

Sistema de Arquivos UNIX

Taisy Silva Weber

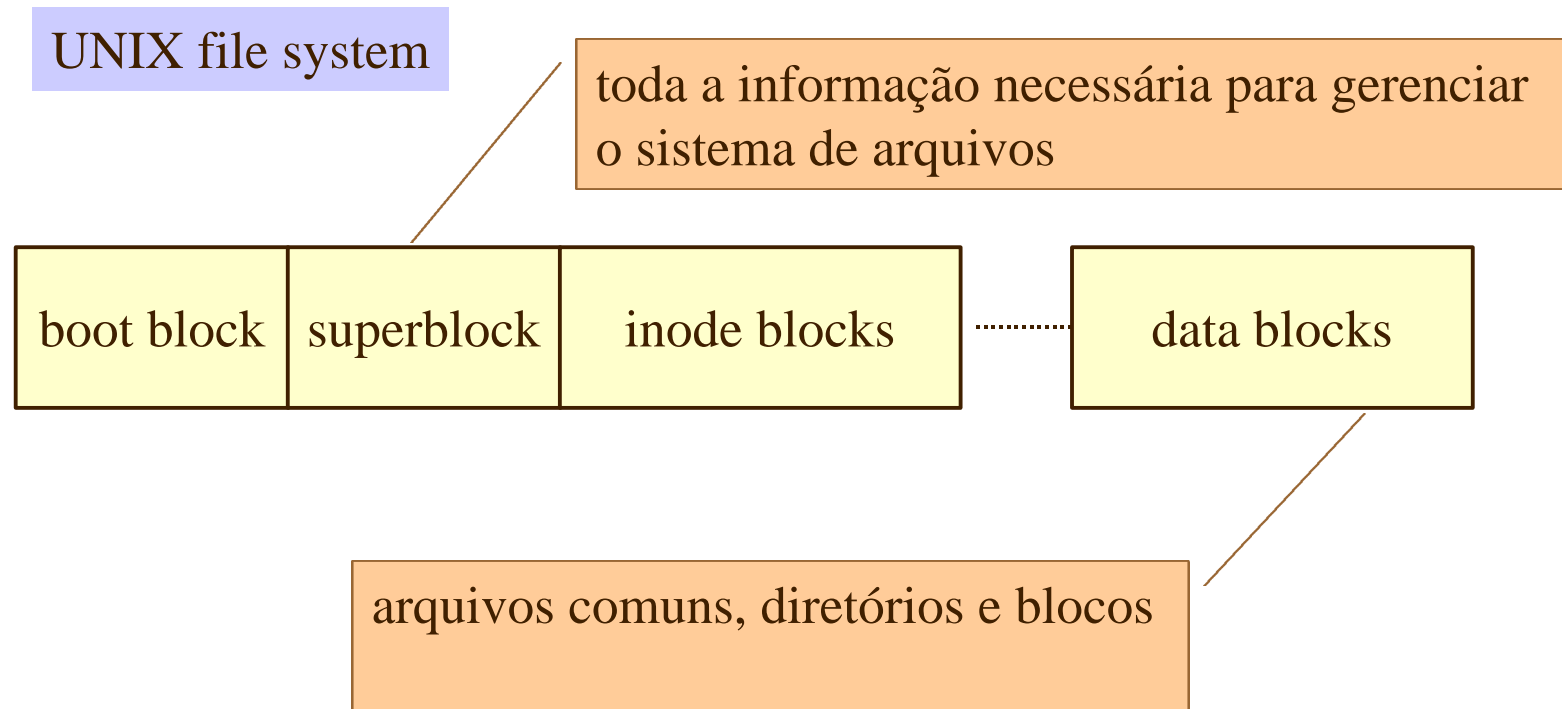
UFRGS

Trabalhando com arquivos

- ✓ Introdução a sistemas de arquivo Linux
 - ✓ conceitos básicos
 - ✓ Sistema de arquivo ext2fs
 - ✓ Virtual File System
 - ✓ /proc

LDP

Estrutura esquemática



Implementações

✓ várias modificações de versão para versão

interface para o FS é simples e bem definida

implementação pode ser alterada sem reflexos significativos para o usuário

Berkeley Fast File System foi um sucesso

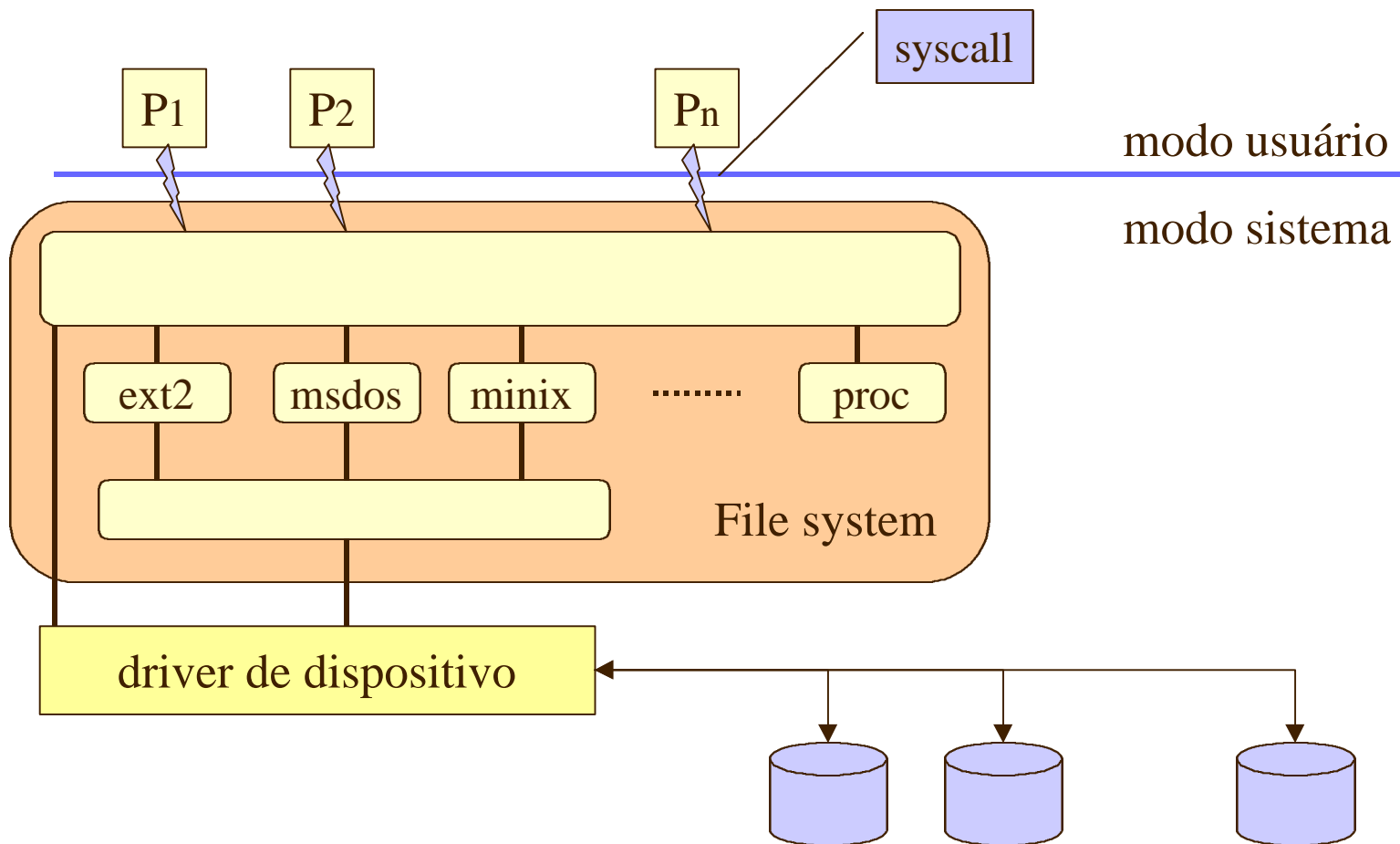
alto desempenho

bloco 8K, fragmento 1K

incorporou várias novas system calls (como truncate)

ext2fs (Linux) inspirou-se no ffs

Camadas do FS em Linux



Virtual file system

Chris Provenzano

✓ projetado segundo orientação a objetos

- um conjunto de definições que determinam como o objeto-arquivo se parece
- uma camada de software para manipular os objetos

✓ 3 tipos principais de objetos

- inode-object } representam arquivos individuais
- file-object } usados para acessar os arquivos
- file-system object
representa um sistema de arquivos completo

Virtual file system

VFS

- para cada um dos 3 tipos de objetos, o VFS define um conjunto de operações que deve ser

cada objeto (de um dos 3 tipos) contém um *pointer* para uma **tabela de funções**

- que implementam as operações para o objeto

assim o VFS pode operar sobre objetos sem conhecer **espécie de objeto** ele está tratando

arquivo de rede, arquivo em disco, socket, diretório

file-system object

VFS

- ✓ uma coleção de arquivos que forma uma hierarquia de diretório auto-contida
 - Linux mantém um único **file-system object** para cada dispositivo de disco montado como um FS e para FS de rede conectado
- ✓ objeto FS permite acesso ao inodes
 - VFS identifica cada **inode** por um par unívoco
par: file-system / inode-number
 - VFS acha um **inode** fornecendo esse par ao FS-object

inode-object & file-object

✓ inode-object representa um arquivo

não pertence a um processo

mantém informações como
tamanho, data da última alteração

✓ file-object pertence a um único processo

representa um ponto de acesso aos dados no arquivo
onde um processo está atualmente lendo ou escrevendo

guarda se o processo pediu permissão de escrita

Ext2fs

Remy Card, Wayne Davidson,
Theodore Ts'o, Stephen Tweedie

✓ extended file system

- extensão do Minix
- ✓ semelhante ao BSD Fast File System (ffs)
 - principal diferença: política de alocação de disco

ffs usa blocos de 8Kbytes e **fragmentos** de 1Kbytes

ext2fs não usa fragmento

ext2fs usa blocos de 1Kbytes (default) mas também suporta 2K e 4K

ext2fs aumenta o desempenho alocando blocos lógicos adjacentes em posições físicas adjacentes no disco

Ext2fs

- ✓ suporta os tipos de arquivos UNIX
 - arquivos regulares, diretórios, arquivos especiais de dispositivos, links simbólicos
- ✓ gerencia FS criados em partições realmente **4TB**
- ✓ suporta nomes longos de arquivos
 - 255 caracteres (podendo estender até 1012)
 - entradas de diretório de tamanho variável
- ✓ reserva alguns blocos para super usuário (
 - normalmente 5%

Estrutura

✓ FS particionado em múltiplos **block groups**

- conceito similar ao **cylinder group** no ffs

no ffs cada grupo corresponde a um cilindro físico

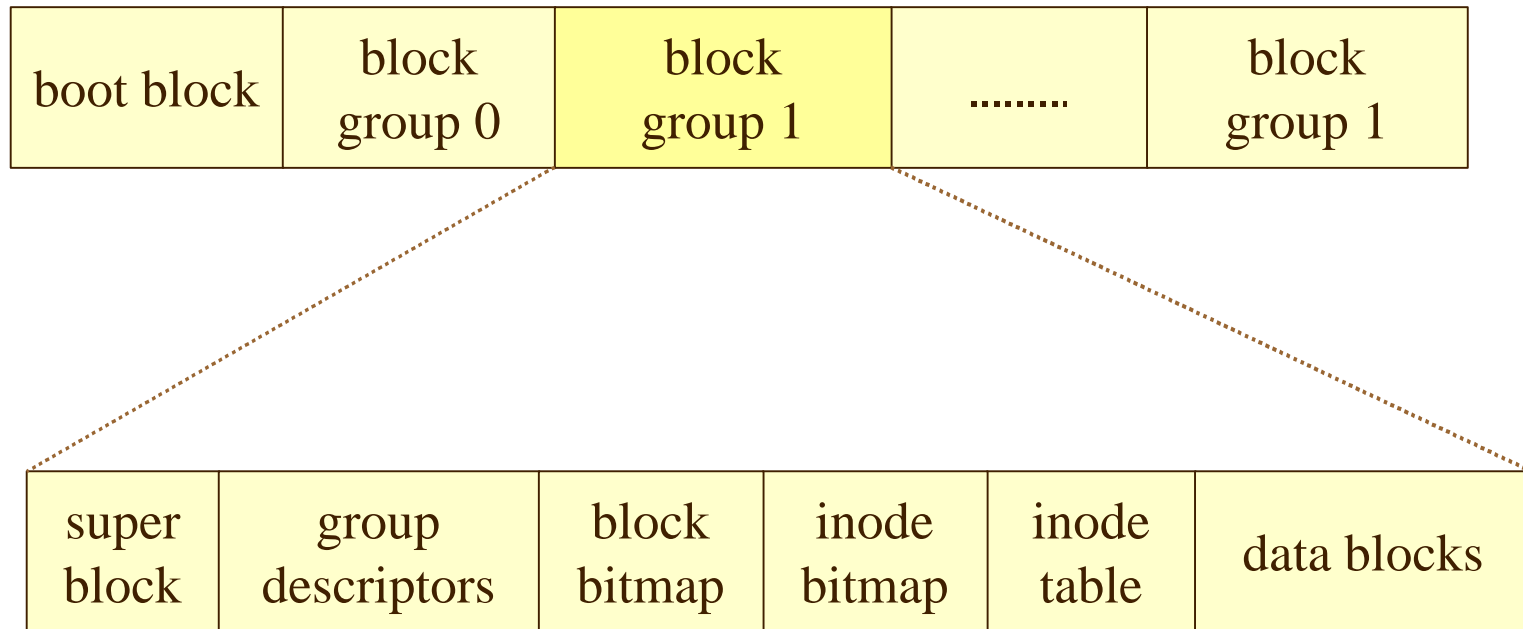
atualmente os cilindros tem tamanho variável

- cada block group contém uma cópia do superblock, inode e data blocks

✓ block groups visam: manter o grupo fisicamente contínuo

- manter blocos de dados perto de seus
de arquivos perto dos seus

Estrutura do ext2fs



Superbloco

- ✓ contém informações de controle do FS
- ✓ possui tamanho de 1024 bytes
 - 1024 bytes é o tamanho mínimo de bloco
o tamanho é alcançado com inserção de *pad bytes*
- ✓ todos os grupos contém um superblock
 - ✓ facilita a recuperação do sistema de arquivos

não encontrei nada sobre como é mantida a consistência entre os

Estrutura do superbloco

	0	3	4	7
0	número de inodes		número de blocos	
8	número de blocos reservados		número de blocos livres	
			primeiro bloco de dados	
	tamanho de bloco		tamanho de fragmento	
	blocos por grupo		fragmentos por grupo	
	inodes por grupo		horário da montagem	
	horário do último write no sb		estado	max. contador
	assinatura	estado	comport. erro	pad word
	horário do último test		max. intervalo de teste	
	sistema operacional		FS revisão	
80	RESUID	RESGID		

Superbloco: campos

✓ tamanho do bloco

deslocar o **tamanho mínimo**
pelo valor desse campo

- fornecido indiretamente

tamanho mínimo é 1024 Kbytes

o administrador pode escolher o tamanho de bloco na criação do FS

✓ max. contador

em modo read/write

- número máximo de mounts até o próximo teste

✓ RESUID e RESGID

- especificam que *users* e *groups* tem permissão de usar os blocos reservados além do

Superbloco: campos

✓ estado (status)

- clean

quando o FS é remontado em modo

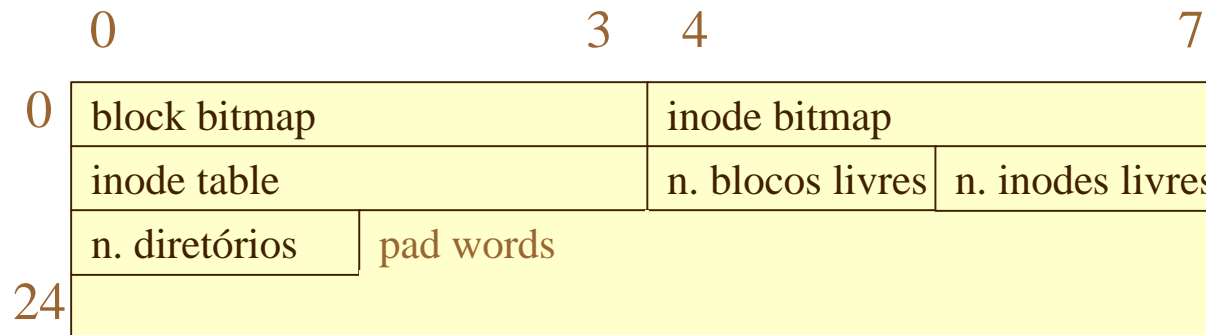
- not clean

quando o FS é montado em modo

- erroneous

quando uma inconsistência é

Descritor de grupo



✓ contém informações sobre o grupo

- aparece logo após o superbloco
- tamanho do descritor de grupo: 32 bytes
- n. de diretórios
 - política de espalhar diretórios em igual número

um novo diretório será montado no grupo com menor número de diretórios

bitmaps

block bitmap e inode bitmap

- cada **bitmap** tem o tamanho de um bloco
- tamanho do grupo fica restrito a 8192 blocos para blocos de 1024 bytes

um bit para cada bloco

inode table

número de blocos contendo inodes é dado no descritor de grupo

✓ lista de blocos consecutivos contendo

aparentemente ainda não implementado

- cada inode de 128 bytes
- em cada inode: *ACL - access control list*
 - para melhor diferenciar permissões de acesso

✓ fast symbolic link

- o nome é armazenado no próprio
 - economiza data blocks
 - acelera acesso (pois não precisa buscar um bloco)
- nome do link limitado em 60 caracteres

inode LINUX

	0	3	4	7
0	tipo/permisões	user (UID)	tamanho do arquivo	
8	horário de acesso		horário de criação	
	horário de modificação		horário de deleção	
	group (GID)	cont. links	número de blocos	
	atributos do arquivo		reservado	
	12 blocos diretos			
	blocos indireto simples		blocos indiretos duplos	
	blocos indiretos triplos		versão do arquivo	
	arquivo ACL		diretório ACL	
	endereço fragmento		reservado	
127				

ACL - access control list

Proc File System

FS sem dados persistentes

- ✓ process file system
 - ✓ file system cujo conteúdo não é guardado em
 - ✓ dados são calculados por demanda de acordo com requisições de E/S do usuário

não é original Linux

implementado pelo UNIX SVR4

Proc File System UNIX

- ✓ interface para suporte a depuração de processos
- ✓ cada subdiretório corresponde a um processo ativo
 - uma listagem do FS mostra um diretório por
 - o nome do diretório é a representação em ASCII do

Linux estende esse conceito

Fim ...

✓ de sistemas de arquivo Linux