

GNU DEBIAN



Por Francisco Anderson,
Marcelo Bulhões e
Vitor Silva



Introdução

Propósito do Debian

Requisitos de aplicação





**Distribuição do sistema
operacional GNU.**

**O Advanced Packaging
Tool (APT).**

Fundador

Ian Murdock(1973-2015)

Propósito do Debian

SER COMPLETAMENTE LIVRE!

- Rodar o software da forma que quiser para qualquer propósito.
- Estudar como o programa funciona, e poder editá-lo.
- Redistribuir cópias.
- Distribuir suas modificações para outras pessoas.



CASO

Firefox

X

Iceweasel

Requisitos de aplicação



Source package.



Sistema de empacotamento



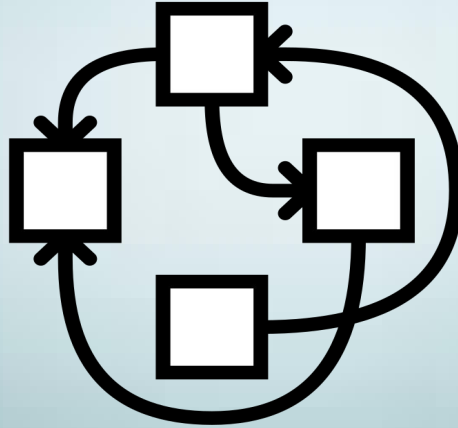
Pacotes de código fonte



Geração de pacotes com compilados



Seções Contrib e Non-free de repositório



Kernels Utilizados

Introdução

1.Arquitetura

2.Gerência de memória

3.Gerência de processos

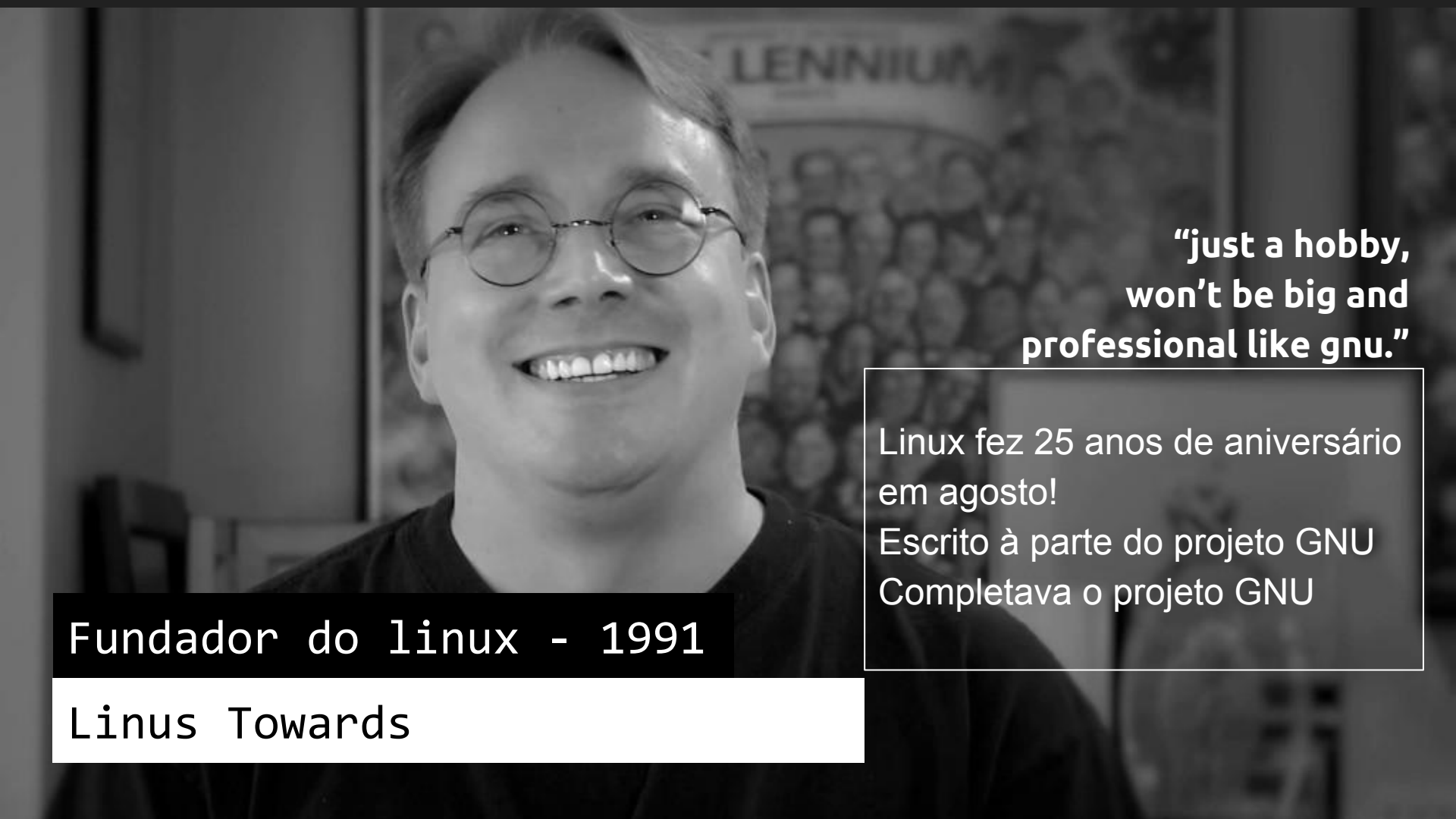
4.Sistemas de arquivos

5.Gerência de interrupção

6.Suporte a threads

7.Segurança





**“just a hobby,
won’t be big and
professional like gnu.”**

Linux fez 25 anos de aniversário
em agosto!

Escrito à parte do projeto GNU
Completava o projeto GNU

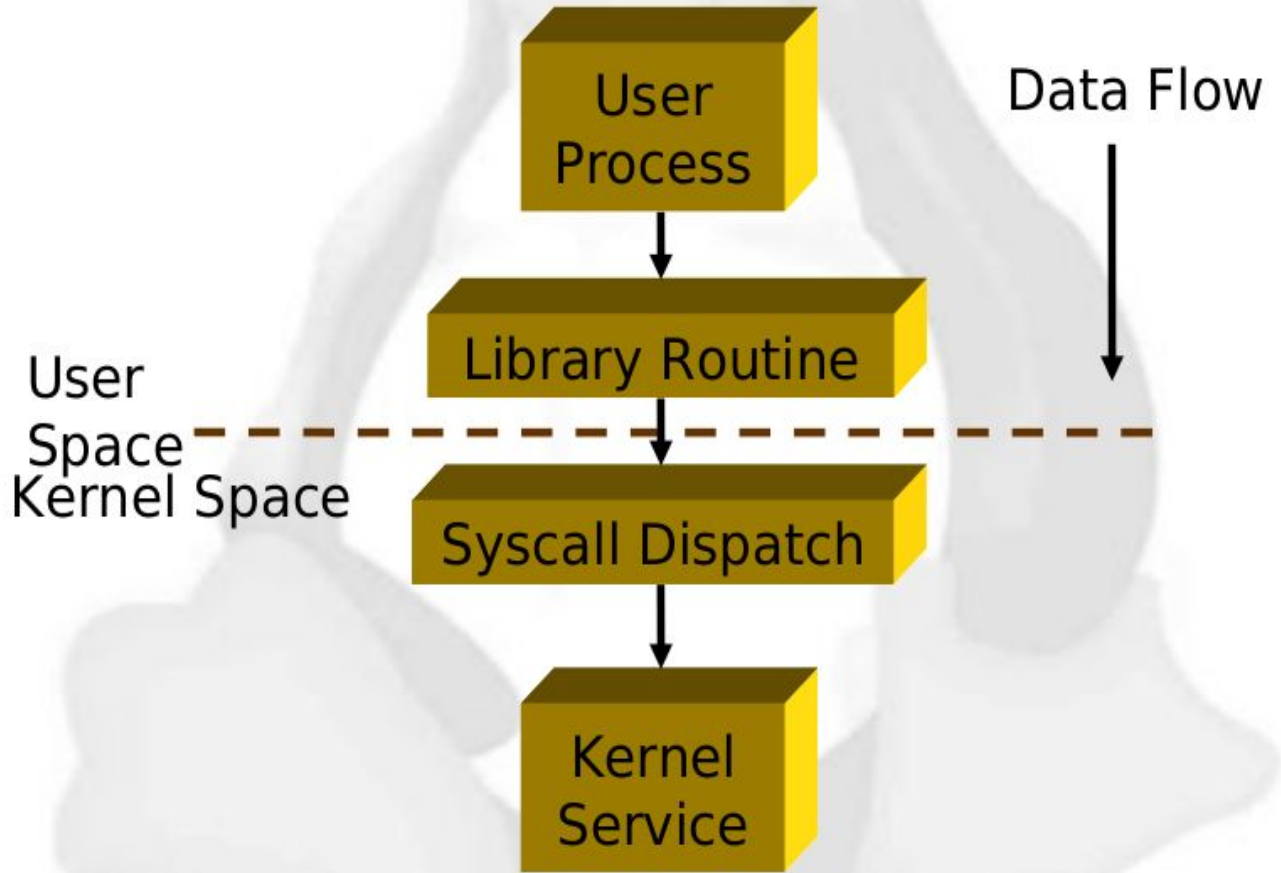
Fundador do linux - 1991

Linus Towards

Linux

- 1.P.U. acessa lib
- 2.libs chama syscall
- 3.Syscall chama kernel
- 4.Kernel atende

ARQUITETURA





Gerência de memória

Segmentação-paginação

Alpha AXP

Memória de usuário e
memória do kernel
independentes em
endereços separados



Gerência de memória

LRU(Least Recently
Used).

Lista de espaços vazios
com coalescência(?)

Escalonador $O(1)$

Suporte a SMP

Array de tarefas ativas e o de expiradas.

Cada array com uma fila de processos por nível de prioridade.

Mapas de bits para descobrir eficientemente a maior prioridade entre as tarefas prontas.

Escalonador $O(1)$

As tarefas são escalonadas com Round Robin.

Trocar os arrays é tão rápido quanto trocar dois ponteiros.

Grande massa de código necessária para calcular heurísticas tornou-se difícil de manejar.

Escalonador CFS

Baseado RSDL(Rotating Staircase Deadline Scheduler)

Justiça

politica de justiça processos bloqueados(sleeper fairness)

CFS mantem uma árvore "red-black" ordenada por tempo

Escalonador BFS

Evita eurísticas

Sem grandes avanços

Sistemas de Arquivos

suportados no linux

ext2

ext3

ext4

btrfs

xf


FAT 16/32

Gerenciamento de Interrupções

Vetor de interrupções

Driver conhece posição da interrupção

Dispositivo informa conclusão da operação




Semelhante a processos

Recursos de um mesmo processo

Escalonamento

Suporte a Threads



Software de criptografia integrado com a distribuição principal.

Ferramentas de suporte a segurança

Versionamento(unstable, a testing e a stable).

Segurança



FreeBSD[®]

Kernel

1. Arquitetura
2. Gerência de memória
3. Gerência de arquivo
4. Gerência de E/S
5. Gerência de processos
6. Suporte a threads
7. Interrupções
8. Segurança





BSD

Berkeley Software Distribution

kFreeBSD

**Possui kernel monolítico,
E sua licença é considerada livre,
Porque não colocar no GNU Debian?**

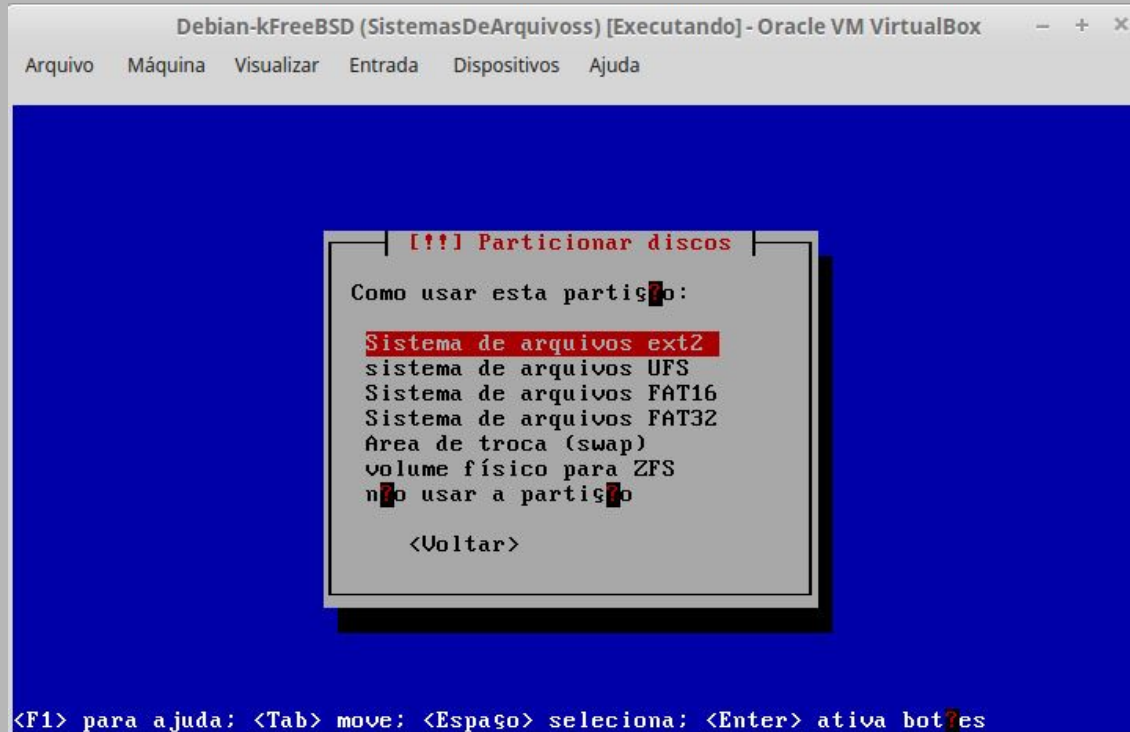


Gerência de memória

LRU(Least Recently Used)



Gerência de arquivos



O Debian oferece suporte para diversos sistemas de arquivos

Gerência de arquivos

ext2 /3* /4* (standard Linux filesystem)

UFS

FAT 16

FAT 32

Gerência de arquivos

	Minix FS	Ext FS	Ext2 FS	Xia FS
Max FS size	64 MB	2 GB	4 TB	2 GB
Max file size	64 MB	2 GB	2 GB	64 MB
Max file name	16/30 c	255 c	255 c	248 c
3 times support	No	No	Yes	Yes
Extensible	No	No	Yes	No
Var. block size	No	No	Yes	No
Maintained	Yes	No	Yes	?

4. Gerência de E/S

Escalonamento do elevador.

Organiza a fila em ordem de crescente offset de E/S.

Maximizar a transferência aumentando o número de E/S em uma busca no disco,

Outras políticas

Gerência de Processos

Escalonador ULE

timeslice e fila de execução

Favorecimento de processos interativos.

Suporte à aplicações de tempo real.

outros.

Suporte a Threads

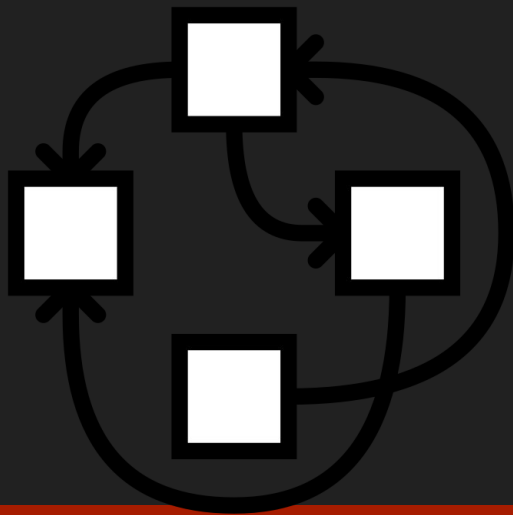
O FreeBSD tem suporte as threads POSIX.

Segurança

Modos de usuário

Grupos de usuário

Versões(Squeeze, Wheezy e Jessie)



Hurd

O que é

Arquitetura

Informações

Sobre

Aquele que deveria ter sido o kernel do GNU

Conceitualmente melhor

Implementação complexa

Bom o bastante se sobrepõe sobre o melhor

Arquitetura

Cliente-Servidor

Microkernel

GNU/Mach

L4

Coyotos

Viengoos

Gerencia E/S

Portas de comunicação

Mecanismo de tradução

No momento só tem suporte para ext2fs

Suporte a Threads

Agressivo

Inicialmente usando cthreads

No momento possui algum suporte para POSIX Threads

Segurança

Microkernel: mínimo de código executado no modo núcleo.

Ainda não possui versão estável.

Falta de desenvolvedores.

GNU
DEBIAN



FIM