FLISoL

Mas que treco é esse de Software Livre?

Na história do mundo, quais os eventos que forçam a evolução tecnológica?

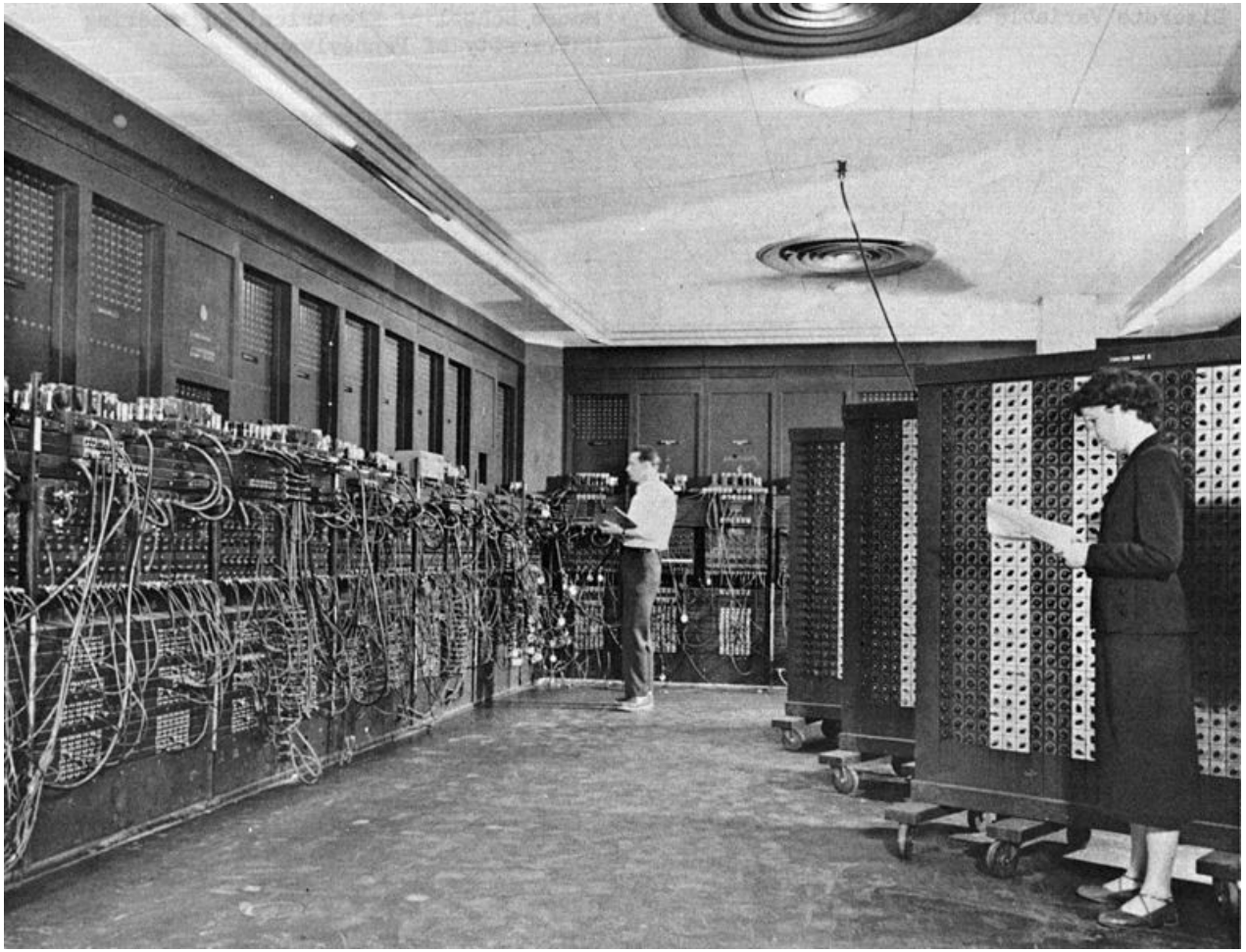
Durante a **Segunda Guerra Mundial**, houve um grande impulso para o desenvolvimento de novas tecnologias militares, incluindo radares, aviões a jato, mísseis balísticos, criptografia e computação. Muitas dessas tecnologias foram posteriormente adaptadas para uso civil após a guerra.

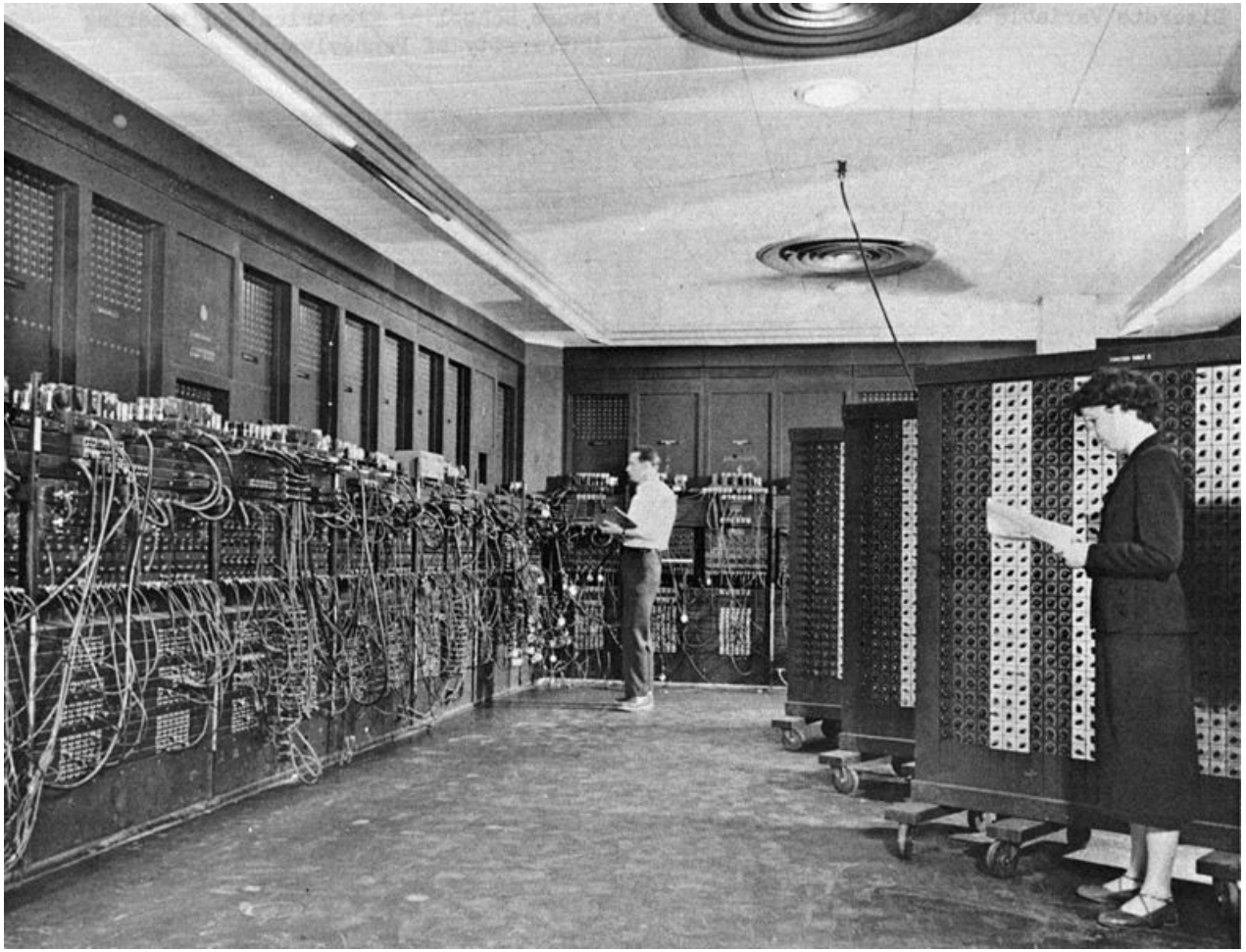
Durante a **Guerra Fria**, houve um **grande incentivo** para a evolução tecnológica em ambas as superpotências, em áreas como armamento, espaço e **computação** resultou em avanços significativos em tecnologias que acabaram tendo aplicações civis importantes.

Durante a **Guerra Fria**, houve um **grande incentivo** para a evolução tecnológica em ambas as superpotências, em áreas como armamento, espaço e **computação** resultou em avanços significativos em tecnologias que acabaram tendo aplicações civis importantes.

A **Guerra Fria** também foi responsável por impulsionar o desenvolvimento da **internet**. O Departamento de Defesa dos EUA criou a **ARPANET** em **1969**. A **ARPANET** foi a base da internet moderna.

Mas o que a guerra tem a ver com o Software?





Este é o **ENIAC**, dito por muitos como o **primeiro computador eletrônico**.

Foi construído entre **1943** e **1945** na Universidade da Pensilvânia, nos Estados Unidos, como parte de um projeto militar para calcular trajetórias de artilharia.

Seu tamanho?

167 m²

Introdução feita!

Vamos focar em Software...

ANOS 60

ANOS 60

- Os computadores eram **grandes**... **bem grandes**!

ANOS 60

- Os computadores eram **grandes**... **bem grandes**!

- Computador era coisa de **Universidades**, **Governos** e **grandes empresas**...

ANOS 60

- Os computadores eram **grandes**... **bem grandes**!

- Computador era coisa de **Universidades**, **Governos** e **grandes empresas**...

- Cada computador tinha seu próprio **sistema operacional** e seus respectivos softwares.

ANOS 60

- Os computadores eram **grandes**... **bem grandes**!

- Computador era coisa de **Universidades**, **Governos** e **grandes empresas**...

- Cada computador tinha seu próprio **sistema operacional** e seus respectivos softwares.

- Não havia sistemas operacionais **multiplataformas**...

ANOS 60

- Os computadores eram **grandes**... **bem grandes**!

- Computador era coisa de **Universidades**, **Governos** e **grandes empresas**...

- Cada computador tinha seu próprio **sistema operacional** e seus respectivos softwares.

- Não havia sistemas operacionais **multiplataformas**...

*Mas no final da década, o **vício** falou alto...*

Esta é a possível tela do jogo **Space Travel**

Segundo as más línguas, esse jogo pode ter sido responsável por um grande passo da humanidade... Afinal de contas o que um **viciado** não faz para manter o seu vício?

Eis os **viciados** (segundo as más línguas... claro!)

Eis os **viciados** (segundo as más línguas... claro!)

A esquerda **Ken Thompson** e a direita **Dennis Ritchie**.

Mas eles são gênios! Eles criaram muita coisa!

Mas eles são gênios! Eles criaram muita coisa!

Ken Thompson criou a **Linguagem** de programação **B**

Mas eles são gênios! Eles criaram muita coisa!

Ken Thompson criou a **Linguagem** de programação **B**

Que logo foi substituida pela **Linguagem C** criada por **Dennis Ritchie**.

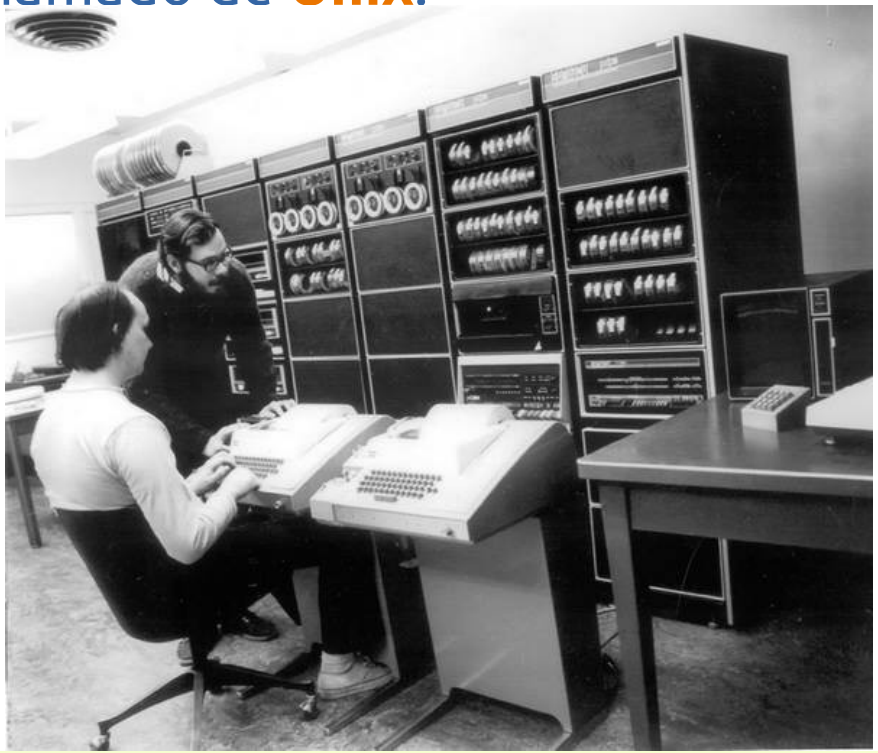
Mas eles são gênios! Eles criaram muita coisa!

Ken Thompson criou a **Linguagem** de programação **B**

Que logo foi substituida pela **Linguagem C** criada por **Dennis Ritchie**.

A **Linguagem C** é ainda uma das linguagens mais usadas no mundo!

Segundo as más línguas, em **1969**, o **PDP-7** em que eles *jogavam, opss digo* trabalhavam na **Bell Labs** (AT&T atualmente) foi desativado e eles ficaram sem acesso ao jogo. Então, eles decidiram criar seu próprio sistema operacional para poder jogar novamente. Esse sistema operacional posteriormente foi chamado de **Unix**.



Mas o que é o Unix?

Mas o que é o Unix?

Foi o primeiro **Sistema Operacional** que...

Mas o que é o Unix?

Foi o primeiro **Sistema Operacional** que...

- **Multiplataforma**, funcionar em qualquer hardware.

Mas o que é o Unix?

Foi o primeiro **Sistema Operacional** que...

- **Multiplataforma**, funcionar em qualquer hardware.

- **Multitarefas**, executar várias tarefas ao mesmo tempo.

Mas o que é o Unix?

Foi o primeiro **Sistema Operacional** que...

- **Multiplataforma**, funcionar em qualquer hardware.

- **Multitarefas**, executar várias tarefas ao mesmo tempo.

- **Multiusuários**, vários usuários usando ao mesmo tempo!

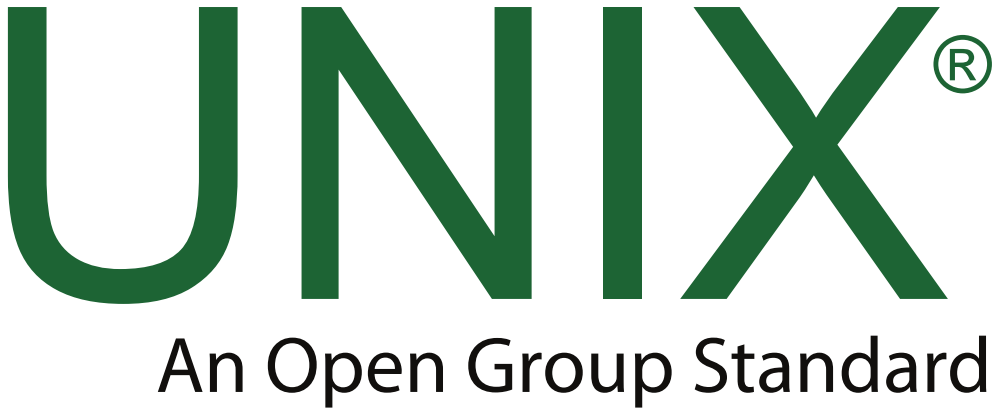
Mas o que é o Unix?

O **Unix** tornou-se uma opção popular para **empresas** e **instituições de ensino superior** que precisavam de um sistema operacional **poderoso** e **flexível** para suportar muitos usuários simultâneos.

Mas o que é o Unix?

O **Unix** tornou-se uma opção popular para **empresas** e **instituições de ensino superior** que precisavam de um sistema operacional **poderoso** e **flexível** para suportar muitos usuários simultâneos.

O **Unix** também introduziu muitos **conceitos** que se tornaram **padrão** em **sistemas operacionais modernos**, como a utilização de uma **interface de linha de comando**, um **sistema de arquivos hierárquico** e o **conceito de "tudo é um arquivo"**. Ainda hoje, muitos sistemas operacionais, incluindo os **GNU/Linux** e o **MacOS**, são baseados no **Unix** ou foram fortemente influenciados por ele.


UNIX®
An Open Group Standard

UNIX®

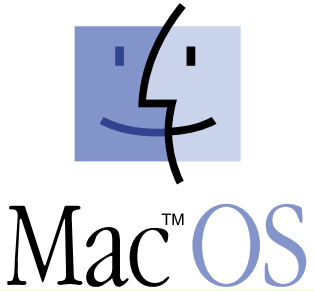
An Open Group Standard

UNIX®

An Open Group Standard

UNIX®

An Open Group Standard



SOFTWARE PROPRIETÁRIO

William Henry Gates III (**Bill Gates**)

28/10/1955

Seattle - Washington, Estados Unidos

Fundou a Microsoft em 1975.

Em **1976** (aos 20 anos de idade) ficou conhecido por escrever a **carta** entitulada de **Open Letter to Hobbyists** para o **Congresso Americano**.

Esta carta foi um **marco** para a definição do **Software Proprietário**.

SOFTWARE PROPRIETÁRIO

"Na **carta**, **Gates** argumentou que a **prática de compartilhar softwares sem pagar pelos direitos autorais era prejudicial para a indústria de software**, pois desencorajava as empresas de investirem na criação de novos softwares e, consequentemente, na inovação tecnológica. **Gates** afirmou que a **Microsoft®** não poderia continuar a desenvolver softwares se não pudesse lucrar com eles e pediu aos usuários de computadores pessoais que respeitassem os direitos autorais da empresa.

A carta foi muito importante para a história do software proprietário, pois marcou o início da mudança de mentalidade em relação aos **direitos autorais do software**. Antes da carta de Gates, muitas pessoas **consideravam o compartilhamento de softwares sem pagamento como uma prática aceitável e até mesmo desejável**. A **carta ajudou** a estabelecer a ideia de que os **softwares deveriam ser protegidos por direitos autorais e que as empresas deveriam ter o direito de lucrar com seus produtos**."

OpenAI ChatGPT

CONSEQUÊNCIAS

ANTES:

CONSEQUÊNCIAS

ANTES:

- **Softwares** eram criados em maioria em **Instituições Educacionais**.

CONSEQUÊNCIAS

ANTES:

- **Softwares** eram criados em maioria em **Instituições Educacionais**.

- A **colaboração** e a **troca** de softwares era comum entre **programadores**, **comunidades**, **grupos**, **Instituições** e **empresas**.

CONSEQUÊNCIAS

ANTES:

- **Softwares** eram criados em maioria em **Instituições Educacionais**.

- A **colaboração** e a **troca** de softwares era comum entre **programadores**, **comunidades**, **grupos**, **Instituições** e **empresas**.

- Se distribuía **códigos-fontes** e não **binários**.

CONSEQUÊNCIAS

ANTES:

- **Softwares** eram criados em maioria em **Instituições Educacionais**.

- A **colaboração** e a **troca** de softwares era comum entre **programadores**, **comunidades**, **grupos**, **Instituições** e **empresas**.

- Se distribuía **códigos-fontes** e não **binários**.

DEPOIS:

CONSEQUÊNCIAS

ANTES:

- **Softwares** eram criados em maioria em **Instituições Educacionais**.

- A **colaboração** e a **troca** de softwares era comum entre **programadores**, **comunidades**, **grupos**, **Instituições** e **empresas**.

- Se distribuía **códigos-fontes** e não **binários**.

DEPOIS:

- **Códigos-fontes fechados!** Softwares passaram a ser distribuídos/vendidos em **binários**.

CONSEQUÊNCIAS

ANTES:

- **Softwares** eram criados em maioria em **Instituições Educacionais**.

- A **colaboração** e a **troca** de softwares era comum entre **programadores**, **comunidades**, **grupos**, **Instituições** e **empresas**.

- Se distribuía **códigos-fontes** e não **binários**.

DEPOIS:

- **Códigos-fontes fechados!** Softwares passaram a ser distribuídos/vendidos em **binários**.

- **Comunidades** e **programadores** passaram a serem proibidos de usar livremente vários os softwares que **usavam/contribuíam a anos**.

SOFTWARE LIVRE

Richard Matthew **Stallman**

16/03/1953

Nova York, Estados Unidos.

Formado em Física na Universidade Harvard em **1974.**

Trabalhou no **Laboratório** de **Inteligência Artificial** do **MIT** (Massachusetts Institute of Technology) como **programador** e **pesquisador** em tempo integral. Ele continuou a desenvolver sua carreira no **MIT,** onde mais tarde se tornou um dos desenvolvedores mais **proeminentes** da **comunidade hacker** do **MIT.**

SOFTWARE LIVRE

Na década de **1970**, **Stallman** trabalhava no **Laboratório de Inteligência Artificial do MIT** (Massachusetts Institute of Technology), onde desenvolveu o editor de texto **Emacs**.

SOFTWARE LIVRE

Na década de **1970**, **Stallman** trabalhava no **Laboratório de Inteligência Artificial do MIT** (Massachusetts Institute of Technology), onde desenvolveu o editor de texto **Emacs**.

Stallman protegeu o **Emacs** com uma licença chamada **Emacs Public License** (**EPL**), em si a **EPL** garantia que o **Emacs** continuasse **Livre**.

SOFTWARE LIVRE

Na década de **1970**, **Stallman** trabalhava no **Laboratório de Inteligência Artificial do MIT** (Massachusetts Institute of Technology), onde desenvolveu o editor de texto **Emacs**.

Stallman protegeu o **Emacs** com uma licença chamada **Emacs Public License** (**EPL**), em si a **EPL** garantia que o **Emacs** continuasse **Livre**.

Em **1978**, o **MIT** fez um **acordo** com a empresa **Symbolics** para desenvolver computadores Lisp baseados no hardware **PDP-10**.

SOFTWARE LIVRE

Em **1980**, **Stallman** trabalhava com uma **impressora laser Xerox**, e a mesma aparentava um **defeito** ao imprimir. Como o driver da mesma era acompanhado de seu **código-fonte**, **Stallman** pode **corrigir** o defeito.

SOFTWARE LIVRE

Em **1980**, **Stallman** trabalhava com uma **impressora laser Xerox**, e a mesma aparentava um **defeito** ao imprimir. Como o driver da mesma era acompanhado de seu **código-fonte**, **Stallman** pode **corrigir** o defeito.

Tempos depois houve a troca da impressora por um **outro modelo** do mesmo fabricante, e o modelo novo também apresentava o **mesmo defeito**. **Stallman** ao ir corrigir percebeu que o driver **não mas era acompanhado do código-fonte**.

SOFTWARE LIVRE

Em **1980**, **Stallman** trabalhava com uma **impressora laser Xerox**, e a mesma aparentava um **defeito** ao imprimir. Como o driver da mesma era acompanhado de seu **código-fonte**, **Stallman** pode **corrigir** o defeito.

Tempos depois houve a troca da impressora por um **outro modelo** do mesmo fabricante, e o modelo novo também apresentava o **mesmo defeito**. **Stallman** ao ir corrigir percebeu que o driver **não mas era acompanhado do código-fonte**.

Ele contatou com a **Xerox** pedindo o código-fonte para poder consertar o problema, e a resposta da **Xerox** foi que liberaria o código-fonte para que ele o fizesse, inclusive o pagaria pelo serviço, mas o **código-fonte continuaria fechado para todos!**

SOFTWARE LIVRE

Em **1983**, a **Symbolics** decidiu incluir o **Emacs** em seus computadores Lisp como parte de seu sistema operacional. Para isso, eles fizeram uma **oferta** para **comprar** os **direitos** do **Emacs** do **MIT**.

SOFTWARE LIVRE

Em **1983**, a **Symbolics** decidiu incluir o **Emacs** em seus computadores Lisp como parte de seu sistema operacional. Para isso, eles fizeram uma **oferta** para **comprar** os **direitos** do **Emacs** do **MIT**.

A **oferta** foi **aceita** pelo **MIT**, mas com a condição de que o **Emacs** continuasse a ser distribuído sob a **Licença Pública Emacs**. No entanto, a **Symbolics violou** essa condição ao comercializar o **Emacs sem disponibilizar seu código-fonte** e **sem permitir que outros desenvolvedores fizessem modificações e distribuíssem suas próprias versões do software**.

SOFTWARE LIVRE

Em **1983**, a **Symbolics** decidiu incluir o **Emacs** em seus computadores Lisp como parte de seu sistema operacional. Para isso, eles fizeram uma **oferta** para **comprar** os **direitos** do **Emacs** do **MIT**.

A **oferta** foi **aceita** pelo **MIT**, mas com a condição de que o **Emacs** continuasse a ser distribuído sob a **Licença Pública Emacs**. No entanto, a **Symbolics violou** essa condição ao comercializar o **Emacs sem disponibilizar seu código-fonte** e **sem permitir que outros desenvolvedores fizessem modificações e distribuíssem suas próprias versões do software**.

Quando **Stallman** descobriu a venda do **Emacs** pela **Symbolics**, ele ficou **indignado** e enviou uma carta para a empresa **exigindo** que eles se **conformassem** com a **EPL**. A **Symbolics** ignorou a carta!

SOFTWARE LIVRE

Inconformado, **Stallman** a **reescreveu** o **Emacs** do zero, sem usar nenhuma linha do **Symbolics Emacs**.

SOFTWARE LIVRE

Inconformado, **Stallman** a **reescreveu** o **Emacs** do zero, sem usar nenhuma linha do **Symbolics Emacs**.

Diante do ocorrido houve uma **disputa legal**.
(a disputa terminou em **1992**, com a **Symbolics** concordando em renunciar a qualquer direito de propriedade sobre o **Emacs**.)

SOFTWARE LIVRE

Inconformado, **Stallman** a **reescreveu** o **Emacs** do zero, sem usar nenhuma linha do **Symbolics Emacs**.

Diante do ocorrido houve uma **disputa legal**.
(a disputa terminou em **1992**, com a **Symbolics** concordando em renunciar a qualquer direito de propriedade sobre o **Emacs**.)

Além de **Stallman**, vários outros desenvolvedores também estavam passando por problemas semelhantes aos dele.

SOFTWARE LIVRE

Inconformado, **Stallman** a **reescreveu** o **Emacs** do zero, sem usar nenhuma linha do **Symbolics Emacs**.

Diante do ocorrido houve uma **disputa legal**.
(a disputa terminou em **1992**, com a **Symbolics** concordando em renunciar a qualquer direito de propriedade sobre o **Emacs**.)

Além de **Stallman**, vários outros desenvolvedores também estavam passando por problemas semelhantes aos dele.

Outro problema era que todos os sabores do **Unix** disponível se tornaram **Proprietários**. (o único livre era **BSD** da **Universidade de Berkeley**, mas estava em disputa em um longo processo judicial por direitos autorais do **Unix**, logo **não disponível**);

SOFTWARE LIVRE

Inconformado, **Stallman** a **reescreveu** o **Emacs** do zero, sem usar nenhuma linha do **Symbolics Emacs**.

Diante do ocorrido houve uma **disputa legal**.
(a disputa terminou em **1992**, com a **Symbolics** concordando em renunciar a qualquer direito de propriedade sobre o **Emacs**.)

Além de **Stallman**, vários outros desenvolvedores também estavam passando por problemas semelhantes aos dele.

Outro problema era que todos os sabores do **Unix** disponível se tornaram **Proprietários**. (o único livre era **BSD** da **Universidade de Berkeley**, mas estava em disputa em um longo processo judicial por direitos autorais do **Unix**, logo **não disponível**);

Em **setembro de 1983 Stallman** decide criar um projeto...

Nasce o Projeto **GNU**!!!

Nasce o Projeto **GNU**!!!

GNU? Porque **GNU**?

Nasce o Projeto **GNU**!!!

GNU? Porque **GNU**?

GNU é um nome acrônimo redundante

Nasce o Projeto **GNU**!!!

GNU? Porque **GNU**?

GNU é um nome acrônimo redundante

GNU é **GNU** is **N**ot **U**nix

O **GNU** surgiu para ser um **Sistema Operacional Completo**!

O **GNU** surgiu para ser um **Sistema Operacional Completo**!

O **GNU** teria como principais características.

O **GNU** surgiu para ser um **Sistema Operacional Completo**!

O **GNU** teria como principais características.

- **Multiplataforma**, funcionar em qualquer hardware.

O **GNU** surgiu para ser um **Sistema Operacional Completo**!

O **GNU** teria como principais características.

- **Multiplataforma**, funcionar em qualquer hardware.

- **Multitarefas**, executar várias tarefas ao mesmo tempo.

O **GNU** surgiu para ser um **Sistema Operacional Completo**!

O **GNU** teria como principais características.

- **Multiplataforma**, funcionar em qualquer hardware.

- **Multitarefas**, executar várias tarefas ao mesmo tempo.

- **Multiusuários**, vários usuários usando ao mesmo tempo!

O **GNU** surgiu para ser um **Sistema Operacional Completo**!

O **GNU** teria como principais características.

- **Multiplataforma**, funcionar em qualquer hardware.

- **Multitarefas**, executar várias tarefas ao mesmo tempo.

- **Multiusuários**, vários usuários usando ao mesmo tempo!

Apesar de ele não ser o **Unix**, seria um **Unix-like**, por seguir os preceitos do **Unix**.

O **GNU** surgiu para ser um **Sistema Operacional Completo**!

O Projeto GNU consiste em inúmeros pequenos subprojetos mantidos por voluntários.

O **GNU** surgiu para ser um **Sistema Operacional Completo**!

O Projeto GNU consiste em inúmeros pequenos subprojetos mantidos por voluntários.

Para **inovar** perante ao **Unix** que possuía um kernel tido como simples (**monolítico**), iniciou-se o desenvolvimento do **Hurd**, um projeto constituído de uma arquitetura de **microkernel** , que teria **muitos ganhos** sobre o kernel **monolítico**.

O **GNU** surgiu para ser um **Sistema Operacional Completo**!

O Projeto GNU consiste em inúmeros pequenos subprojetos mantidos por voluntários.

Para **inovar** perante ao **Unix** que possuía um kernel tido como simples (**monolítico**), iniciou-se o desenvolvimento do **Hurd**, um projeto constituído de uma arquitetura de **microkernel** , que teria **muitos ganhos** sobre o kernel **monolítico**.

Mas enquanto o **Hurd** era desenvolvido, os demais subprojetos iam de vento em poupa e como eram compatíveis com o **Unix**, podiam ir sendo executados nos **Unix** disponíveis.

FREE SOFTWARE
F O U N D A T I O N

Em **outubro de 1985 Stallman** e demais membros do Projeto GNU criaram a **Fundação do Software Livre** com o intuito de atuar na proteção dos Software Livres.

FREE SOFTWARE
FOUNDATION

A **FSF** nasceu para proteger o Projeto GNU e para difundir o Software Livre!

FREE SOFTWARE FOUNDATION

A **FSF** nasceu para proteger o Projeto GNU e para difundir o Software Livre!

Para proteger as centenas de softwares livres do **GNU** ela criou um conjunto de **licenças**. Sendo as principais:

FREE SOFTWARE FOUNDATION

A **FSF** nasceu para proteger o Projeto GNU e para difundir o Software Livre!

Para proteger as centenas de softwares livres do **GNU** ela criou um conjunto de **licenças**. Sendo a principal:

GNU General Public License (GPL)

GNU General Public License (GPL)

A **GPL** é a licença mais adotada por Softwares Livres! Ela prega as 4 liberdades?

GNU General Public License (GPL)

A **GPL** é a licença mais adotada por Softwares Livres! Ela prega as 4 liberdades?

A liberdade de executar o programa, para qualquer propósito (liberdade nº 0)

GNU General Public License (GPL)

A **GPL** é a licença mais adotada por Softwares Livres! Ela prega as 4 liberdades?

A liberdade de executar o programa, para qualquer propósito (liberdade nº 0)

A liberdade de estudar como o programa funciona e adaptá-lo às suas necessidades (liberdade nº 1).

GNU General Public License (GPL)

A **GPL** é a licença mais adotada por Softwares Livres! Ela prega as 4 liberdades?

A liberdade de executar o programa, para qualquer propósito (liberdade nº 0)

A liberdade de estudar como o programa funciona e adaptá-lo às suas necessidades (liberdade nº 1).

A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo (liberdade nº 2).

GNU General Public License (GPL)

A **GPL** é a licença mais adotada por Softwares Livres! Ela prega as 4 liberdades?

A liberdade de executar o programa, para qualquer propósito (liberdade nº 0)

A liberdade de estudar como o programa funciona e adaptá-lo às suas necessidades (liberdade nº 1).

A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo (liberdade nº 2).

A liberdade de aperfeiçoar o programa e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie deles (liberdade nº 3). O acesso ao código fonte é um pré-requisito para esta liberdade.

O **GNU** surgiu para ser um **Sistema Operacional Completo**!

No final da década de **1980** o **Projeto GNU** já possuía **milhares** de Softwares, Bibliotecas, Drivers, etc. Todos sendo executados nos **Unix** existentes.

Faltava apenas o **Kernel** para que o Sistema Operacional ficasse completo!

Mas seu **Kernel Hurd** ainda **não está pronto!** Transformando o **GNU** em um **corpo sem coração!**

Stallman chegou a sugerir a adoção do **Mach** da **Universidade Carnegie Mellon** (CMU) mas não foi liberado pela **CMU**.

Linus Benedict **Torvalds**

28/12/1969

Helsinque, Finlândia.

Formado em Ciência da Computação Universidade Helsinque entre **1988** a **1996.**

Em 25 de agosto de 1991, Linus enviou um e-mail para a lista de discussão "comp.os.minix" anunciando seu projeto de criar um novo kernel...

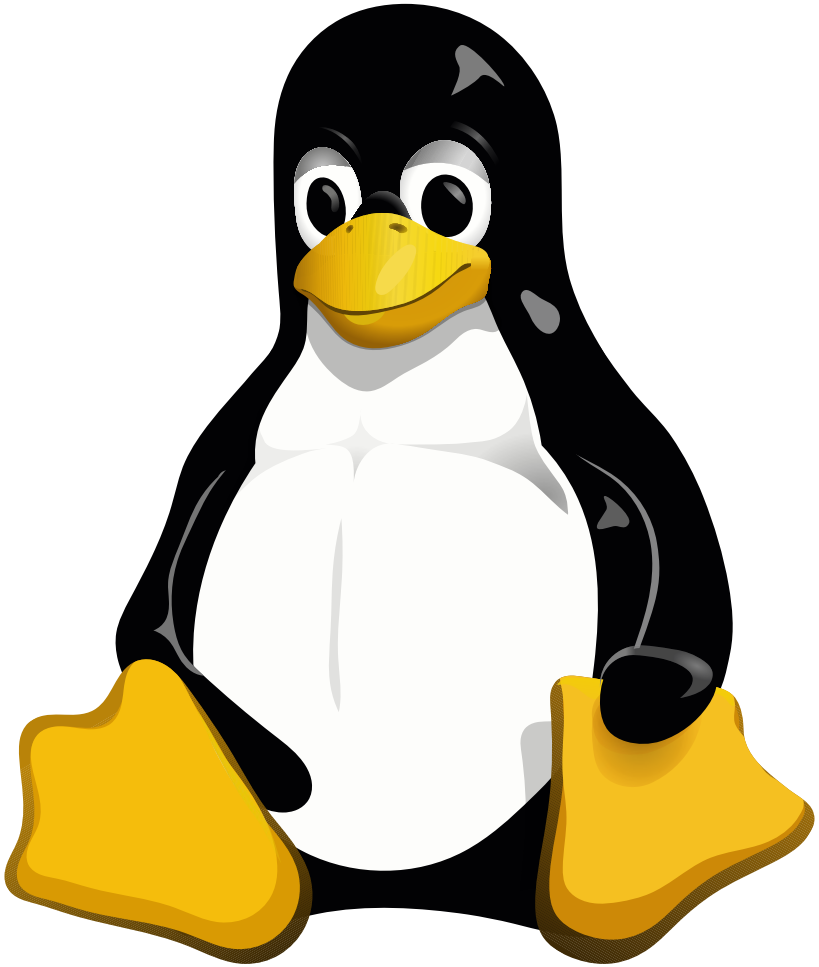
De: torvalds@klaava.Helsinki.FI (**Linus Benedict Torvalds**)
Assunto: O que vocês acham de um Sistema Operacional Novo e Livre?
Newsgroups: comp.os.minix
Data: **1991-08-25** 20:57:08 GMT

Olá a todos os interessados em minix!

Estou fazendo um sistema operacional (gratuito, apenas um hobby, não será grande e profissional como o GNU) para clones AT 386(486). Isso já está há algum tempo em desenvolvimento (desde abril) e está começando a ficar pronto. Gostaria de receber comentários sobre o que as pessoas acham do minix, do que gostariam que fosse diferente, etc, e sugestões de recursos.

Atualmente, o sistema operacional está em uma fase em que não pode ser distribuído (já que não está completo e nem compatível com o minix), mas estou postando aqui para ver se alguém está interessado em ajudar no desenvolvimento. Qualquer sugestão é bem-vinda, mas não prometo implementar nada :-)

Linus (torvalds@kruuna.helsinki.fi)



LINUX

O Kernel LINUX

Torvalds, pretendia criar algo simples e gratuito

O Kernel LINUX

Torvalds, pretendia criar algo simples e gratuito

Diferente de **Stallman** que criou muita coisa sozinho, **Torvalds** criou um embrião e chamou as comunidades para participar da criação.

O Kernel LINUX

Torvalds, pretendia criar algo simples e gratuito

Diferente de **Stallman** que criou muita coisa sozinho, **Torvalds** criou um embrião e chamou as comunidades para participar da criação.

O **Unix do Linux** como inicialmente era chamado, depois se tornando **Linux**, é um **kernel monolítico** e não **microkernel** como o **Hurd**, por ser simples e não querer reinventar ficou pronto rapidamente.

O Kernel LINUX

Torvalds, pretendia criar algo simples e gratuito

Diferente de **Stallman** que criou muita coisa sozinho, **Torvalds** criou um embrião e chamou as comunidades para participar da criação.

O **Unix do Linux** como inicialmente era chamado, depois se tornando **Linux**, é um **kernel monolítico** e não **microkernel** como o **Hurd**, por ser simples e não querer reinventar ficou pronto rapidamente.

Torvalds adotou como licença a **GNU GPL** para seu kernel

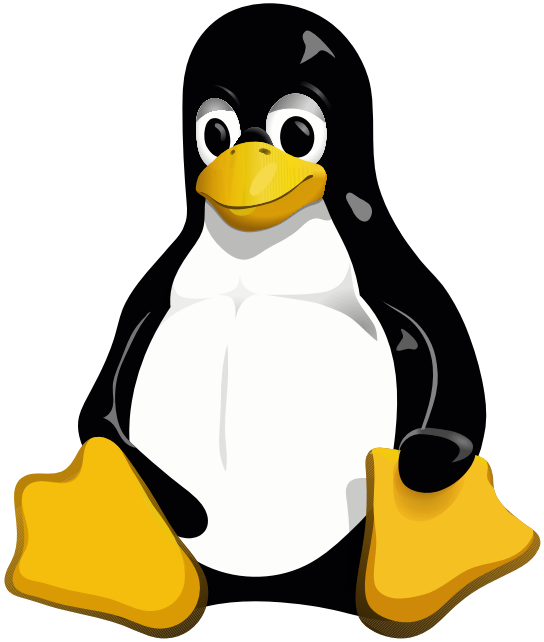
O Kernel LINUX

CURIOSIDADE:

Em **1996**, quando o **kernel Linux** estava se tornando cada vez mais popular e estava começando a ser usado por empresas e organizações em todo o mundo. Na época, **Torvalds** decidiu que o kernel precisava de um **mascote**.

Ele pediu sugestões para a comunidade Linux e acabou escolhendo um desenho enviado por **Larry Ewing**, O desenho era de um pinguim fofinho e gorducho, com uma expressão amigável no rosto.

O nome "**Tux**" foi sugerido por **James Hughes**, e é uma referência a "**Tuxedo**", um tipo de roupa formal que os pinguins usam. Desde então, o **Tux** se tornou um símbolo icônico do **Linux**.



GNU/LINUX

Como o **kernel Linux** era compatível com o **GNU** e adotava a mesma licença. A união foi rápida e natural.

GNU/LINUX

Como o **kernel Linux** era compatível com o **GNU** e adotava a mesma licença. A união foi rápida e natural.

Da parte do **GNU** inicialmente a adoção do **kernel Linux** foi uma medida temporária para se ter um sistema operacional até que o **Hurd** ficasse pronto.

GNU/LINUX

Como o **kernel Linux** era compatível com o **GNU** e adotava a mesma licença. A união foi rápida e natural.

Da parte do **GNU** inicialmente a adoção do **kernel Linux** foi uma medida temporária para se ter um sistema operacional até que o **Hurd** ficasse pronto.

Desta forma surgiu o **Sistema Operancional GNU/Linux**, corpo **GNU** e coração **Linux**!

Distribuições GNU/LINUX

No começo do GNU/Linux, era uma verdadeira odisseia ter um instalado, pois cada software era instalado individualmente, ou seja tinha que se baixar na internet programa por programa, descompactar, compilar e por final instalar.

Distribuições GNU/LINUX

No começo do GNU/Linux, era uma verdadeira odisseia ter um instalado, pois cada software era instalado individualmente, ou seja tinha que se baixar na internet programa por programa, descompactar, compilar e por final instalar.

Para facilitar esse processo em março de **1992** surgiu a **primeira distribuição** a **MCC Interim Linux**, criada pela Universidade de Manchester.

Distribuições GNU/LINUX

Depois da MCC vieram:

Slackware, lançada em 16 de julho de 1993.

Debian, lançada em 16 de agosto de 1993.

SuSE, lançada em 16 de outubro de 1994.

Red Hat, lançada em 3 de novembro de 1994.

Conectiva, lançada em 28 de abril de 1995.

Mandrake, lançada em 7 de julho de 1998.

Arch, lançada em 11 de março de 2002.

Gentoo, lançada em 31 de março de 2002.

Ubuntu, lançada em 20 de março de 2004.

CÓDIGO ABERTO

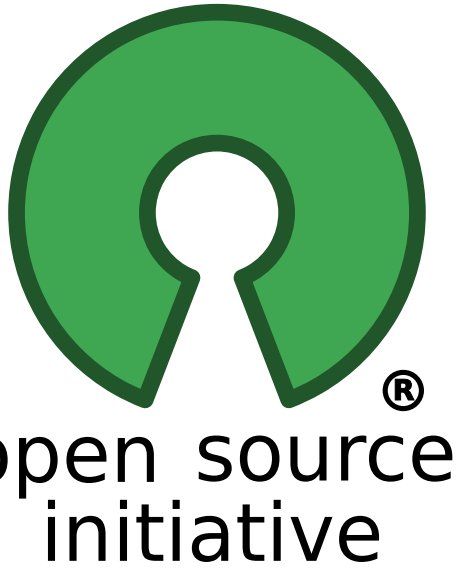
Eric Steven **Raymond**

4/12/1957

Boston, Massachusetts - Estados Unidos.

Programador, escritor e ativista do movimento de software livre e de código aberto (FOSS).

Autor de vários livros importantes na área de software livre e de código aberto, incluindo **"A Catedral e o Bazar"** (**The Cathedral and the Bazaar**), que é considerado um dos textos mais influentes sobre o desenvolvimento colaborativo de software.


®
open source
initiative

Open Source Iniciative

Open Source Iniciative

Fundada em fevereiro de **1998**, por **Bruce Perens**, **Jon “Maddog” Hall**, **Larry Augustin**, **Eric S. Raymond** entre outros.

Open Source Iniciative

Fundada em fevereiro de **1998**, por **Bruce Perens**, **Jon "Maddog" Hall**, **Larry Augustin**, **Eric S. Raymond** entre outros.

A **Open Source Initiative** (**OSI**) - Iniciativa pelo código aberto - é uma organização dedicada a promover o software de código aberto.

Open Source Iniciative

Fundada em fevereiro de **1998**, por **Bruce Perens**, **Jon "Maddog" Hall**, **Larry Augustin**, **Eric S. Raymond** entre outros.

A **Open Source Initiative** (**OSI**) - Iniciativa pelo código aberto - é uma organização dedicada a promover o software de código aberto.

Ela foi criada para **incentivar uma aproximação** de **empresas** com o **Software Livre**. Pois o mercado na época considerava o Software Livre muito radical.

Open Source Iniciative

Todo **Software Livre** é um software de **código aberto**!

Open Source Iniciative

Todo **Software Livre** é um software de **código aberto**!

Mas nem todos os softwares de **código aberto** são **Software Livre**!

Open Source Iniciative

Todo **Software Livre** é um software de **código aberto**!

Mas nem todos os softwares de **código aberto** são **Software Livre**!

Para ser considerado **Software Livre** é preciso adotar licenças que garantam as **4 Liberdades**. Mas a **OSI** aceita um grande números de **licenças que não garantem** as **4 Liberdades**.

Obrigado!

Esta palestra está disponível para download em
https://martinelli.eti.br/flisol/