição da família *Red Hat* é o *CentOS* que possui uma forte semelhança e compatibilidade com o *RHEL*. Todavia, o *CentOS* é uma alternativa gratuita mantida pelo *CentOS Project*, mas possuindo o mesmo padrão em suporte (que é de, no mínimo, 7 anos) e segurança da versão *Enterprise* (Empresa). Utilizam pacotes “.RPM” possuindo suporte a um vasto número deles. O *YUM* faz o gerenciamento desses pacotes, uma ferramenta poderosa para pesquisa e manuseio de *softwares* do sistema [3].

### 2.2.4 Ubuntu

*Ubuntu* é uma distribuição filha do *Debian*, sendo hoje uma das mais populares. Seu foco está em atender bem o usuário final com suas versões *desktop*<sup>9</sup>, conhecidas pelo uso descomplicado que vai desde a instalação ao dia a dia utilizando o sistema. A *Canonical* é a empresa responsável por manter a distro e a cada seis meses faz o lançamento de novas versões (em abril e em outubro). Dentre estas, a cada 2 anos, a versão lançada no mês de abril é de suporte longo (*LTS – Long Term Support*), possuindo suporte estendido por 5 anos em detrimento as outras versões com suporte de 9 meses. Apesar de ser filha do *Debian*, utilizando o mesmo sistema de gerenciamento de pacotes, o “*apt-get*” e, conseqüentemente, fazendo uso do sistema de empacotamento “.deb”, o *Ubuntu* possui ferramentas que torna-

---

<sup>9</sup> *Desktop* – Do inglês tem o significado de “Área de Trabalho”, mas pode ser uma referência também para computadores de uso pessoal, ou seja, uma espécie de sinônimo para *PC*.

ram seu uso mais facilitado que na distro pai e atualmente possui um repositório de pacotes próprio (assim como o uso de *PPA*<sup>10</sup> tornando a aquisição de pacotes ainda mais facilitada) [3].

### 2.2.5 *Linux Mint*

*Linux Mint* é filha do *Ubuntu*, tem compatibilidade e usa os mesmos repositórios da distro pai. A diferença está no seu foco em ser uma distribuição “*out of the box*”<sup>11</sup> que busca estar apta para atender o usuário sem a real necessidade de um processo de pós-instalação, aonde programas seriam adicionados para deixar o Sistema Operacional mais completo. Diante disso, esta distro possui *drivers*<sup>12</sup> e *codecs*<sup>13</sup> proprietários que já vem instalados por padrão, além de ferramentas em modo gráfico que facilitam ainda mais as tarefas do *Ubuntu*, como as que seriam feitas em modo texto. O *Linux Mint* possui uma versão alternativa chamada *LMDE* que é baseada no *Debian Stable*, unindo a estabilidade e robustez do *Debian* com a praticidade encontrada no *Linux Mint*. [3].

### 2.2.6 *openSUSE*

*OpenSUSE*, anteriormente chamada de *SUSE*, é uma distro filha do *Slackware*. Sua versão comercial é a *Suse Linux Enterprise Server (SLES)*, mantida pela empresa *SUSE* que é a patrocinadora do *OpenSUSE*. Seu diferencial e instrumento mais poderoso é o *YaST (Yet Another Setup Tool)*, um programa com funcio-

---

<sup>10</sup> *PPA* – *Personal Package Archives*, programas armazenados em servidores de uso pessoal, ou seja, pacotes que podem ser adicionados mesmo estando fora dos repositórios oficiais da distro.

<sup>11</sup> “*Out of the box*” - Termo que recebe o significado de algo que está pronto para uso assim que adquirido.

<sup>12</sup> *Driver* – *Software* responsável por fazer o controle de um tipo de *hardware*.

<sup>13</sup> *Codec* – Um dispositivo cujo o nome “CoDec” é um acrônimo para Codificador/Decodificador de sinais.



namento exemplar permitindo instalar, personalizar e configurar o sistema de forma dinâmica e facilitada. Por conta desta ferramenta, a distro é comumente utilizada de forma empresarial, mesmo em sua versão livre, já que o *YaST* facilita os processos envolvendo configurações que são bastante requeridas em servidores. Utiliza o sistema de empacotamento “.rpm”, com o gerenciamento de pacotes feito de forma gráfica com o *YaST* ou pela linha de comando com o *zypper* (gerenciador de pacotes). Possui uma versão chamada openSUSE Tumbleweed, uma distro *Rolling Release*, esse tipo de sistema é constantemente atualizado, contendo as versões estáveis de pacotes mais recentes, em contraste com o modelo mais habitual, *Fixed Release*, que observa ciclos de lançamentos em períodos mais estritos [3].

### 2.2.7 Arch Linux

*Arch Linux* é uma distribuição que mantém o sistema e seus pacotes sempre atualizados, sendo uma distro do tipo *Rolling Release*. Mais uma distribuição adepta ao princípio *K.I.S.S.*, a simplicidade, elegância e minimalismo costumam caminhar juntos no uso do *Arch Linux*. O sistema é altamente customizável, inclusive o processo de instalação da distro instala somente o básico (o necessário para o carregamento do sistema, sem interfaces gráficas ou utilitários adicionais) e deixa nas mãos do usuário o controle sobre a escolha dos demais pacotes. Seu gerenciamento de pacotes é feito pelo *Pacman* que usa pacotes “*tar.xz*”. Além dos repositórios oficiais, o *Arch Linux* conta com o repositório dos usuários, chamado AUR (*Arch User Repository*), aonde os próprios usuários podem armazenar *scripts*<sup>14</sup> para a construção de pacotes [3].

---

<sup>14</sup> *Scripts* – Aqui este termo se refere a linguagens de programação, mais precisamente linguagens de *script*. *Scripts* podem ser executados interiormente a um programa ou em conjunto com outras linguagens de programação. Um exemplo mais comum no Linux são *Shell Scripts*.

### 2.2.8 Gentoo

*Gentoo* é um sistema adaptável e veloz, diferente das distros anteriormente mencionadas, ele é conhecido por não fazer uso de aplicativos empacotados, sendo a maior parte compilada na máquina do usuário antes da instalação. Utiliza o *portage* um dos mais poderosos gerenciadores de pacotes. O manuseio da distro costuma exigir mais de quem a utiliza, portanto tendo como público-alvo desenvolvedores e usuários avançados que buscam usufruir de otimizações dada a compilação do sistema montada em seu próprio *hardware*<sup>15</sup>. É uma distro *Rolling Release*, mas também pode ser usada com pacotes mais estáveis, bem como uma periodicidade mais longa entre as atualizações.

## 2.3 OUTROS SISTEMAS LINUX

Existem outras distribuições populares e/ou importantes, muitas são filhas de uma das mencionadas ou tem um princípio de funcionamento semelhante. Temos ainda sistemas que não são categorizados como distribuições *GNU/Linux*, mas que derivam do *Unix*, portanto assemelhando-se ao *Linux*.

O *FreeBSD* utiliza um *kernel* próprio, sendo uma distro da família *BSD*, desenvolvido pela Universidade de Berkeley, mais comumente usado em servidores do que em *desktops*. Estabilidade e potência são pontos fortes nesse sistema que herda muito da filosofia *Unix*, tendo, portanto, semelhanças e diferenças para com os sistemas *GNU/Linux*, mas predominando uma forte compatibilidade com o mesmo (*FreeBSD* pode utilizar um *kernel Linux*, assim como certas distribuições também podem fazer uso de um *kernel BSD*).

---

<sup>15</sup> *Hardware* – Elementos físicos da máquina, as peças que compõe o computador.

O sistema *Android*<sup>16</sup> utiliza um *kernel Linux* que foi modificado pela Google, desenvolvedora deste sistema, e também adaptado pelos fabricantes de *smartphones*<sup>17</sup>, sendo um dos mais usados na categoria *mobile* (dispositivos móveis).

Temos ainda empresas e organizações renomadas que utilizam o *Linux*, ou outras variações/adaptações deste sistema operacional, em dispositivos como: servidores, satélites, controle de tráfego, eletroeletrônicos, armamentos tecnológicos, trens-bala, veículos autônomos, caixas eletrônicos, submarinos, dentre outros [4].

Mesmo fora do âmbito das distribuições, o *Linux* é amplamente utilizado. No capítulo 4, voltaremos a analisar as distros sobre uma nova perspectiva. Falaremos sobre elementos como Ambientes *Desktop*, Gerenciadores de Janela, dentre outras personalizações que diferenciam graficamente as distribuições das demais. Apontaremos abordagens que ajudarão na escolha das distros.

---

<sup>16</sup> *Android* – Principal sistema operacional presente em dispositivos móveis, como os celulares do tipo *smartphones*.

<sup>17</sup> *Smartphone* – Em tradução literal seria “telefone inteligente” ou “telefone esperto”. Aparelhos telefônicos dotados de um sistema operacional de forma análoga a um computador, bem como é capaz de executar funções sofisticadas extensíveis por meio de seus aplicativos que também são instalados de forma semelhante aos softwares (programas) do *PC*.

### 3 O KERNEL LINUX

O *kernel* carrega a responsabilidade em fazer a correspondência entre os elementos do *hardware* e os *softwares* do sistema. Portanto, esse núcleo tem como principal propósito gerir os recursos da máquina, inclusive extraindo o melhor proveito da memória *RAM* para manter a saúde do computador [6].

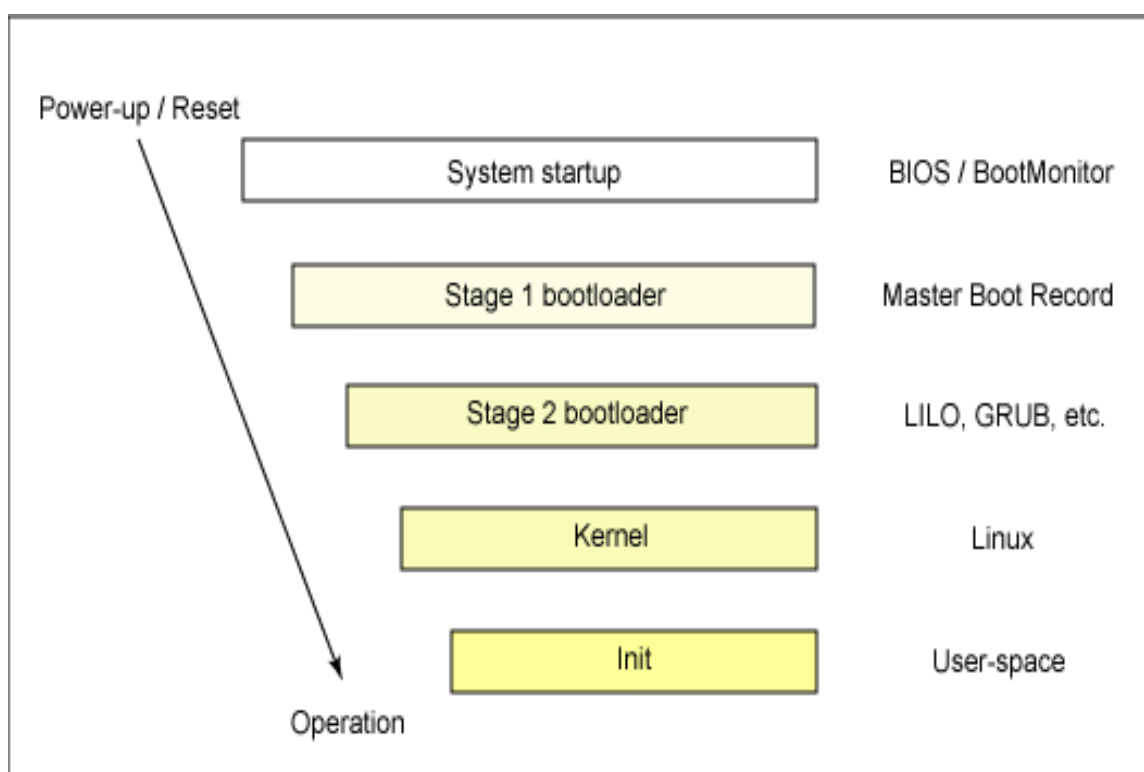


Figura 1: Processo de Inicialização de Sistemas *Linux* [8]

O funcionamento do *kernel* é dado desde o processo de inicialização do sistema, no momento posterior a leitura do *firmware* da placa-mãe (como o *BIOS* ou *UEFI* que são armazenados em uma memória não volátil, como *ROM* ou *flash*<sup>18</sup>) as-

<sup>18</sup> *Flash* – Memórias *Flash* são um tipo de memória que em detrimento a *RAM*, permanecem com o seu conteúdo sem necessidade de fontes de alimentação.

sim que a máquina é ligada. Após a *BIOS* (ou *UEFI*, levando em consideração computadores de uso pessoal) realizar o autoteste do *hardware* chamado *POST*, a unidade de *boot* determinada (ou seja, aquela que está selecionada como primeira opção para o *boot*) é carregada com a aprovação no autoteste. São exemplos de unidade de *boot*: HD, SSD, *Pendrive* (ou outro dispositivo USB), uma unidade de mídia removível (como CD, DVD ou disquete), dentre outros [7]. O *bootloader* de primeiro estágio é carregado na RAM e executado assim que um desses dispositivos de boot é encontrado. A responsabilidade desse *loader* (leitor) de *boot* (que tem um único setor com menos de 512 bytes) é carregar o *bootloader* de segundo estágio (denominado também como *loader* de *kernel*, os dois estágios juntos formam o gerenciador de inicialização, como demos o exemplo do *grub2*, referenciado no início do capítulo anterior). Uma tela de inicialização é exibida, no momento em que o *bootloader* (de segundo estágio) foi alocado para a memória *RAM* e está executando, permitindo a escolha de um *kernel Linux*, de outro SO ou de outras ferramentas de inicialização. Selecionando um Sistema *Linux*, a imagem do *kernel* recebe a direção do sistema, somente após uma imagem *Linux* ser carregada e o *bootloader* (de segundo estágio) passar o controle para esse núcleo que é descompactado e inicializado. Em sequência, temos o processo de detecção do *hardware*, a montagem do dispositivo raiz, o carregamento dos módulos de *kernel* necessários e os recursos que são alocados a medida que são requisitados pelos programas em sua execução, todos esses são exemplos das principais incumbências do *kernel* [8].

Ao término do carregamento do *kernel*, o processo *init* é o primeiro programa no espaço de usuário (*user-space*) que é inicializado e sua execução perdura até o desligamento do sistema. Ele comumente recebe o *ID* de processo (*PID*) de número 1, sendo o responsável pela inicialização do sistema de alto nível. Em outras palavras, o processo *init* executa diversos *scripts* que desencadeiam no arranque dos demais processos e tarefas do sistema [8].

### 3.1 A ARQUITETURA DO *KERNEL LINUX*

A arquitetura de um Sistema Operacional *GNU/Linux* é dividida em níveis (ou camadas), podemos idealizá-la em dois espaços: o espaço do usuário e o espaço do *kernel* [9].

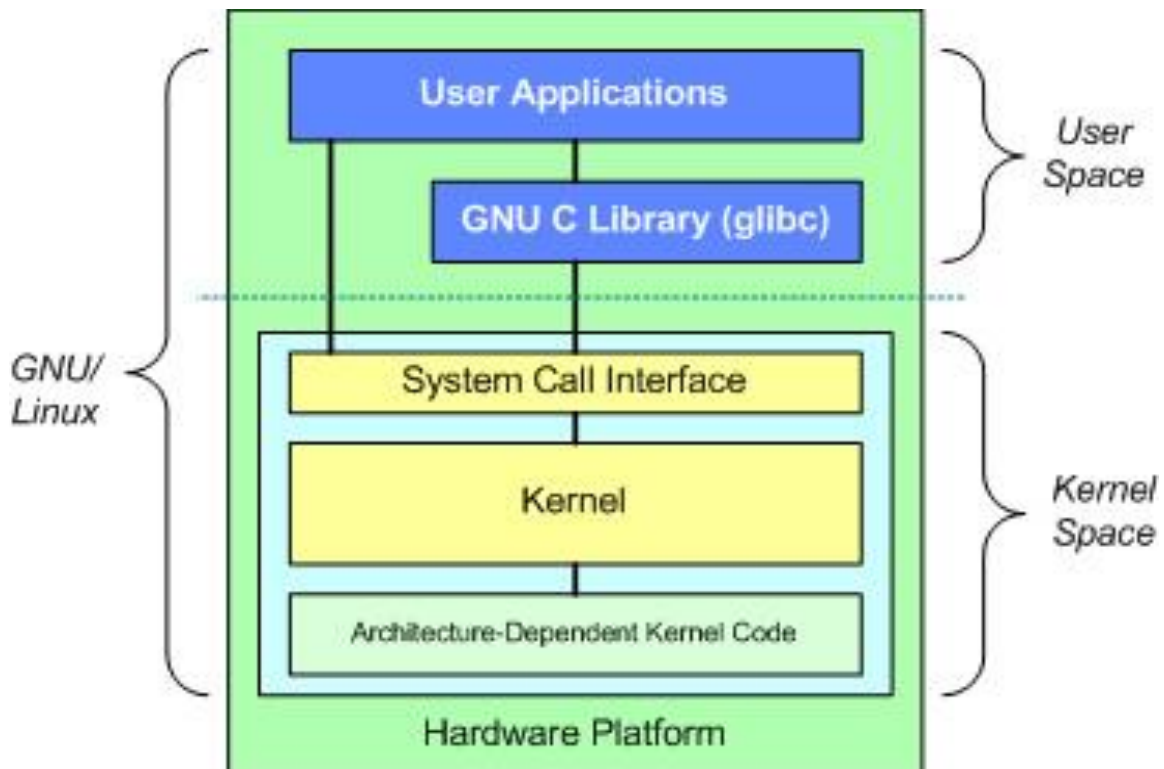


Figura 2: *GNU/Linux*, a arquitetura básica do sistema [9]

Os programas requisitados e executados pelos aplicativos do usuário ficam no espaço do usuário, onde também reside a *glibc* (Biblioteca C do *GNU*). As transições entre as aplicações do usuário e o *kernel* são realizadas graças ao mecanismo fornecido pela *glibc*, disponibilizando uma interface de chamada do sistema para se conectar ao *kernel* (devemos nos ater ao fato da importância desta biblioteca, uma vez que são ocupados espaços de endereços distintos e protegidos pelo *kernel* e pelo aplicativo do usuário). A interface de chamada do sistema, o *kernel* (entendendo essa parte do *kernel* como a que independe de uma arquitetura) e o código do *kernel* (dependente de uma arquitetura), residem nessa respectiva ordem do

nível mais acima para o mais abaixo no espaço do *kernel*. Funções fundamentais, como as de leitura e escrita, são implementadas na interface de chamada do sistema. Tendo em vista as arquiteturas de processadores as quais o *Linux* abrange, o *kernel* possui um código que é comum a todas elas e está abaixo da interface do sistema. Por fim temos o código do *kernel*, já este sendo específico para uma arquitetura de processador e também variando conforme a distribuição *Linux* na qual será aplicado [9].

### 3.2 O TIPO DE *KERNEL* DO *LINUX*

O *kernel Linux* é um único arquivo binário (um arquivo executável, em outras palavras, um único programa) onde as suas estruturas internas de dados e rotinas são acessíveis por todos os componentes funcionais do *kernel*, caracterizando-se como um *kernel* monolítico (“que forma um só bloco”, significado mais específico da palavra monólito que originalmente se refere ao termo “de uma só pedra”) [11].

Sob outra perspectiva, componentes de um Sistema *Linux* podem ser carregados e descarregados dinamicamente quando o usuário necessita deles, como módulos (estes que são referências frequentes aos *drivers*) podendo ser adicionados ou removidos (quando foram previamente desvinculados, não se fazendo mais necessários) em qualquer momento após o sistema ter iniciado. Esta característica modular é da estrutura do tipo *micro-kernel*, onde o *kernel* realiza comunicações, utilizando mensagens, com os subsistemas que estão no espaço do usuário. Portanto, o *kernel Linux* possui mecanismos de comunicação estritos entre unidades separadas que são divisões das partes funcionais do núcleo [10].

Diante disso, concluímos que a classificação do *kernel Linux* é ao mesmo tempo do tipo monolítico e modular, mesclando características de ambos. Todavia, nem sempre é possível generalizar o sistema quanto ao seu tipo, pois, a liberdade de modificar e adaptar o *kernel* para uma diversidade de usos, permite que o núcleo do *Linux* tenha características distintas das que foram apresentadas (somando, removendo ou alterando-as) [11].

### 3.3 VERSÕES DO *KERNEL*

As últimas versões do *kernel*, de acordo com o site <https://www.kernel.org/> (acesso em 03 de Dezembro de 2016), são: *mainline*<sup>19</sup> 4.9-rc7Os, *stable*<sup>20</sup> 4.8.12 e *longterm*<sup>21</sup> 4.4.36. Adicionalmente, temos a versão *Prepatch* (ou "RC" - *Release Candidates* – Candidatas a Lançamento), precisando ser compilada do código fonte, direcionada a desenvolvedores do *kernel* e aos fãs mais entusiasmados com o *Linux* (que gostam de usufruir dos recursos mais atuais, ao preço dos riscos eminentes do processo de testes e desenvolvimento). Linus Torvalds é o principal responsável e é ele quem faz o lançamento das versões *Prepatch*, sendo o mantenedor desta e também da versão estável [12].

O *kernel Mainline* entra em estágio "*Stable*", logo após seu lançamento, trazendo todas as correções de erros, ou fixação de *bugs*<sup>22</sup> da árvore *Mainline*, sendo aplicadas por um mantenedor do *Stable*. Entretanto, o *kernel* poderá sofrer algumas correções até o lançamento da próxima versão, a não ser que ela se torne uma versão de manutenção prolongada (ou seja, ficando por mais tempo sendo mantida – *longterm maintenance kernel* – versão do *kernel* aonde novos recursos não são inseridos, recebendo apenas correções de erros relevantes, existindo falhas de segurança alterações são recebidas, mas atualizações não são comuns) [14].

O próprio Linus Torvalds criou o *software* livre *Git*, em primeira instância, para o desenvolvimento do *kernel*, uma ferramenta hoje extensível no universo Open Source contemplando diversos *softwares* e que faz o controle de versão do *kernel* do *Linux* [13].

Percebemos que o *kernel*, assim como todo *software*, possui uma numeração relativa à sua versão podendo ser analisada a luz dos campos que são formados por esses números, cada campo carregando um sentido em particular. Tomemos esses campos na forma A.B.C.D (podendo aparecer também no formato A.B.C-D):

---

<sup>19</sup> *Mainline* – Versão de linha principal.

<sup>20</sup> *Stable* – Versão estável.

<sup>21</sup> *Longterm* - Versão de longo prazo.

<sup>22</sup> Bugs – Termo comum no âmbito computacional caracterizando defeitos em programas.



- A (*Version*) - O primeiro número é o mais pertinente, indicando a versão principal do *kernel* [13].
- B (*Patchlevel*) - Esse é o número da revisão indicando as mudanças fundamentais no funcionamento do *kernel* do *Linux*. Anteriormente, podíamos diferenciar entre uma versão estável e de desenvolvimento, observando quando um número nesse campo fosse par ou ímpar, respectivamente. Contudo, desde a versão 2.6, essa identificação foi abolida utilizando em substituição as letras RC (*Release Candidate* – Candidata a Lançamento) para as versões de desenvolvimento [13].
- C (*Sublevel*) - Este campo possui o número de revisão secundária, indicando correção de *bugs*, o suporte a novos dispositivos e pequenos *upgrades* (melhorias) no sistema [13].
- D (*Extraversion*) - Este campo é relativo as revisões imediatas, sendo utilizado quando temos interesse em distinguir duas compilações de um *kernel* de mesma versão, com o intuito de estabelecer módulos distintos [13].

### 3.4 O KERNEL E AS DISTRIBUIÇÕES

Nem todo *kernel* é diretamente vinculado aos desenvolvedores do *kernel Linux*, na verdade o mais corriqueiro é que a manutenção e/ou suporte prolongado do *kernel* sejam oferecidos e mantidos pelas próprias distribuições. Em outras palavras, estas versões não são fornecidas no endereço *kernel.org* e, portanto, não tendo nenhum suporte oficial dos desenvolvedores originais. Versões baixadas do site *kernel.org* são chamadas de versões de *kernel Vanilla*, o *kernel linux* “puro”, as distros habitualmente fazem adições de “*patches*<sup>23</sup>” que geram um *kernel* com peculiaridades conforme a distribuição em que será aplicado. Identificamos de forma sim-

---

<sup>23</sup> *Patches* – “Remendo” em uma tradução literal. São modificações comuns nas distros, mas feitas sem nenhuma relação com o *kernel* distribuído originalmente pelo “*kernel.org*”.

ples se um *kernel* é fornecido por uma distribuição ou se é um *kernel* Vanilla. Já que para ter um *kernel* “puro” é necessário compilar e instalar a versão do *kernel* adequada, caso não tenha realizado esse processo temos a convicção de que a versão do *kernel* é a da sua distribuição. Pela linha de comando é possível verificar qual versão do *kernel* está ativa no sistema *Linux*, com o comando “*uname -r*” (ou “*uname -a*”) [12].

### 3.5 O SHELL LINUX

O termo “*shell*”, em tradução literal, significa “casca”. O *shell* é um programa imprescindível ao *Linux*, sendo incumbido de interpretar e transmitir comandos para o sistema, retornando o resultado. A interface entre o *kernel* e o usuário, através do terminal, é dependente da atuação do interpretador de comandos (*shell*). As diversas tarefas do usuário do sistema são aplicações (programas) que só são realizadas ao serem antes invocados pelo *shell*. O *shell* mais usual é o *Bash*, sendo utilizado também o *ZSH* (*Zero shell*), dentre outros. Cada distro possui um *shell* instalado por padrão, todavia é possível alterá-lo numa determinada sessão executando o arquivo correspondente. Exemplificando, para o *bash* temos o caminho “/bin/bash” que o executa [13].

O terminal, habitualmente, possui em sua composição, na ordem da direita (de quem olha para a tela) para a esquerda: nome da máquina, escolhido pelo usuário durante a instalação ou modificado posteriormente; Seguido de “@” (arroba); Em sequência o nome do usuário, também definido pelo mesmo; Depois temos “:” (dois pontos); Seguido do diretório corrente; E, finalmente, a sequência termina com um caractere indicando o tipo de usuário que manipula o sistema. O “#” (cerquilha, trilha, jogo da velha, *sharp*, etc) referencia o usuário *root* (ou seja, aquele que possui privilégios administrativos) ou “\$” (cifrão) tratando-se de um usuário comum (sem privilégios de “superusuário” ou *root*). Uma prática comum é utilizar um programa que permite ao usuário normal executar tarefas como se fosse do grupo *root* (utilizando o programa *sudo*, por exemplo, concedendo momentaneamente privilégios de usuário *root*). Assim a segurança do sistema é melhor mantida, pois como somente

o usuário *root* pode executar tarefas administrativas, dificulta a ação de aplicações maliciosas [13].

## 4 QUAL DISTRIBUIÇÃO ESCOLHER?

Uma vantagem do *Linux* é a quantidade de opções que temos disponíveis, desde a escolha entre as distribuições até a escolha dos *softwares*, com uma grande variedade de ambientes gráficos, gerenciadores de janela, utilitários, ferramentas, etc.. O sistema operacional é construído conforme a escolha desses programas. Contudo, quem está dando os primeiros passos pode ficar desorientado diante de tantas possibilidades. Duas distros com os mesmos pacotes de *software* vão ser iguais em aparência, a diferença estará por dentro do sistema.

Agora faremos uma analogia um tanto quanto inusitada, comparando o Sistema *Linux* com um almoço. Quando pensamos em um almoço, pensamos em um prato com comida. A comida que vai nesse prato e a forma como ela é preparada é determinada conforme nossas escolhas. Pensemos em três formas de obter um almoço: ir a um restaurante, onde somos servidos com um cardápio e temos um prato feito; ir a um restaurante do tipo *self-service*<sup>24</sup>, onde temos uma certa variedade de alimentos e podemos escolher o que vamos comer; e ir a um supermercado, tendo uma grande variedade de alimentos que escolhemos para posteriormente prepará-los em nosso almoço. Dentre essas opções precisamos escolher uma que se encaixe melhor ao nosso gosto e/ou necessidade.

Gostando mais de ter algo prático e já preparado, provavelmente escolheríamos o prato feito. No entanto, se a preferência é ir direto a fonte e escolhermos o que precisamos, tendo uma diversidade de produtos e podendo poupar recursos nessa escolha, mesmo que tenhamos que preparar tudo ao nosso modo, notavelmente o supermercado seria nossa escolha. Ainda que busquemos um meio termo, não querendo ter o trabalho de preparar tudo, mas gostando de ter certas possibilidades no que vai ao nosso prato, ficaríamos com a opção de um *self-service*. Cada uma dessas opções possui vantagens e desvantagens, ficando a critério de cada um

---

<sup>24</sup> *Self-service* – Restaurante aonde pode-se fazer o prato, conforme as escolhas em uma grande demanda de opções alimentícias.

de acordo com suas necessidades e preferências. Assim também é com as distribuições *Linux*, imagine esse sistema no lugar do prato de almoço, nós escolhemos o que iremos por nele e de que forma colocaremos.

#### 4.1 ADQUIRINDO UMA DISTRO

Na escolha de uma distribuição, vemos que elas são disponibilizadas para download referenciadas como *releases* (lançamentos) e utilizando comumente o formato *ISO*, aonde pode ser montada em um dispositivo *USB* ou mídia *DVD*. Uma distro costuma ter mais de um tipo de “*release*” e, eventualmente, encontramos opções com mais de um ambiente *desktop* (*Desktop Environment*). E é nesta interface gráfica que temos o maior distanciamento visual entre uma escolha e outra.

#### 4.2 INTERFACE GRÁFICA

Um ambiente *desktop* é um conjunto de *softwares* para dar formatos, aparências, disponibilizar ferramentas, visualizações, dentre outras utilidades requeridas para as aplicações que funcionam em modo gráfico. Podemos dizer que o seu componente principal é o gerenciador de janelas (*Window Manager*), ele é o responsável por projetar a forma como visualizamos as janelas na tela do sistema.

Alguns ambientes são disponibilizados para quase todas as distros e outros são específicos de certas distribuições. O mais comum é que uma distribuição tenha uma gama de escolhas entre ambientes, inclusive possuindo dentre eles um ambiente padrão que costuma se destacar no uso com a distro.

*Gnome*<sup>25</sup> e *KDE*<sup>26</sup> são os ambientes mais completos de *softwares* e ferramentas, dotados de recursos visuais avançados, porém são os que mais costu-

---

<sup>25</sup> *Gnome* – Um dos *desktop environment* mais comuns, atualmente disponibiliza o ambiente *Gnome Shell* dotado de belos e modernos efeitos gráficos, bem como um vasto conjunto de ferramentas

mam consumir poder de processamento e memória. *LXDE*<sup>27</sup> e *XFCE*<sup>28</sup> são mais simples, precisando de mais trabalho para se tornarem mais atraentes graficamente, todavia o consumo é otimizado pela economia nos recursos do sistema.

Temos outros ambientes que são variações ou adaptações de um ambiente existente. Um exemplo é o *Unity*, ambiente que vem por padrão no *Ubuntu*, temos também o *Mate* e o *Cinnamon*, oriundos do *Linux Mint* e utilizados como ambientes padrões nessa distro. Os três são articulações do ambiente *Gnome*, sendo o *Unity* uma “camada externa” (*Shell*) do *Gnome 3*, já o *Mate* e o *Cinnamon* são “ramificações” (*forks*) do *Gnome 2* e *3*, respectivamente.

Fizemos menção a apenas alguns dos principais ambientes, tendo outros semelhantes ou até que se distanciam dos que foram referenciados. Contudo, é de costume que um ambiente *desktop* compartilhe o uso da biblioteca gráfica *GTK+*<sup>29</sup> ou da biblioteca gráfica *Qt*<sup>30</sup> que são as mais usadas.

Podemos então observar que um *Desktop Environment* é um conjunto de ferramentas gráficas como painéis, menus, barra de tarefas, botões, ícones, geren-

---

igualmente belas, sendo o ambiente utilizado como padrão em muitas distros como *Debian*, *Fedora* e outras.

<sup>26</sup> *KDE* – Ambiente *Desktop* conhecido também como *Plasma Desktop*, sendo um dos mais bonitos e que faz uso do *framework* (biblioteca) *Qt* para a escrita de seus aplicativos, em contraste ao *GTK* utilizado pelos aplicativos do ambiente *Gnome*. Contudo, ambas as aplicações podem coexistir harmoniosamente graças ao uso de ferramentas que permitem ter uma aparência e funcionalidade uniforme em ambos os ambientes. Também é utilizado como ambiente padrão em muitas distros, como por exemplo o *OpenSUSE* (que disponibiliza uma das melhores experiências encontradas com esse ambiente).

<sup>27</sup> *LXDE* – *Lightweight X11 Desktop Environment* – “Ambiente de Trabalho Leve para o X11”, talvez esse seja o ambiente *desktop* mais rápido, perdendo em leveza apenas para os *Window Managers*. Funcional, sendo ideal para quem procura economia em recursos, com pouca estética no ambiente e aplicações.

<sup>28</sup> *XFCE* – Esse *Desktop Environment* é leve e altamente personalizável, adaptando-se facilmente a modificações impostas, possui um amplo conjunto de softwares igualmente leves e funcionais.

<sup>29</sup> *GTK+* – A princípio nomeado como “*GIMP toolkit*”, é uma ferramenta para criação de interfaces gráficas desenvolvidas em C, advinda do ambiente *gnome* é amplamente utilizada em diversas aplicações.

<sup>30</sup> *Qt* – Interfaces gráficas desenvolvidas em C++ com uso deste *framework* oriundo do ambiente *KDE*.

ciadores de arquivos, dentre outros, e ele precisa necessariamente de um gerenciador de janelas para funcionar. Por isso, regularmente, os ambientes se encaixam de forma harmônica com um gerenciador de janelas.

Temos gerenciadores de janela simplistas consumindo poucos recursos, ao custo de disponibilizar um aspecto precário. Alguns são dotados de funcionalidades diversas, manuseando de diferentes formas as janelas, requerendo uma atenção especial para aprender como utilizá-los. Outros são dotados de efeitos visuais, fornecendo uma aparência formidável, contudo utilizando mais recursos do sistema.

Encontramos gerenciadores de janela numa diversidade maior do que a dos ambientes *desktop*, temos como exemplo: *Kwin* (nativo no *KDE*, possuindo um belo visual moderno e alguns efeitos avançados), *Mutter* (nativo no *Gnome 3* e ambientes derivados, também é dotado de efeitos visuais), *Compiz* (nativo no *Unity*, possuindo diversos efeitos 3D), *Openbox* (nativo no *LXDE*, é um gerenciador ideal para quem procura pouco consumo de memória), *Tiling Window Managers* (uma categoria que engloba gerenciadores de janelas “em mosaico”, dispendo as janelas “lado a lado” por padrão, costumam ser minimalistas, com muitas ferramentas em modo texto para manipular o controle sobre as janelas e a exibição destas), dentre outros.

Diante dessa diversidade de ambientes e gerenciadores, temos alguns fatores que podem guiar nossas escolhas.

#### 4.2.1 Interfaces minimalistas com foco em leveza

Se temos um sistema carente de recursos ou precisando de mais memória livre para processar outras tarefas, faremos uso de um ambiente e/ou gerenciador leve como *LXDE*, *XFCE*, *LXQT*, *Enlightenment*, *Openbox*, *i3wm*, etc.. Ambientes desse tipo podem ser lapidados, por conta do usuário “pôr a mão na massa”, tornando-se belos e mantendo sua eficiência. Por isso é uma opção mesmo para quem dispõe de recursos, sabendo manuseá-los podemos ter um ambiente “magro” e agradável. Assim sendo, costumam ser mais personalizáveis, podemos deixá-los exatamente como queremos.

#### 4.2.2 Interfaces completas com foco em aparência e efeitos visuais

A busca por beleza, ferramentas de gerenciamento, modernização, diversidade em efeitos visuais, certamente levará a escolha de um ambiente *desktop* como *Gnome*, *KDE*, *Unity*, *Cinnamon*, etc.. Diante do trabalho em adicionar ferramentas e suavizar a aparência dos ambientes mais leves, um iniciante por vezes desconhece esse processo de customização ou prefere um ambiente mais completo por não requerer ajustes. Contudo, a customização nesses ambientes costuma ser mais restrita ou não ter tanto impacto visual como nos ambientes mais leves.

#### 4.2.3 Interfaces intermediárias com foco em balancear aparência e leveza

Certos ambientes tentam manter uma boa aparência e um bom uso de recursos, não tendendo o foco mais para um lado ou para o outro e sim em atender ambos. São mais completos, mais atraentes, em primeira mão, do que as interfaces mais leves e são mais enxutos que as interfaces mais completas. Exemplos de ambientes desse tipo são o *Mate*, *Budgie*, *Pantheon* (sendo este um dos mais bonitos e elegantes, ambiente padrão na distribuição *Elementary* que é baseada no *Ubuntu*), dentre outros.

Diante do que foi visto acima, um ambiente *desktop* costuma seguir o mesmo foco adotado pelo gerenciador de janelas que ele faz uso. Se o gerenciador funciona com um foco em simplicidade e economia de recursos, ou em aparência e efeitos visuais, o conjunto de *softwares* escolhidos para o ambiente funcionará de forma análoga.

Em outra via, muitas vezes um gerenciador pode ser utilizado fora de um ambiente *desktop*, sendo possível customizá-lo modificando seu gerenciador de janelas nativo por outro, uma estratégia para mesclar o melhor de dois mundos. Tomando como exemplo o *Openbox* que é o gerenciador de janelas padrão no *LXDE* e *LXQT* (uma versão do *LXDE* unida ao antigo *desktop razor-qt*, entregando um novo



ambiente semelhante a leveza do *LXDE* e fazendo uso das belas aplicações da biblioteca *Qt*), podemos utilizá-lo em outros ambientes como, por exemplo, o *Gnome*, dessa forma tornando o ambiente mais leve, ao custo de perda de foco em aparência e efeitos, mas mantendo a gama de *softwares* que abrange o ambiente *gnome*.

Em alguns gerenciadores, ainda temos a opção de utilizá-lo isolado de qualquer ambiente *desktop*, como é o caso do próprio *Openbox*, adicionando outras ferramentas conforme a necessidade. Dessa forma, as possibilidades de usos e combinações são muitas.

Por isso essa escolha de um ambiente relaciona-se intrinsecamente com a escolha de uma distro, porque muitas distribuições oferecem determinado ambiente para pronto uso, encaixando-se de forma sinérgica ao sistema.

### **4.3 ESTILOS DE USO EM SISTEMAS *LINUX***

Relembrando da analogia ao almoço, sabemos agora o que pode ir ao nosso prato, portanto, podemos analisar de que maneira queremos nosso almoço, ou melhor, que *Linux* nós queremos.

Podemos ter um sistema feito para funcionar em total concordância com um determinado ambiente, para quem busca praticidade de uso, devendo tomar ciência de que outras alternativas para remanejamentos ou customizações são mais difíceis ou não se fazem presentes. Basicamente, ele será do início ao fim desta forma como adquirimos, mas se estamos certos que nossa opção de ambiente e sistema é esta, não tendo pretensão de mudá-la em um futuro, essa é a escolha mais proveitosa e sensata. Também é ótimo para quem está começando: se você nunca fez um almoço ou mesmo foi em um restaurante, é melhor escolher algo já pronto, depois de ir degustando, você terá opiniões para as próximas oportunidades.

Contudo, se queremos algumas opções de escolha, variações nos ambientes, mas sem ter que encaixar tudo por conta própria, temos aqui o estilo de uso encontrado na maior parte dos sistemas *Linux*. Podemos começar com um ambiente e posteriormente adicionar outros, ficando com mais de um ou escolhendo aquele que, por fim, julgamos como melhor para nosso uso. No *self-service* não somos nós

que preparamos a comida, mas somos totalmente livres para pôr em nosso prato o que estiver disponível.

E finalmente, temos a opção onde podemos poupar recursos, ficando com somente aquilo que queremos, ao custo de mais tempo e um certo conhecimento para o manuseio. Essa é a opção para usuários avançados que já sabem o que querem e como proceder. Para quem já tem uma certa noção e deseja aprender, esse também é o melhor caminho. Colocar a mão na massa é uma tarefa árdua, gasta mais tempo, mas temos a certeza de um maior controle sobre os custos, bem como o total selecionamento sobre o que queremos em nosso sistema. Com o conhecimento do que devemos fazer, podemos tirar melhor proveito do fruto do nosso trabalho.

É comum também uma distro ter mais de uma forma de utilização e estar em mais de uma das 3 categorias citadas. Vale ressaltar que esses estilos de uso são apenas referências para auxiliar quem está tendo um primeiro contato. Uma distribuição não segue um estilo veementemente como regra. Nós podemos usar as distros que são voltadas para a facilidade de uso de forma avançada, utilizando parcialmente ou totalmente a linha de comando e configurar o sistema pela edição de arquivos em formato de texto. Existem ferramentas que facilitam o uso de distribuições que exigem mais dos usuários, tornando-as nesse aspecto, semelhantes as distribuições com o foco em compreensibilidade. Embora esses não sejam os usos mais comuns, são exemplos de que a determinação do estilo está também nas mãos de quem vai utilizar as ferramentas do sistema. Cabe a nós, como usuários, conhecermos nosso estilo de uso que pode ser obtido, basicamente, com a união das nossas necessidades e gostos pessoais, para assim aproveitarmos melhor o que a família de *softwares Linux* nos oferece.

## 5 CONCLUSÕES E TRABALHOS FUTUROS

Podemos coligir alguns dos conceitos importantes em torno de Sistemas *Linux*. Entendemos o que é o *kernel*, o porquê de sua importância na história do *Linux* e seu contínuo desenvolvimento, sendo um dos maiores exemplos de potencial oriundo do universo de *softwares* livres. Analisamos um pouco da estrutura e funcionamento deste núcleo. Outrora criado com pouco mais de 10 mil linhas de código, a sua evolução permitiu que atualmente ele atingisse acima de 20 milhões de linhas de código. O *kernel* é um elemento importante, não só para um usuário de uma distro, sendo mais pertinente para empresas e instituições que dependem do seu funcionamento correto no cotidiano. Esse tipo de público são atraídos por pontos importantes como a estabilidade e seguridade do sistema, impactando no desenvolvimento constante do *kernel*, conseqüentemente providenciando diversas atualizações de segurança.

Distribuições *Linux* costumam focar em um ou mais tipos de uso, dentre eles temos: segurança, performance, inovações, *user friendly* (termo em inglês com significado de ser amigável ao usuário), dentre outros.

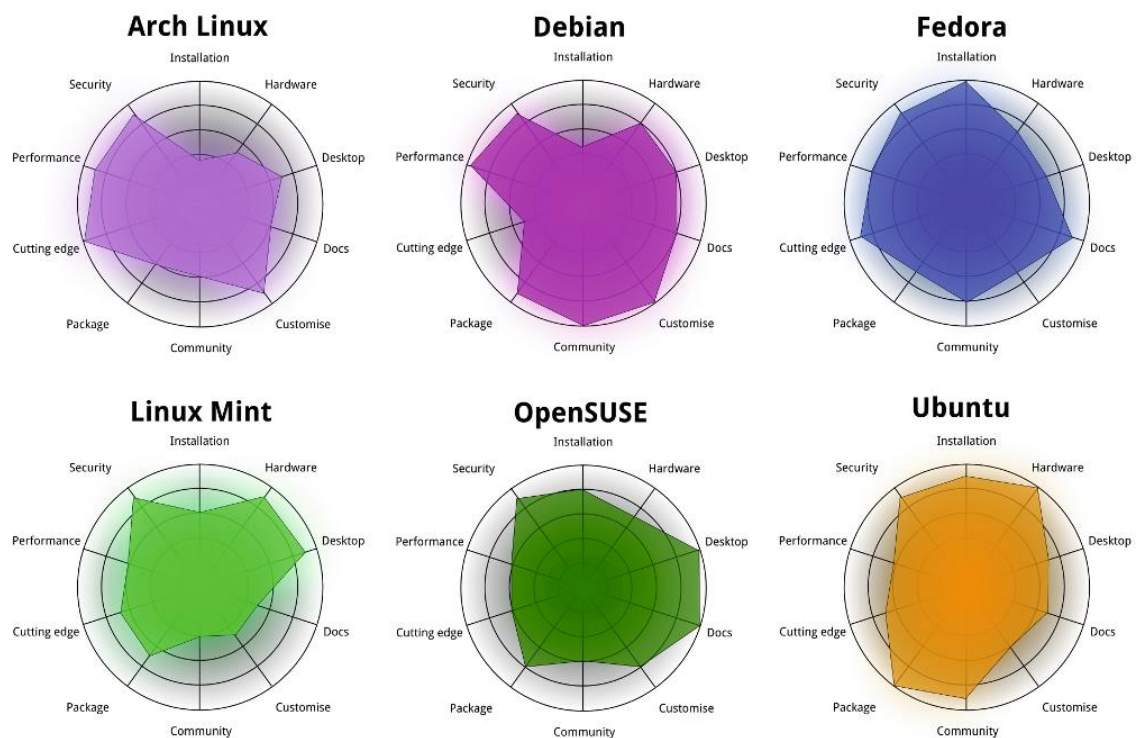


Figura 3: Comparativo entre Distribuições *Linux* em 2011 [15]

Algumas distribuições possuem uma comunidade ativa que costuma compartilhar e atender (em certos casos) situações de uso do sistema. Para quem está dando os seus primeiros passos no *Linux*, pode se ver com dificuldades em determinadas ocasiões e este é um ponto importante, uma vez que experiências de usuários contribuem para o auxílio de outros.

A quantidade de pacotes possuídas por uma distro deve atender, no mínimo, as necessidades mais básicas dos usuários. Quem procura aplicações de uso mais específico, pode não ser atendido de antemão se a distribuição possui poucos pacotes. Do contrário, compilar e instalar diretamente do código fonte pode ser uma opção, inclusive observamos distros que possuem um foco voltado para este tipo de utilização. Contudo, este não é um caminho corriqueiro, principalmente para quem é iniciante em Sistemas *Linux*.

Uma boa documentação é algo a ser valorizado em uma distro, já que esta vai ser a referência utilizada uma vez que pensemos em algo como um “manual de instruções”. Do mesmo modo que uma comunidade ativa é importante para o nova-

to, uma boa documentação também é. Na verdade uma boa documentação é um ponto bem querido por todo tipo de usuário, pois, esta será uma das primeiras orientações na utilização do sistema.

Para quem possui um tipo mais específico de *hardware*, como uma máquina muito antiga ou arquiteturas de computadores menos convencionais, como as de dispositivos embarcados, utilizar uma distribuição que atenda esse tipo de computador é algo essencial ou se tornaria inviável a utilização do Sistema *Linux*. Normalmente as distribuições possuem duas versões uma para arquiteturas x86 (32 bits) e outra para x86\_64 (64 bits), algumas contemplam *hardwares* mais específicos como os da arquitetura *ARM* (*Advanced RISC Machine* – arquitetura de processador presente, em sua maior parte, em dispositivos embarcados).

De todo modo, podemos ver que usuários podem se servir tendo em vista o vasto cardápio do mundo *GNU/Linux*. Observamos que muitas são as distros que atendem usuários iniciantes, com ferramentas e utilitários facilitando a sua instalação bem como a utilização do sistema. Em outra via, temos algumas distros aonde recursos gráficos costumam ser economizados, mantidos quando possível em modo texto, para o ganho de desempenho e não ferir certos princípios comuns na computação (como observamos no princípio *K.I.S.S*, manter as coisas simples e funcionais em assimetria a aplicações mais “estéticas” que por vezes são carregadas internamente de uma complexidade desnecessária, deixando assim os processos mais lentos).

Em nossas mãos temos o controle do que vamos utilizar, já observamos isso no âmbito da interface gráfica, a customização costuma ser um dos pontos chaves em boa parte das distros populares. Em contra partida, temos algumas distribuições que são especialmente montadas com um ambiente específico e um conjunto de *softwares* voltados para um tipo de usuário. Porém, as personalizações dentro de um ambiente são habitualmente universais a todas as distros (escolha de temas, ícones, mudar o posicionamento ou adicionar objetos e utilitários, dentre outras alterações).

Diante disso, perfazemos o conteúdo apresentado reportando ao *Linux* como uma espécie de guia aos novatos e interessados. Podemos obter aprofundamentos em questões de personalizações de distribuições e ambientes *desktop* de forma difundida na internet, não obstante o conteúdo interno do sistema ainda care-

ce de um material sólido e atualizado aqui em terras tupiniquins. Perante a relevância do *kernel Linux*, o núcleo de sistema mais famoso do mundo, almejo dar continuidade futuramente a esse trabalho dissecando o assunto de uma forma mais técnica figurando os elementos específicos do *kernel* e os sistemas que dele se beneficiam.

## REFERÊNCIAS BIBLIOGRÁFICAS

1. CAMPOS, Augusto. **O que é *Linux***. <<http://br-linux.org/faq-linux>> Acesso em 10 set. 2016
2. MORIMOTO, Carlos. ***Linux*, Guia Prático - Introdução: Uma breve história do *Linux***. <<http://www.hardware.com.br/livros/linux/introducao-uma-breve-historia-linux.html>> Acesso em 12 out. 2016.
3. <<http://distrowatch.com/dwres.php?resource=major>> Acesso em 22 out. 2016.
4. <<http://www.diolinux.com.br/2015/10/30-grandes-empresas-que-usam-linux.html>> Acesso em 13 nov. 2016.
5. FERREIRA, Ricardo. **Muito além do *kernel* – conheça todos os elementos que formam a estrutura do sistema *Linux***. <<http://www.linuxdescomplicado.com.br/2016/09/muito-alem-do-kernel-conheca-todos-os-elementos-que-formam-a-estrutura-do-sistema-linux.html>> Acesso em 15 nov. 2016.
6. <<https://www.oficinadanet.com.br/post/13858-o-que-e-kernel>> Acesso em 01 dez. 2016.
7. <<http://www.ibm.com/developerworks/br/library/l-lpic1-v3-101-2/index.html>> Acesso em 01 dez. 2016.
8. <<http://www.ibm.com/developerworks/br/library/l-linuxboot/index.html>> Acesso em 01 dez. 2016.
9. <<https://www.ibm.com/developerworks/br/library/l-linux-kernel/>> Acesso em 03 dez. 2016.
10. <<http://www.tldp.org/LDP/tlk/>> Acesso em 03 dez. 2016.
11. <<https://www.vivaolinux.com.br/artigo/O-Kernel-Linux?pagina=1>> Acesso em 03 dez. 2016.
12. <<https://www.kernel.org/category/releases.html>> Acesso em 03 dez. 2016
13. <<https://www.vivaolinux.com.br/artigo/O-Kernel-Linux?pagina=3>> Acesso em 03 dez. 2016.

14. <<https://elias.praciano.com/2015/12/conheca-as-diferentes-categorias-do-kernel-linux/>> Acesso em 03 dez. 2016.

15. <<http://tuxradar.com/content/best-distro-2011>> Acesso em 03 dez. 2016.