

# Sistemas de Arquivos

## Sistemas de Arquivos

- Objetivos
  - Armazenar uma quantidade grande de informação
  - Sobreviver ao término do processo
  - Múltiplos processos devem poder acessar um dado
- Duas operações básicas
  - Leitura
  - Escrita

## Arquivos – Identificação

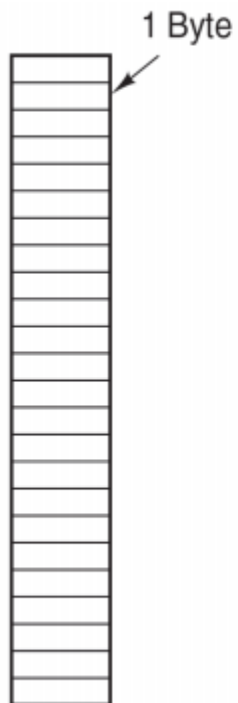
- É a característica mais importante para o usuário final
- Nome e extensão
- Extensão costuma ser apenas uma convenção
- No Windows pode relacionar um arquivo a um programa

## Arquivos – Estruturas

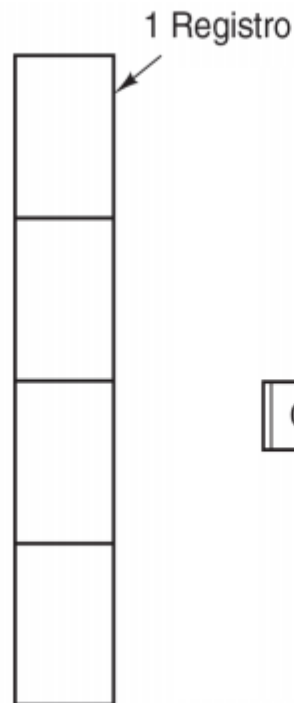
- Arquivos – Estruturas
  - Acesso e leituras byte a byte
- Arquivos em registros
  - Acesso e leituras em blocos de  $n$  bytes que compõem um registro
- Arquivos em listas ligadas
  - Acesso através de uma chave

### Arquivos - Estruturas

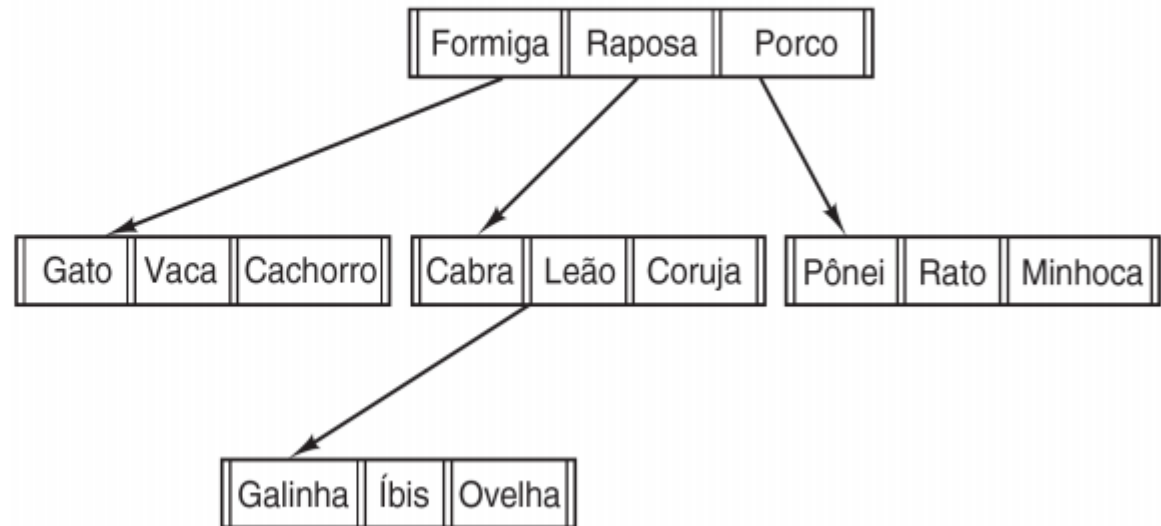
Três tipos de arquivos



(a) Sequência de bytes



(b) Sequência de registros



(c) Árvore

## Arquivos – Tipos

- Arquivos comuns
- Arquivos Especiais de bloco
- Arquivos Especiais de caractere
- O número mágico
  - Comando file no Linux

## Arquivos – Atributos

- Também chamados metadados
- Tempos de criação, alteração acesso
- Tamanho
- Arquivo oculto, backup, temporário, lock

## Arquivos – Operações

- Create
- Delete
- Open
- Close
- Read
- Write
- Append
- Seek
- Get Attributes
- Set Attributes
- Rename

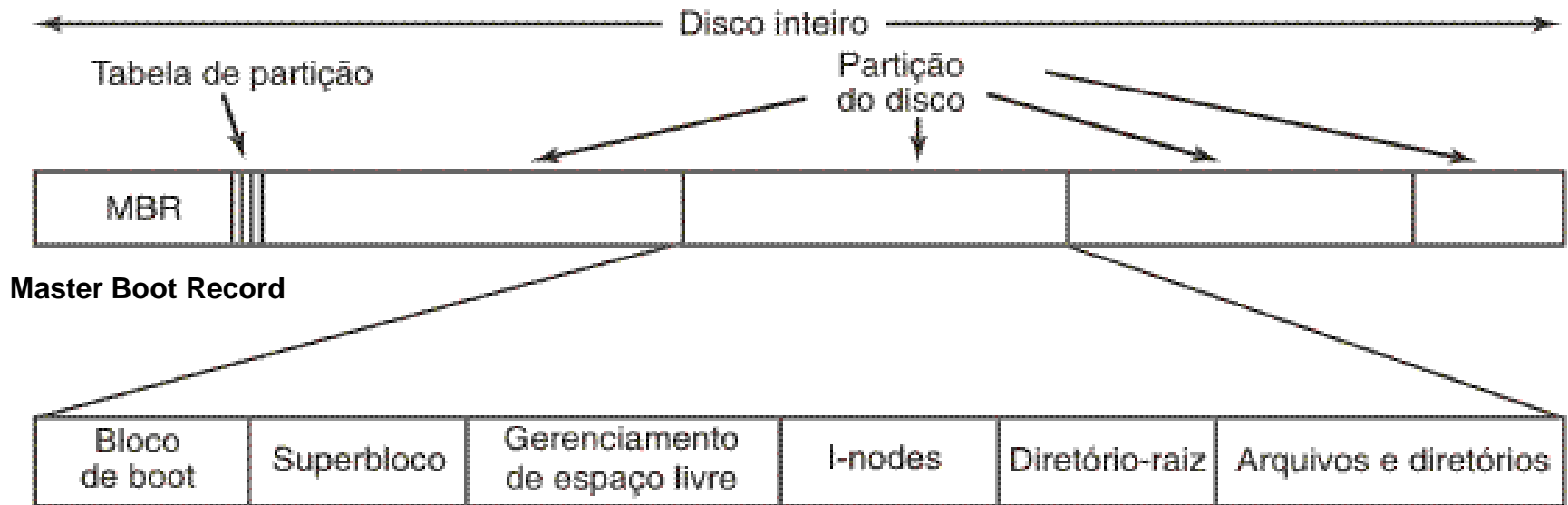


## Caminho de arquivos

- Caminho Absoluto
  - Ex.: /unip/CC/Aulas/Aula.ppt
- Caminho Relativo
  - Considera o diretório atual ou diretório de trabalho
  - Ex.:Aulas/Aula.ppt
- O diretório de trabalho é relativo ao processo
- As entradas . e ..
  - . Diretório Atual
  - .. Diretório Pai

### Layout do Sistema de Arquivos

- A tabela de partições indica o início de cada partição
- Dentro da partição o S.O. decide o layout

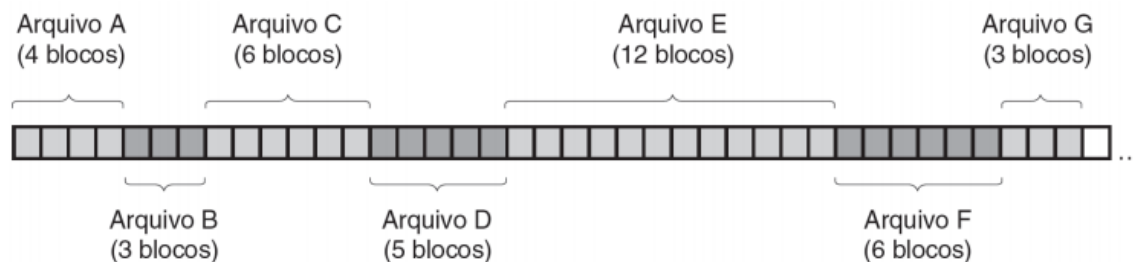


## Implementação de arquivos

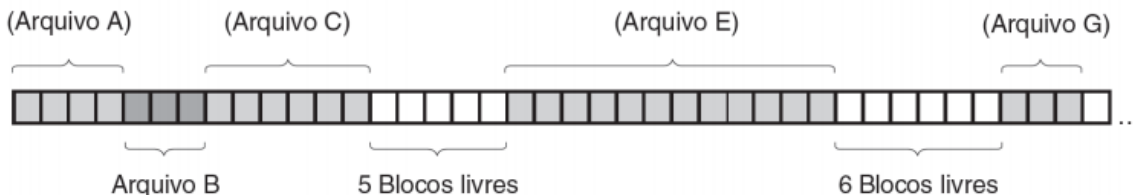
- O principal aspecto é:
  - Como indicar onde cada arquivo está
- Quatro abordagens
  - Alocação contigua
  - Alocação baseada em listas ligadas
  - Alocação baseada em listas ligadas usando uma tabela em memória
  - **I-Nodes** (Index Node) são a estrutura responsável por conter informações básicas (como permissões de acesso, identificação do utilizador, alterações, data e hora de criação e do último acesso realizado...) sobre os arquivos e pastas que estão guardados em qualquer dispositivo)

### Alocação contígua

- Ótima velocidade
- Fácil implementação
- Problema de fragmentação



A alocação contígua do espaço em disco para sete arquivos.

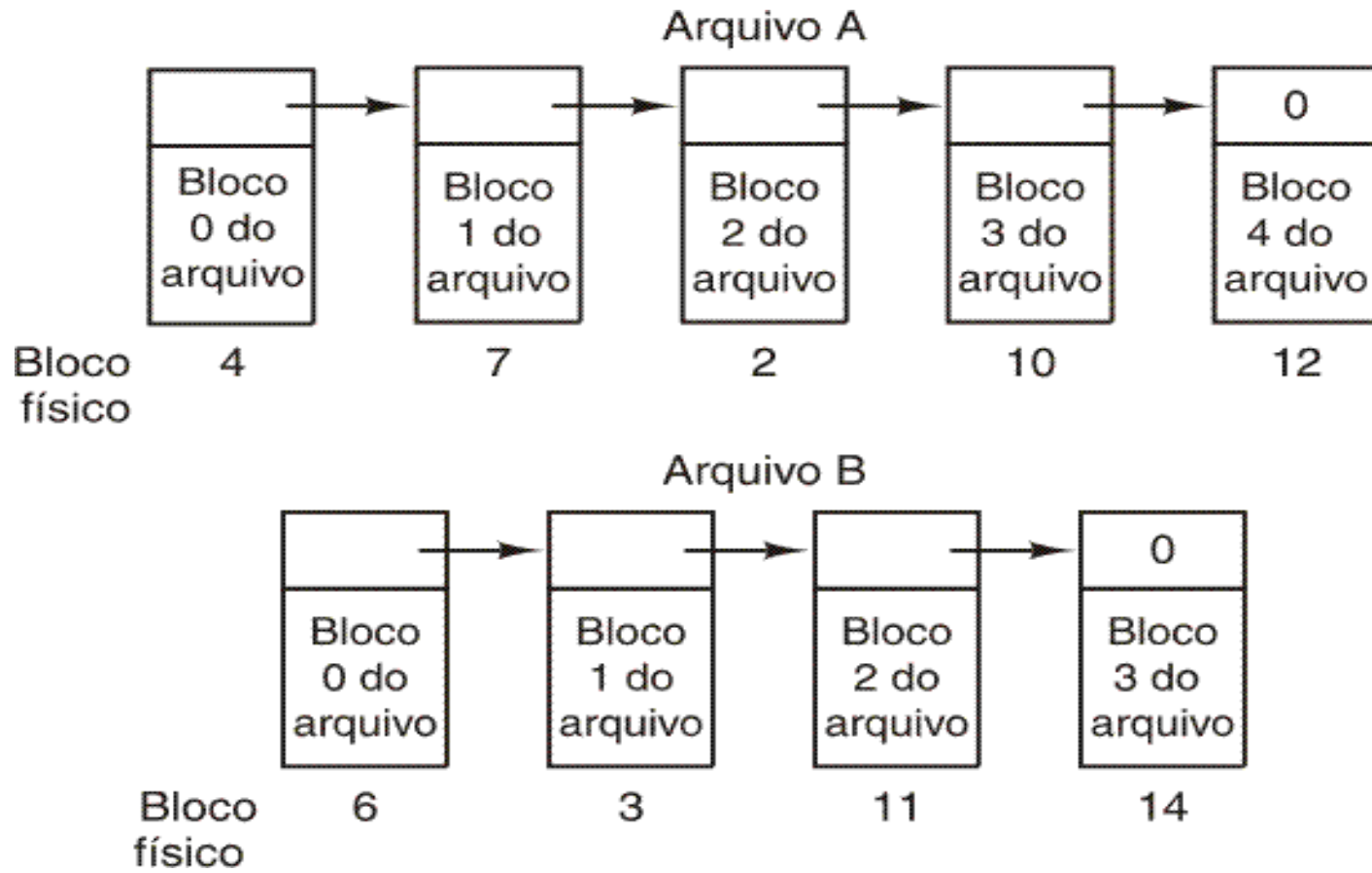


O estado do disco depois de os arquivos D e F terem sido removidos.

## Alocação baseada em listas ligadas

- Evita fragmentação
- Problemas para leitura não sequencial
- Espaço desperdiçado em cada bloco para o ponteiro

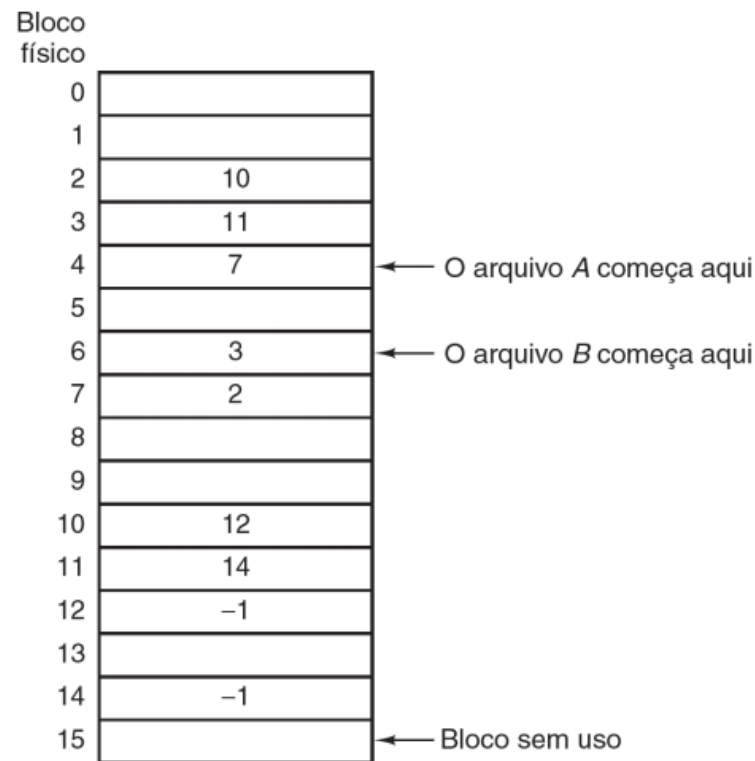
### Alocação baseada em listas ligadas



### Alocação baseada em listas ligadas usando uma tabela em memória

- Busca resolver os dois problemas da lista ligada
- Retira os ponteiros do bloco e transfere para uma tabela separada
- Conhecida com Tabela de Alocação de Arquivos (FAT)
- A tabela fica gravada no disco mas é carregada para a memória
- O espaço da tabela em memória é muito grande

### Alocação baseada em listas ligadas usando uma tabela em memória

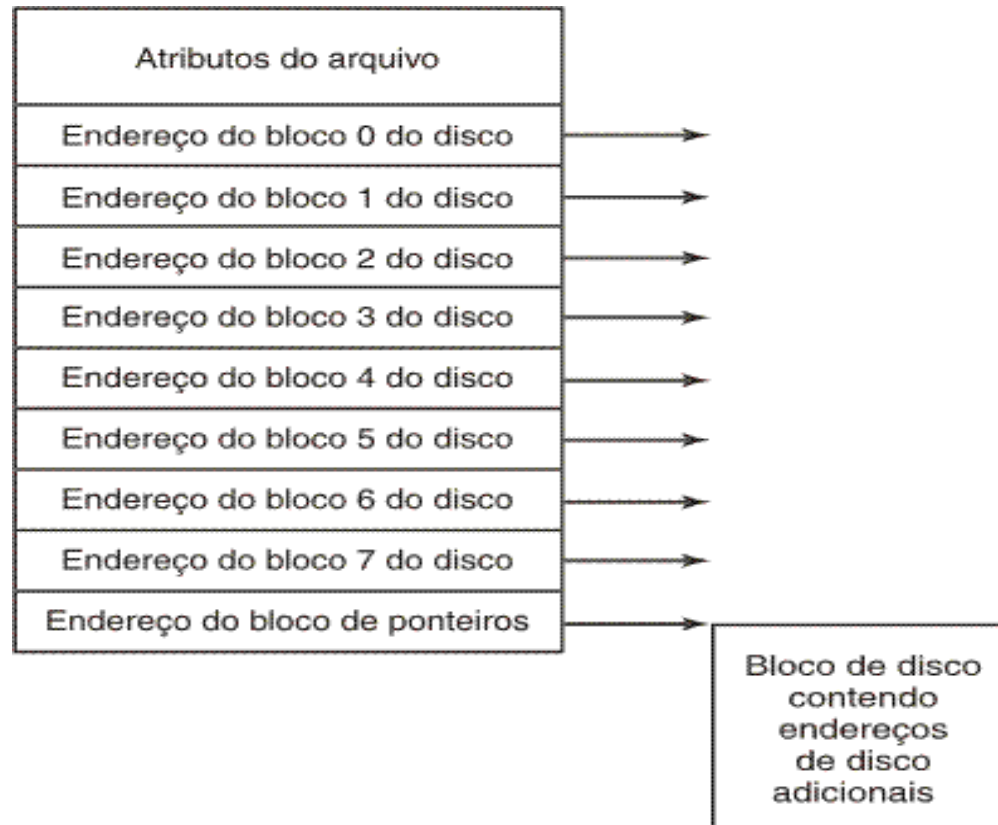




## I-Nodes

- Uma estrutura de dados que armazena os metadados do arquivo e o endereço de seus blocos
- Apenas os i-nodes dos arquivos abertos no momento precisam estar em memória
- Um problema é para arquivos grandes onde o número de blocos supera a capacidade de um i-node
  - Solução: Endereço de Bloco de Link Indireto

### I-Nodes



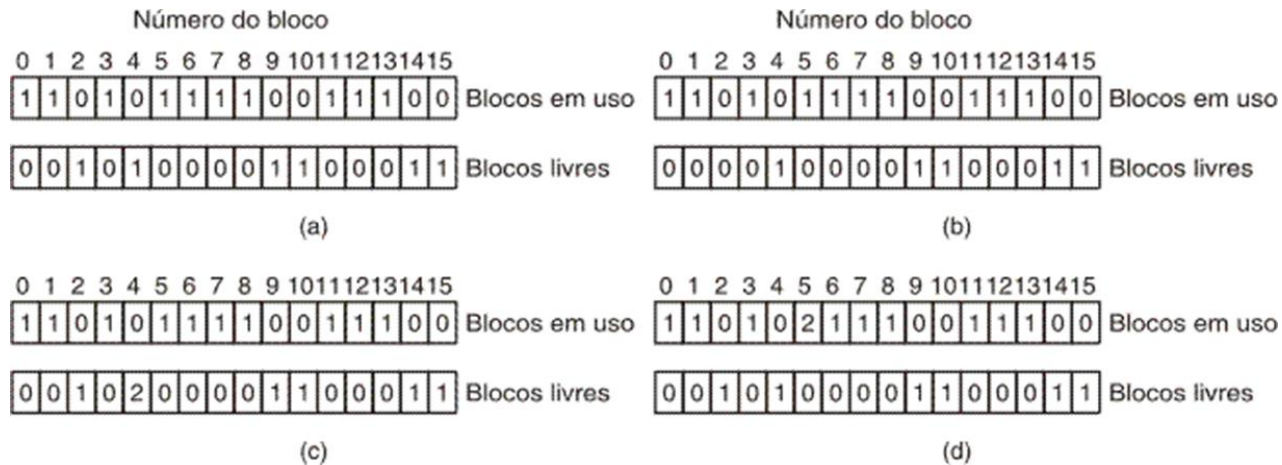
## Confiabilidade

- Sistemas de arquivos com journaling
  - Conceito de transações atômicas
  - Journaling grava qualquer operação que será feita no disco em uma área especial chamada “journal”. Assim, se acontecer algum problema durante alterações no disco, ele pode voltar ao estado anterior do arquivo, ou finalizar a operação
  - Ex.: ext3, ReiserFS, NTFS

## Checagem de disco

- Fck (File Check)
- Duas tabelas
  - Uma indica quantas vezes um bloco aparece na lista de free-blocks
  - Outra indica quantas vezes um bloco aparece referenciado por um arquivo

### fsck



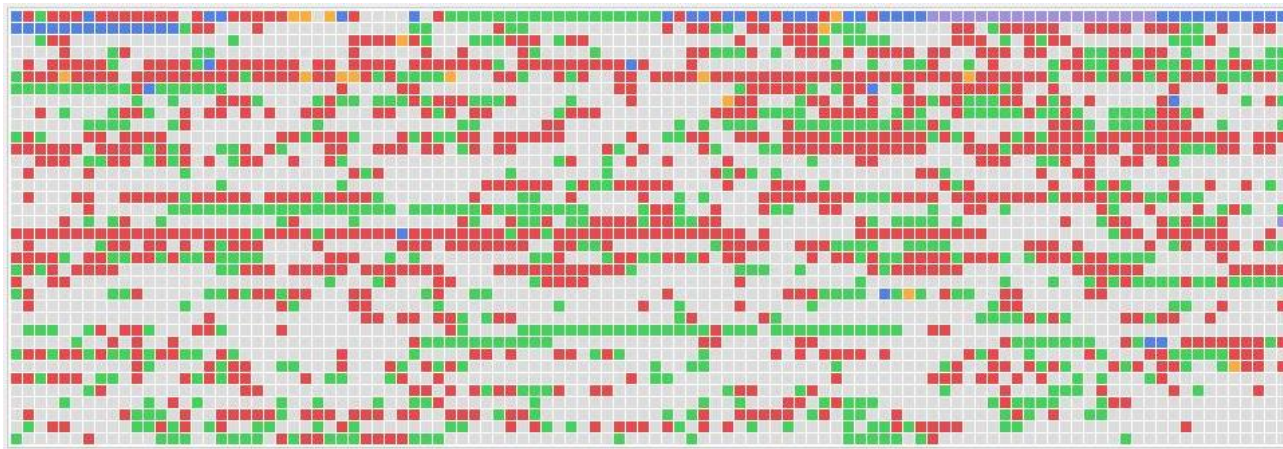
- Estados do sistema de arquivos
  - a) consistente
  - b) bloco desaparecido
  - c) bloco duplicado na lista de livres
  - d) bloco de dados duplicados

## Performance

- Técnicas para aumentar a performance em sistemas de arquivos
- Cache
  - Pode usar os mesmos algoritmos de paginação
  - Duas características podem melhorar a performance
    - É provável que o bloco será usado novamente?
    - O bloco é essencial para a consistência do FS?
- Leitura de bloco antecipada
  - Ler o próximo bloco do disco pertencente ao arquivo recentemente lido
    - Bom para arquivos em leitura sequencial
    - Ruim para leitura aleatória pois desperdiça tempo do disco

## Desfragmentação

- Útil para agrupar os arquivos após muitas leituras e gravações
- Defrag



## Linux

- Múltiplos sistemas de arquivos carregáveis
  - Ext3, ReiserFS, XFS, JFS
  - Módulos do kernel
- Links
- Locks
- Sistema de Arquivos Virtual (VFS)
  - Permite que se utilize múltiplos sistemas de arquivo
  - Abstrai detalhes do tipo de FS usado



## Linux

- Minix 1
  - Arquivos de até 64MB
- Ext
  - Arquivos maiores
- ext2
  - Melhora na performance do ext
- ext3
  - Implementação de Journalling

### ext2

- Divide o disco em grupos
- Cada grupo consiste de:
  - Superbloco
  - Descritor do grupo
  - Bitmap de blocos livres
  - Bitmap de i-nodes livres
  - Inodes
  - Blocos de dados

## ext2 - Superbloco

- Contem informações sobre o layout do sistemas de arquivos
- Ex.:
  - Número de blocos
  - Número de i-nodes
  - Início da lista de blocos livres

## ext2 - Descritor do grupo

- Informações do grupo
  - Local do Bitmaps i-nodes livres
  - Local do Bitmaps de blocos livres
  - Número de diretórios neste grupo

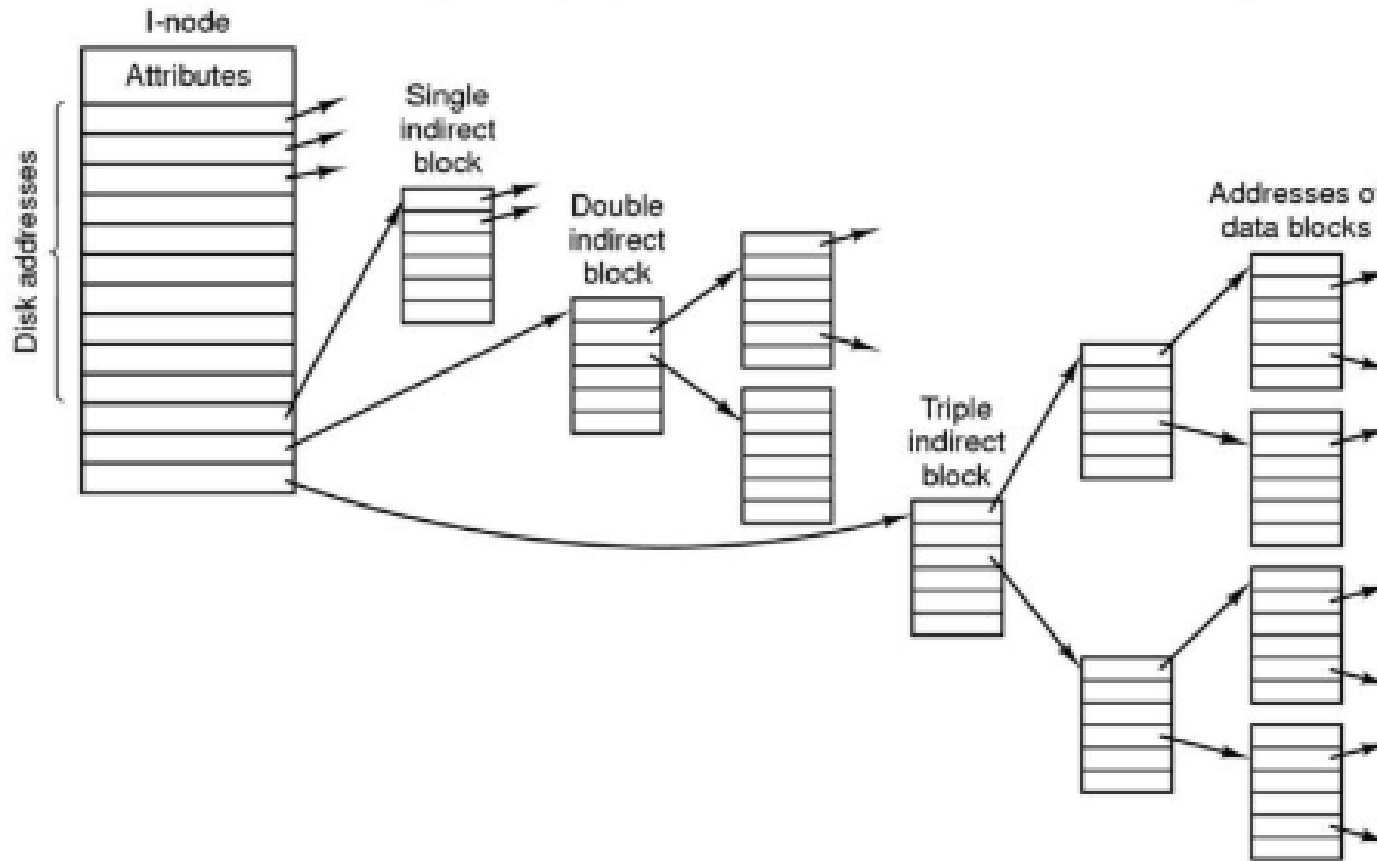
## ext2 - Bitmaps

- Um mapa de bits para blocos livres
- Um mapa de bits para i-nodes livres

## ext2 - Inodes

- Um para cada arquivo
- Um I-node ocupa 128 bytes e descreve um único arquivo
- Campos do I-node
  - Modo
  - NLinks
  - UId/GId
  - Tamanho
  - Hora de Acesso/Gravação/Criação

### I-Nodes



## ext2 - Blocos de dados

- Onde os dados reais estão fisicamente gravados
- Os blocos de um mesmo arquivo não precisam necessariamente ser contínuos



## ext3

- É uma evolução do ext2
- Uso de Journalling
- O journal pode ser armazenado no mesmo disco ou em outro
- Pode ser configurado para fazer log apenas de metadados ou de todas as operações do disco

## Windows

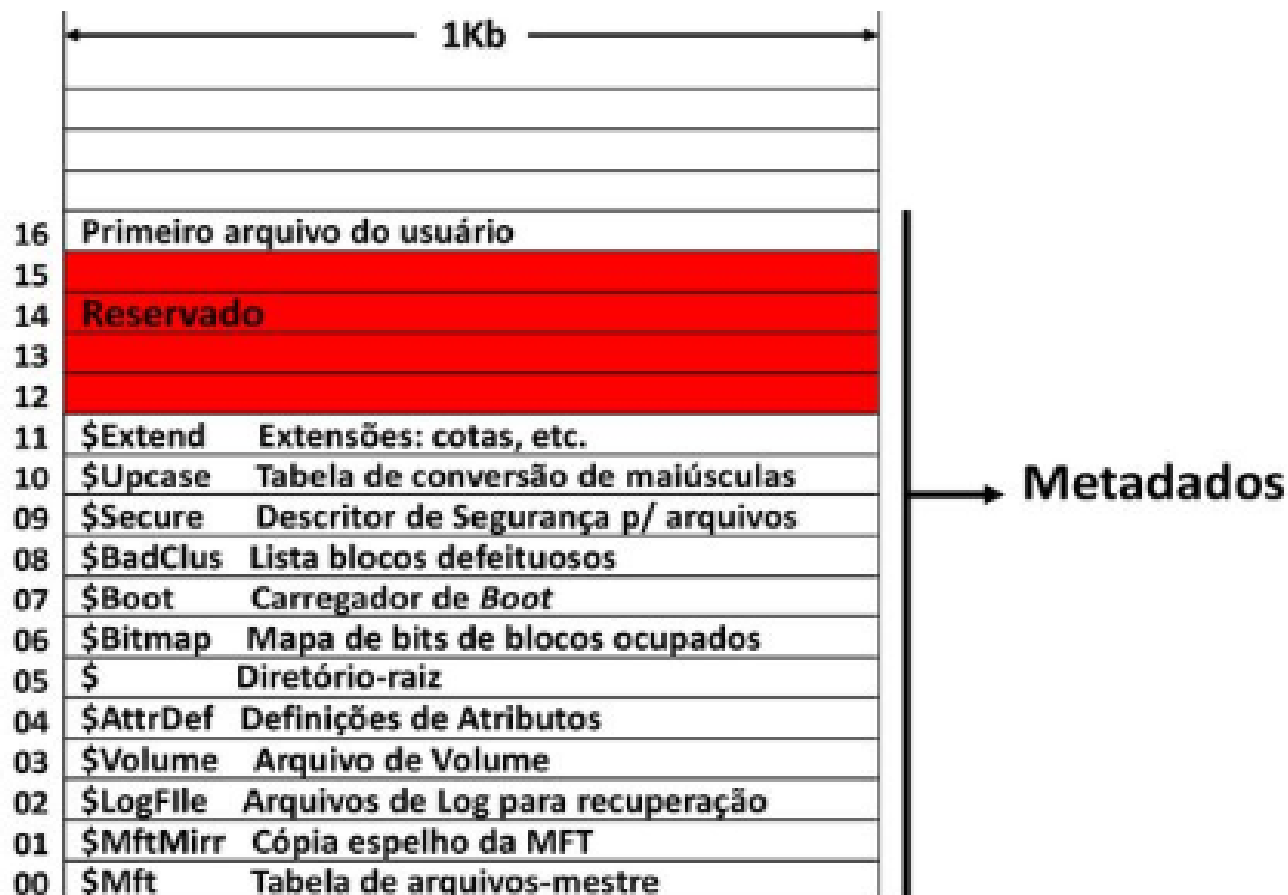
- Sistemas de arquivos FAT e NTFS
- FAT16 e FAT32
  - MsDOS até Windows98 e início do Windows XP
- NTFS
  - Windows NT, Windows XP e sucessores

## NTFS

- Usa uma área do disco para manter uma tabela com os dados dos arquivos
- Master File Table MFT
- Um bloco pode ir de 512 bytes a 64 KB
  - Depende do tamanho da partição
- O comum é 4KB

### NTFS - MFT

A MFT (Master File Table) é o principal arquivo de metadados, e contém ou aponta para todos os outros arquivos do NTFS.



## NTFS - MFT

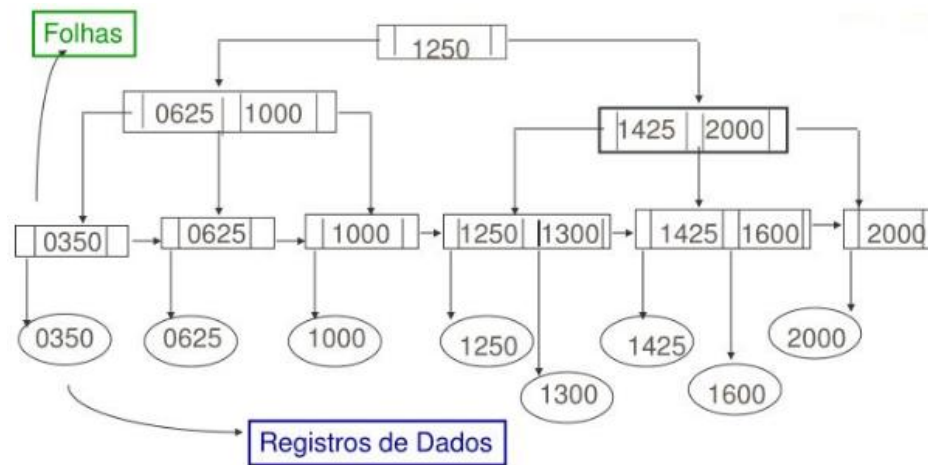
- Uma entrada de registro contêm:
  - Nome do arquivo
  - Lista de atributos
  - Ponteiros para outra entrada contendo a continuação do arquivo
  - Dados
    - Podem estar direto na MFT
    - Na maioria dos casos é um ponteiro para um bloco do disco

## NTFS - Diretórios

- É uma entrada na MFT
- O diretório contém para cada arquivo:
  - O tamanho do nome do arquivo
  - O nome do arquivo
  - Vários outros campos e flags

### NTFS - Diretórios

- Para diretórios grandes (com muitos arquivos)
  - É implementado como uma árvore B+
  - Facilita a busca nominal de um arquivo



## NTFS – Outras características

- Compressão
- Arquivos esparsos
- Journalling
- Criptografia de arquivos