**Unix: O pai de todos os sistemas operacionais atuais**

É nos anos 60 que surge o embrião do Unix, o pai de praticamente todos os SO’s existentes hoje em dia.

O Unix é um sistema operacional criado por Kenneth Thompson após um projeto de sistema operacional não ter dado certo. O Unix foi o primeiro sistema a introduzir conceitos muito importantes para SOs como suporte a multiusuários, multitarefas e portabilidade.

Além disso, o Unix suporta tanto alterações por linhas de comando, que dão mais flexibilidade e precisão ao usuário, quanto às definições via interface gráfica, uma opção normalmente mais prática e menos trabalhosa do que a anterior.

Sua história remonta aos anos de 1960, quando Thompson, Dennis Ritchie e outros desenvolvedores se juntaram para desenvolver o sistema operacional Multics nos Laboratórios Bell da AT&T. A ideia era criar um sistema capaz de comportar centenas de usuários, mas diferenças entre os grandes grupos envolvidos na pesquisa (AT&T, General Eletronic e Instituto de Tecnologia de Massachusetts) levaram o Multics ao fracasso. Contudo, em 1969, Thompson começou a reescrever o sistema com pretensões não tão grandes, e aí surge o Unics.

O passo seguinte foi um retoque no nome e ele passa a se chamar Unix. Em 1973, com ajuda de Dennis Ritchie, a linguagem empregada no sistema passa a ser a C, algo apontado como um dos principais fatores de sucesso do sistema. Atualmente, uma série de SOs são baseados no Unix, entre eles, nomes consagrados como Gnu/Linux, Mac OS X, Solaris e BSD.

**Multitarefa e multiusuário**

É provável que o primeiro grande motivo da popularidade deste sistema sejam os conceitos que ele lançou no [mundo dos SOs](http://www.baixaki.com.br/tecnologia/2031-a-historia-dos-sistemas-operacionais.htm). Ao contrário de seus principais “concorrentes”, o Unix propôs um sistema multitarefa, capaz de executar dezenas de processos simultaneamente. De fato, a execução no Unix se dava de forma extremamente rápida, o que o fazia parecer ser multitarefa.

Outra característica do Unix é o suporte a multiusuário. O sistema permite que várias aplicações sejam executadas de modos independente e concorrente por usuários diferentes. Assim, eles podem compartilhar não somente hardwares, mas também softwares e componentes como discos rígidos e impressoras.

Atualmente, esses recursos parecem óbvios, mas há 40 anos eram novidades e fortes diferenciais para a escolha de um sistema operacional. Vale lembrar ainda que, nessa época, computadores pessoais também pareciam um sonho distante e o Unix era usado, basicamente, por universidades, governos e grandes indústrias.

**Distribuição livre**

Outro fator que com certeza influenciou na popularidade do Unix foi ele ter funcionado sob uma licença livre em seus primeiros anos de vida, tendo sido distribuído gratuitamente para universidades e órgãos governamentais dos Estados Unidos. Apenas depois de algum tempo a licença se tornou proprietária.

Contudo, a maioria dos sistemas criados com base no Unix funciona sob um sistema total ou parcial de código aberto. Desse modo, a proliferação do sistema foi impulsionada pela licença livre, principalmente das famílias BSD, Open Solaris e Linux.

**Mais seguro?**

Essa é uma questão um tanto quanto polêmica. Muitos falam que o Windows é vulnerável demais, enquanto outros dizem que sistemas Linux e Mac (baseados no Unix) não são infectados por vírus. A controvérsia é tanta que até já fez parte da seção “[Mito ou Verdade](http://www.baixaki.com.br/tecnologia/2911-mito-ou-verdade-outros-sistemas-sao-mais-seguros-que-o-windows-.htm)” aqui no Tecmundo.

O que se pode dizer é que, normalmente, sistemas baseados no Unix têm uma estrutura de execução de processos e de instalação de aplicativos um pouco mais complicada do que o Windows. Acaba sendo mais simples instalar um aplicativo executável no Windows do que compilar um pacote TAR.GZ no Linux, por exemplo.

Isso, somado à esmagadora popularidade do Windows entre os usuários, acaba por torná-lo um sistema mais vulnerável do que seus concorrentes. Contudo, o diretor executivo da Symantec, criadora do Norton Antivirus, Enrique Salem, [afirmou em fevereiro](http://www.baixaki.com.br/tecnologia/8648-macs-nao-sao-mais-seguros-que-pcs.htm) deste ano que Mac OS não é mais seguro do que Windows.

Vários movimentos no sentido de descomplicar o uso das dezenas de distribuições de Linux têm tornado o uso do sistema cada vez mais convencional, o que pode acarretar em problemas semelhantes aos do Windows. De qualquer modo, a estrutura dos sistemas tipo Unix talvez torne mais difícil a infecção por malwares.

**Linux**

Quando *Linus Torvalds* criou o **Linux**, certamente não sabia da importância que este projeto teria para os mais variados aspectos da computação. Graças à sua iniciativa, o Linux está presente em centenas de milhares de servidores, computadores pessoais e dispositivos portáteis em todo o mundo. Mas, como isso aconteceu? Por que Torvalds criou o Linux? Por qual motivo este projeto teve uma aceitação tão grande? Efetivamente, *o que é Linux?* Para obter as respostas para estas e outras questões relacionadas, nada melhor do que conhecer a história do Linux. É isso que o InfoWester apresenta a seguir.

Para muita gente, o Linux é meramente um sistema operacional. Esta definição não está errada, mas também não está completa. Na verdade, o Linux é parte de um todo, mais precisamente, é um kernel de código-fonte\* aberto, que foi - e é desenvolvido - ao longo do tempo graças à colaboração voluntária de desenvolvedores de várias partes do mundo. Em poucas palavras, código-fonte é um conjunto de instruções baseado em uma linguagem de programação que, depois de compilado ou interpretado, forma um software. Tendo acesso ao código-fonte, é possível saber como determinado programa ou recurso de software foi desenvolvido.

**O que é kernel?**

**Kernel** pode ser entendido como o núcleo do sistema operacional, isto é, como a parte essencial deste. Cabe ao kernel fazer o intermédio entre o hardware e os programas executados pelo computador. Isso significa que a junção do kernel mais os softwares que tornam o computador usável (drivers, protocolos de comunicação, entre outros), de acordo com a sua aplicação, é que formam o sistema operacional em si.

Para compreender melhor, você pode imaginar o kernel como sendo o chassi de um veículo. De acordo com a aplicação em questão, uma montadora pode adquirir um chassi e utilizá-lo para montar um carro para transportar cargas ou, se a necessidade for esta, para construir um automóvel de passeio para uma família.

Perceba que o kernel não é, necessariamente, um software manipulável pelo usuário. Ou seja, não se trata de algo tão simples a ponto de poder ser instalado e, logo em seguida, estar pronto para uso, como um programa de edição de textos, por exemplo. O kernel é uma base complexa, que serve de estrutura para o sistema, atuando nos "bastidores". Assim, o usuário sequer precisa saber de sua existência para poder utilizar o computador.

**A história do Linux**

A história do Linux começa no ano de 1991, pelas mãos de um estudante universitário finlandês chamado **Linus Torvalds**. O Linux foi criado por ele, não totalmente do "zero", mas sim como uma variação do *MINIX*.

O MINIX é um sistema operacional simples, criado por [Andrew S. Tanenbaum](http://en.wikipedia.org/wiki/Andrew_S._Tanenbaum), um renomado professor de computação que é conhecido pelos diversos livros que escreveu para a área.

Tanenbaum disponibilizou o MINIX principalmente para servir de auxílio no ensino de computação. Trata-se de um sistema operacional simples, que exige poucos recursos de hardware e cuja primeira versão foi lançada em 1987.

Dadas as suas finalidades acadêmicas, não só o MINIX foi disponibilizado de maneira gratuita e livre, como também o seu código-fonte completo. Assim, os estudantes de computação podiam - e podem - estudá-lo inteiramente para desenvolver suas habilidades ou mesmo para criar projetos derivados. Foi assim que Linus Torvalds entrou nesta história.

**A relação do Linux - e do MINIX - com o UNIX**

O MINIX também não foi escrito do "zero". Trata-se, na verdade, de um projeto baseado em um sistema operacional que tem grande participação na história da computação: o *UNIX*.

O surgimento do UNIX se deu em 1969, como um projeto da Bell Labs, laboratório pertencente à AT&T. Mas somente em meados da década seguinte tornou-se um sistema efetivamente disponível no meio acadêmico, o que permitiu a sua evolução e o surgimento de variações.

O UNIX, na verdade, começou em meados dos anos 1960 como um projeto a ser desenvolvido por um grupo de habilidosos programadores, entre eles, *Ken Thompson* e *Dennis Ritchie*: um sistema operacional de nome *Multics*.

O Multics era um projeto ambicioso, mas enfrentou vários problemas, entre eles, falta de recursos computacionais. Assim, no mesmo ano, Ken Thompson decidiu criar algo mais "realista", chamando o novo projeto de *Unics*. Tempos depois, o nome foi mudado para *UNIX*, denominação que permanece até hoje.

Apesar de haver outros programadores envolvidos com a criação do UNIX, Ken Thompson e Dennis Ritchie são os nomes mais lembrados porque ambos, em 1973, praticamente reescreveram o UNIX a partir da linguagem C.

C é uma criação de Dennis Ritchie, daí o seu comprometimento com o trabalho envolvendo a linguagem e o UNIX. Por causa de seus recursos, a linguagem passou a ser utilizada em vários outros projetos mais complexos, inclusive no desenvolvimento de outros sistemas operacionais, fazendo com que o seu criador fosse reconhecido mundialmente como um dos grandes nomes da computação (infelizmente, [Dennis Ritchie faleceu num outubro de 2011](http://www.infowester.com/noticias/falece-dennis-ritchie-um-dos-criadores-do-unix-e-da-linguagem-c/)).

O UNIX teve grande aceitação não somente em universidades, mas também em ambientes corporativos, resultando no surgimento de variações diversas do sistema, como as versões BSD e o Solaris. O MINIX, assim como o Linux, é  uma delas, o que não quer dizer que ambos sejam iguais ao UNIX, mas notoriamente parecidos.

**O surgimento do Linux**

Linus Torvalds, então com quase 20 anos, começou a estudar [ciência da computação](http://www.infowester.com/cursoscomp.php) na Universidade de Helsinki, na Finlândia, em 1988. Cerca de dois anos depois, aproveitando o conhecimento que tinha e estava adquirindo sobre a linguagem C, decidiu criar a sua própria implementação de um terminal em seu recém obtido computador 80386, principalmente para acessar o servidor UNIX da instituição de ensino. Isso porque ele já havia testado o MINIX para esta finalidade, mas não estava satisfeito com os seus recursos.

A intenção de Torvalds era a de fazer com que o projeto rodasse especificamente em sua máquina 80386, com o desenvolvimento sendo feito a partir do MINIX. O trabalho avançou de tal forma que chegou um ponto onde Torvalds já tinha um kernel funcional em mãos.

Em 1991, Linus Torvalds decidiu divulgar abertamente o seu projeto. Para isso, publicou mensagens na Usenet (uma espécie de antecessora da internet, baseada em troca de mensagens) pedindo sugestões e colaborações para a sua iniciativa.

O início da trajetória do Linux não foi isenta de problemas. Uma dos obstáculos que Torvalds teve que enfrentar foram as críticas do professor Andrew S. Tanenbaum, que em suas declarações afirmou que o "Linux é obsoleto", especialmente por este ter "design monolítico".

Tanenbaum não estava contente com o fato de o Linux ter sido preparado especificamente para rodar com o processador 80386 que, além de caro, teria sua arquitetura substituída futuramente, o que, na verdade, não aconteceu.

Linus respondeu às críticas e continuou seu trabalho, contando com o apoio de cada vez mais pessoas. Com o passar do tempo, o Linux acabou inclusive sendo portado para várias outras plataformas, o que certamente contribuiu para o seu sucesso.

**O nome Linux**

O projeto já era realidade, mas não tinha um nome. Inicialmente, Torvalds atribuiu ao kernel a denominação *Freax*, uma mistura de *free* (livre) com *freak* (monstruoso, esquisito) e a letra 'x', para lembrar o UNIX.

O programador *Ari Lemmke*, depois de sugerir a Torvalds que colocasse o projeto em uma rede para torná-lo mais acessível, decidiu criar no servidor de FTP que hospedaria o software uma pasta de nome "linux" (muito provavelmente, uma mistura de Linus com UNIX), já que não havia gostado de Freak. A denominação "Linux" acabou "pegando" e é, tal como você vê, utilizada até hoje.

**GNU/Linux**

Você já deve ter visto em vários lugares - inclusive aqui no InfoWester - a expressão *GNU/Linux*. O que isso significa?

Tal como você já sabe, o Linux, por si só, é um kernel. Sozinho, um kernel não tem muita utilidade. É necessário "juntá-lo" a um conjunto de softwares para que tenhamos, efetivamente, um sistema operacional em condições de uso. É aí que o projeto [GNU](http://www.gnu.org/) entra.

GNU é a sigla para um nome curioso: "*GNU is Not UNIX* (GNU Não é UNIX)". Trata-se de um projeto que teve início em 1984, pelas mãos de *Richard Stallman*, que queria criar um sistema compatível com UNIX, mas sem utilizar código deste.

Com o passar dos anos, o projeto foi ganhando recursos, como compiladores e editores de texto. Mas, faltava um elemento importantíssimo: um kernel. Stallman e seus colaboradores estavam trabalhando em um kernel de nome *Hurd*, mas dada a demora em concluí-lo, muitos daqueles que precisavam ou queriam usar software GNU decidiram recorrer a algo que souberam ser capaz de atender à necessidade que tinham: o Linux.

Então, basicamente, o Linux que temos hoje é conhecido por trabalhar em conjunto com software GNU. Por isso, muitos integrantes e simpatizantes de movimentos ligados ao software livre defendem a ideia de que, quando houver referência ao sistema operacional como um todo, o nome *GNU/Linux* seja utilizado. Acontece que, por comodidade ou simplesmente desconhecimento, muitas pessoas criaram o hábito de chamar todo o conjunto de Linux e não somente o kernel.

**Distribuições Linux**

Você já sabe que o Linux, ao contrário de outros sistemas baseados no UNIX ou mesmo deste, não é um sistema operacional como um todo. Mas, sendo um kernel disponível de maneira gratuita e com código-fonte aberto, qualquer pessoa ou organização pode juntá-lo a um conjunto de softwares para criar um sistema operacional customizado.

Ao longo dos últimos anos, foi justamente isso que aconteceu. Vários grupos ou mesmo empresas se organizaram e criaram seu próprio sistema operacional baseado em Linux. Cada uma delas recebe o nome de "distribuição Linux" (ou "distribuição GNU/Linux").

Há várias distribuições Linux por aí, para os mais diversos fins. Muitas inclusive fazem parte de negócios rentáveis, onde a empresa fornece, por exemplo, o sistema operacional de graça, mas obtém receita a partir de serviços de suporte técnico. Naturalmente, aquelas distribuições que se destinam ao segmento de usuários domésticos são mais populares.

Para o público em geral, a distribuição mais famosa é o [Ubuntu](http://www.ubuntu.com), da empresa [Canonical](http://www.canonical.com/). Por padrão, são lançadas novas versões do Ubuntu em todos os meses de abril e outubro de cada ano. Há um esquema de numeração que ajuda a identificar a época de lançamento: a versão 11.10 do Ubuntu, por exemplo, foi lançada em 2011, no mês de outubro. É possível saber disso porque a indicação de ano aparece primeiro (11) e, depois do ponto, vem a indicação do número do mês (10).

É claro que há outras distribuições renomadas, entre elas:

* [Fedora](http://fedoraproject.org/) (ligada à [Red Hat](http://www.redhat.com/));
* [Debian](http://www.debian.org);
* [Mandriva](http://www.mandriva.com);
* [Linux Mint](http://linuxmint.com/);
* [CentOS](http://www.centos.org/) (com foco em servidores);
* [Slackware](http://www.slackware.com/).

É possível obter informações sobre estas e outras distribuições Linux no site [DistroWatch.com](http://distrowatch.com) (em inglês).

**Versões do kernel**

Periodicamente, novas versões do kernel Linux são lançadas. Atualizações são naturais para qualquer software e ocorrem para atribuir melhorias a determinadas funcionalidades, para corrigir falhas (*bugs*) e, no caso de sistemas operacionais, para adicionar recursos importantes ao kernel, principalmente compatibilidade com novos hardwares.

Normalmente, cada versão do kernel é representada por três números distintos separados por pontos, por exemplo: 2.6.24. O primeiro número indica a versão do kernel. O segundo número indica a última revisão feita até o momento naquela versão. O terceiro número, por sua vez, indica uma revisão menor, como se fosse uma "revisão da última revisão" do kernel. Um quarto número pode ser utilizado para indicar uma atualização importante naquela versão.

É válido frisar que antes da série 2.6.x, a numeração do kernel tinha o seguinte esquema: se o segundo número da representação fosse ímpar, significava que aquela série ainda estava em desenvolvimento, ou seja, era uma versão instável e em fase de testes ou aperfeiçoamentos. Se o número fosse par, significava que aquela série já tinha estabilidade para ser disponibilizada para uso.

A numeração sofreu outra mudança em julho de 2011, quando foi lançada a versão 3.0 do Linux. Das versões 1.x para 2.x houve significativas alterações no kernel. No entanto, da 2.x para a 3.x não houve nada tão importante assim, tanto que, pelo esquema, esta versão deveria ser 2.6.40.

Tudo indica que a mudança aconteceu por uma questão de comodidade e "estética". Há também quem afirma que a numeração 3.0 foi dada como forma de comemorar o aniversário do Linux, que completou 20 anos de existência em 2011.

**A licença do Linux**

Uma licença é, em poucas palavras, um documento que explica como determinado software pode ser utilizado. No que se refere a programas de código-fonte aberto, há vários tipos de licenças disponíveis. O Linux utiliza a *GPL* (*GNU Public Licence*).

Vale frisar que, inicialmente, Linus Torvalds aplicou ao Linux uma licença própria, que tinha restrições para uso comercial. A GPL só foi adotada somente em 1992, mesmo porque o Linux já era utilizado com software GNU.

A GPL é uma licença criada pela [*Free Software Foundation*](http://www.fsf.org/) (organização fundada por Richard Stallman) baseada nas liberdades que a entidade defende:

* liberdade de executar o programa, para qualquer propósito (liberdade zero);
* liberdade de estudar como o programa funciona e adaptá-lo às suas necessidades (liberdade 1), sendo o acesso ao código-fonte um pré-requisito para esta aspecto;
* liberdade de distribuir cópias de forma que você possa ajudar ao seu próximo (liberdade 2);
* liberdade de melhorar o programa e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie (liberdade três). Novamente, aqui o acesso ao código-fonte é um pré-requisito.

Um software não pode utilizar a GPL se não corresponder a todos estes requisitos.

A GPL surgiu em 1989, mas foi revisada em 1991 para atender a determinadas necessidades, resultando na GPLv2 (GPL versão dois). Em 2007, surgiu a GPLv3 (GPL versão três). É possível consultar a GPL no seguinte link (em inglês): [www.gnu.org/licenses/gpl.html](http://www.gnu.org/licenses/gpl.html).