

Sistemas de Arquivos

Pedro Cruz




EEL770 – Sistemas Operacionais

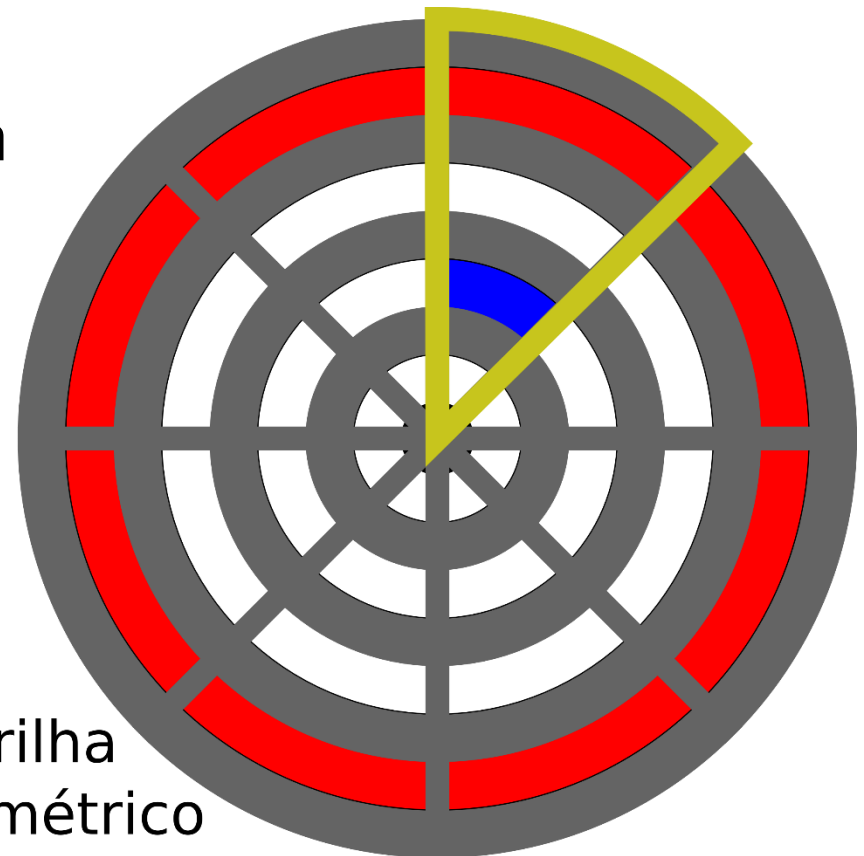
Requisitos de armazenamento



- Persistência
 - Uma vez escritos, dados não podem ser perdidos
- Flexibilidade
 - Volume de dados pode crescer e diminuir
- Acessos múltiplos
 - Diversos processos acessam dados simultaneamente

- Sequência linear de blocos
- Interface de leitura e escrita
 - Ler do bloco k
 - Escrever no bloco k

-  Trilha
-  Setor de trilha
-  Setor geométrico



Problemas com discos



- Como um processo encontra informações?
- Como um processo encontra espaço livre?
- Como garantir **segurança**?

- Confidencialidade
 - Proprietário decide quem tem acesso à informação
- Autenticidade
 - É possível identificar quem tem acesso à informação
- Integridade
 - Modificações na informação são detectadas
- Disponibilidade
 - Informação está sempre disponível para usuários autorizados
- Não-repúdio
 - Operações são sempre mapeáveis em um usuário

- Sempre resolvendo problemas difíceis
 - Processos
 - Espaços de endereçamento
 - **Arquivos**

- Abstração para utilização do disco
- Gerenciamento realizado pelo sistema operacional

Nomes de arquivos



- Forma de buscar os arquivos
 - Arquivos são acessados a partir de seus nomes
- Forma de dar semântica aos arquivos
 - Extensão
 - Windows se importa
 - UNIX ignora
 - Processos podem se importar

Estruturas de arquivos – do ponto de vista do SO



- Sem estrutura
 - Cada arquivo é uma sequência de bytes
- Estrutura de registros
 - Cada arquivo é uma sequência de registros
 - Registro é uma estrutura interna de tamanho fixo
- Estrutura em árvore
 - Cada arquivo é um dicionário (chave, valor)
 - Elementos organizados em árvore
 - Busca por chave

Tipos de arquivo



- Arquivos regulares
- Diretórios
 - Controlam a estrutura de arquivos
- Arquivos especiais de caractere
 - Modelam dispositivos seriais
- Arquivos especiais de blocos
 - Modelam discos
 - Primeiro setor de todo disco recebe um

- Arquivos ASCII (*American Standard Code for Information Interchange*)
 - Legíveis
 - Separados em linhas
 - *Carriage return* (\r)
 - *Line feed* (\n)
- Arquivos binários
 - Compreensíveis para o processo que o gerou

Arquivos executáveis



- Um tipo de arquivo binário
- Reconhecido pelo operacional

Identificador de executável
Tamanho do texto
Tamanho dos dados
Tamanho dos dados não inicializados
Tamanho da tabela de símbolos
Ponto de entrada
Flags
Texto
Dados
Bits de realocação
Tabela de símbolos

Leitura e acesso a arquivos



- Sequencial
 - Fita magnética
 - Não pode pular linha
 - Não pode ler fora de ordem
- Aleatório
 - Discos e estado sólido
 - Pode ler fora de ordem
 - Pode pular linha
 - *seek*

Atributos de arquivos



- Proprietário
- Proteção
- Sistema
- Tamanho de registro
- Momento de criação
- Momento de última leitura
- Momento de última escrita (ou seja, de modificação)
- Tamanho em disco
- Tamanho máximo em disco
- Arquivamento

Operações com arquivos



- Criar (*create*)
- Apagar (*delete*)
- Abrir (*open*)
- Fechar (*close*)
- Ler (*read*)
- Escrever (*write*)
- Acrescentar (*append*)
- Buscar (*seek*)
- Ler atributos (*get attributes*)
- Escrever atributos (*set attributes*)
- Renomear (*rename*)

- Diretório único em sistemas “burros”
- Diretórios hierárquicos
 - Árvore de diretórios
 - Nome de caminho absoluto
 - Nome de caminho relativo

Operações com diretórios



- Criar (*create*)
- Apagar (*delete*)
- Abrir diretório (*openDir*)
- Fechar diretório (*closeDir*)
- Ler diretório (*readDir*)
- Renomear (*rename*)
- Conectar (*link*)
- Desconectar (*unlink*)

Implementação de sistemas de arquivos



- Agora chora

Organização do disco



- Partições
 - Tabela de partições
 - *Master Boot Record* (MBR)
 - Setor 0
 - Atributos do sistema de arquivos
 - Identificação
 - Número de blocos

Implementação de arquivos



- Alocação contígua
- Alocação por lista encadeada
- Alocação por lista encadeada com tabela em memória
- I-nodes

Alocação contígua



- Arquivos são alocados em blocos consecutivos do disco
- Arquivos são escritos no primeiro bloco livre
- Vantagens:
 - Implementação simples
 - Início de cada arquivo
 - Tamanho de cada arquivo
 - Leitura eficiente
 - Busca única encontra arquivo completo
- Desvantagens?

- Arquivos são alocados em blocos consecutivos do disco
- Arquivos são escritos no primeiro bloco livre
- Vantagens:
 - Implementação simples
 - Início de cada arquivo
 - Tamanho de cada arquivo
 - Leitura eficiente
 - Busca única encontra arquivo completo
- Desvantagens
 - Apagar arquivos cria lacunas - **fragmentação**
 - Aproveitar as lacunas é muito complicado

Alocação contígua



- Está morta?

Alocação por lista encadeada



- Primeira palavra de cada bloco aponta para o próximo bloco
 - Próximas palavras armazenam dados
- Vantagem:
 - Fragmentação eliminada
- Desvantagem:
 - Leitura extremamente problemática
 - Tamanho do bloco diferente do tamanho pra armazenamento

Alocação por lista encadeada com tabela em memória



- Tabela representa o próximo bloco de cada bloco físico
- Ponteiro para primeiro bloco do arquivo
- **FAT**
 - *File Allocation Table*
- Desvantagem:
 - A tabela de um disco de 1TB, com blocos de 1K, vai ocupar quantos B na memória?

Bloco físico	Próximo bloco
0	
1	3
2	
3	5
4	
5	0

- *Index-node*
 - Estrutura de dados que guarda os endereços do arquivo
 - Guardada em disco
 - Em memória quando o arquivo está em uso
 - Se arquivo possui mais blocos do que o *i-node* comporta
 - Última entrada aponta para o próximo *i-node*
 - Algumas vezes, dois

Bloco do arquivo	Bloco Físico
0	1
1	3
2	6
3	5
4	134
5	34

Implementando diretórios



- Arquivo
 - Tabela que relaciona nomes com setores de disco
 - Pode armazenar atributos dos arquivos também
 - Sistemas com i-nodes armazenam os atributos no i-node

Entrada do diretório	Valor
0	"receitas" 5GB rw
1	"jogos" 12GB rwx
2	"fotos" 1.2GB rw

Entrada do diretório	i-node
0	3
1	143
2	52

Nomes de arquivos com tamanhos grandes



- Tamanhos de nomes variam muito
 - Tamanho único pode ser problemático
- Solução pode ser em duas maneiras
 - Entradas de arquivo com tamanho variável
 - Entradas de arquivo com ponteiros para o nome de arquivo

Diretórios com entradas de tamanho variável



- Uma entrada pra um arquivo possui uma porção fixa e uma variável
 - Porção fixa possui atributos e o tamanho da porção variável
 - Porção variável possui o nome do arquivo e um caractere de terminação

Diretório em disco

Tamanho da entrada do arquivo 1

Atributos do arquivo 1

Nome

Do

Arquivo 1x

Tamanho da entrada do arquivo 2

Atributos do arquivo 2

Nome

Grande

Do

Arquivo 2x

Diretórios com ponteiro para nome de arquivo



- Cada entrada tem um ponteiro para o nome do arquivo
- Nomes ficam armazenados em espaço separado

Diretório em disco

Ponteiro para nome do arquivo 1

Atributos do arquivo 1

Ponteiro para nome do arquivo 2

Atributos do arquivo 2

Nome do arquivo 1x

Nome do arquivo 2

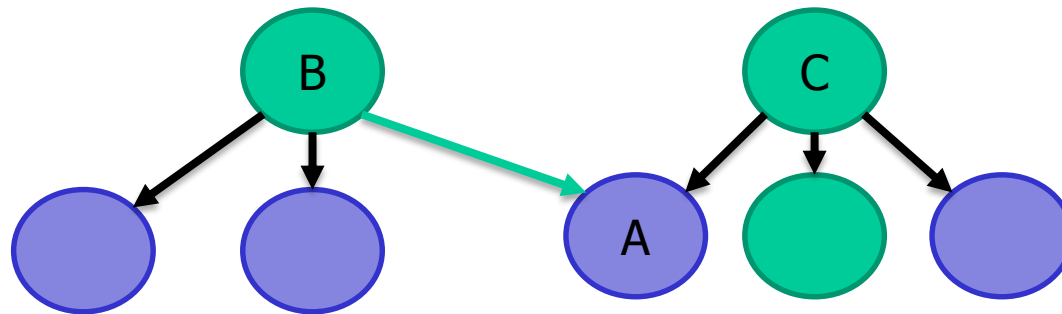
Nome do arquivo 2

Nome do arquivo 2x

Problemas de compartilhamento



- Arquivo possui um proprietário
 - Permissões devem ser obedecidas
- Alterações no arquivo devem ser vistas por todos
 - Simples cópia não funciona
- Manter uma lista de todas as cópias de um arquivo é um problema



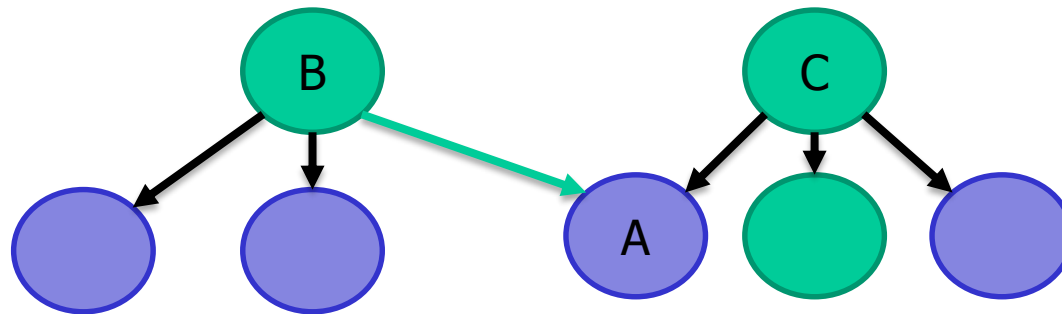
Compartilhamento – soluções possíveis

1. *Hard link*

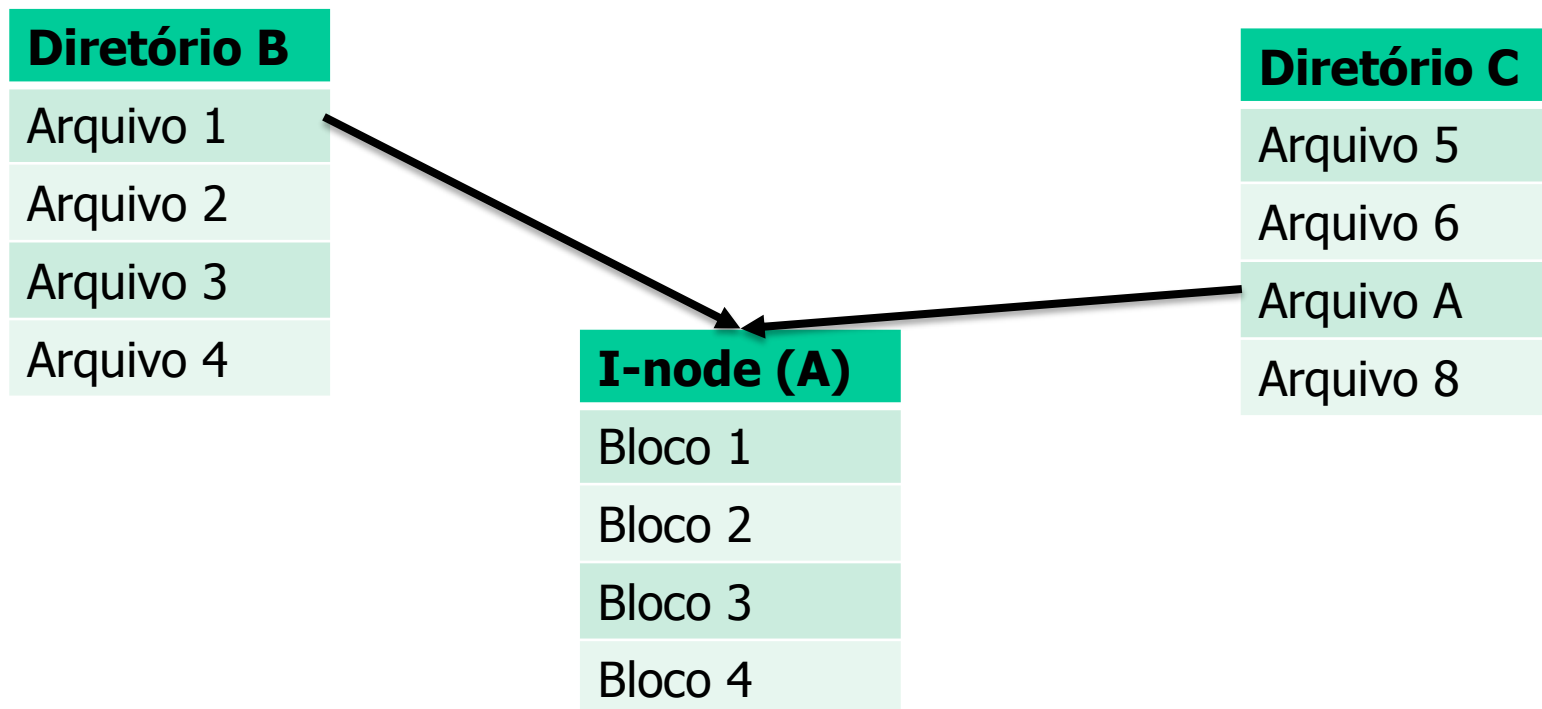
- Diretório B aponta para i-nodes de arquivo A

2. *Soft link*

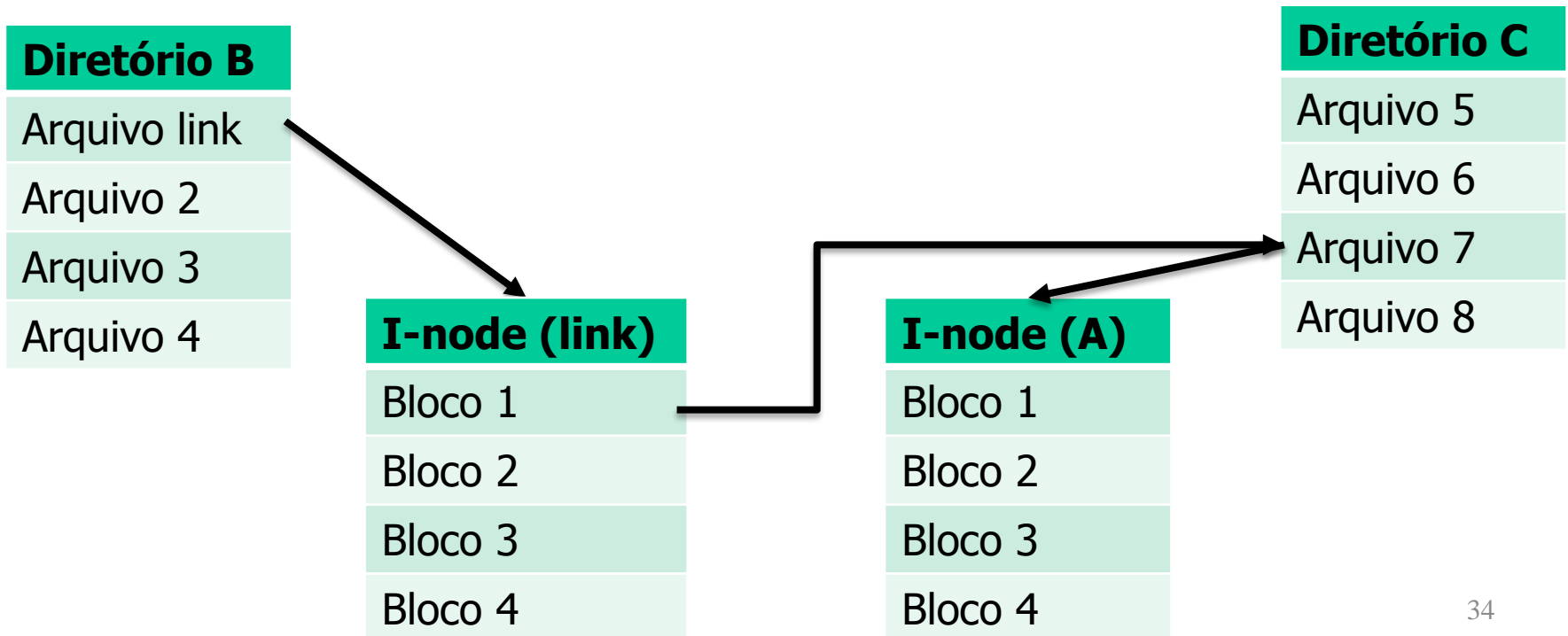
- Arquivo especial em B funciona como *link*



- Diretório B aponta para i-nodes de arquivo A
 - C continua sendo o dono de A
- Apagar arquivo pode ser problemático
 - i-node deve possuir contador de cópias do arquivo



- Arquivo especial em B aponta para o arquivo em C
 - Apagar arquivo A traz problemas



Questões sobre *hard* e *soft* link



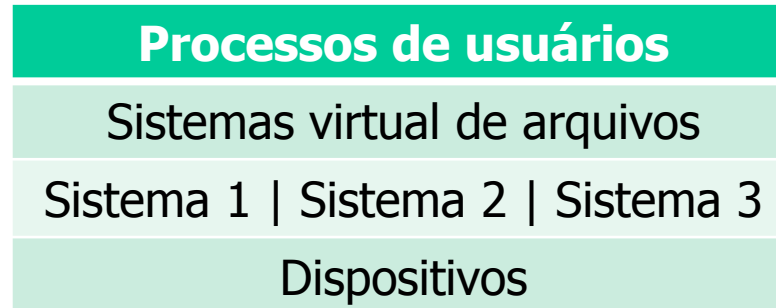
- *Hard link*
 - Não consegue fazer o *link* por nome de arquivo
- *Soft link*
 - Ocupa um i-node
 - Ocupa um bloco
 - Adiciona um salto à busca de um arquivo
 - Consegue se conectar a arquivos remotos
- Ambos
 - Busca por arquivos em vários diretórios pode encontrar o mesmo arquivo várias vezes

- Forma como arquivos são organizados
 - Muitas possíveis
 - NTFS
 - FAT-16
 - FAT-32
 - Ext2
 - Ext3
 -
 - CD-ROM's, DVD's e afins possuem sistemas próprios!

Virtualização do sistema de arquivos



- UNIX oferece um sistema virtual de arquivos
 - Chamadas de software são feitas a um sistema virtual
 - Sistema virtual traduz para uma interface real



Otimização de sistemas de arquivos



- Gerenciamento de espaço em disco
- Monitoramento de blocos livres
- Cotas de disco
- Backups
- Consistência
- Desempenho
- Desfragmentação

Gerenciamento de espaço em disco



- Velho problema
 - Bytes contíguos
 - Busca fácil
 - Alteração difícil
 - Blocos de bytes
 - Busca difícil
 - Alteração fácil
 - Fragmentação interna
 - Tamanho dos blocos

Tamanho dos blocos



- Blocos grandes
 - Fragmentação interna grande
- Blocos pequenos
 - Arquivos ocupam muitos blocos

Solução:

utilizar dados empíricos para decidir tamanho dos blocos

Monitoramento dos blocos livres



- Lista encadeada
 - Alguns blocos contém ponteiros para blocos livres
 - O último ponteiro aponta para um bloco da lista
 - Bloco de 1KB
 - Disco de 1T
 - Identificador de bloco de 32 bits
 - » Cada bloco pode armazenar 255 blocos livres
 - » Disco inteiro livre ocupa 4 milhões de blocos
- Mapa de bits
 - Alguns blocos contém mapa de bits
 - 0: ocupado; 1: livre
 - 130.000 blocos pra mapear 1T

- Usuários possuem limite de uso do disco
 - Limite estrito
 - Usuário nunca pode exceder o limite estrito
 - Limite flexível
 - Usuário pode exceder o limite flexível durante um uso
 - A cada *login*, limite flexível é verificado
 - » Se usuário excedeu o limite, é barrado

- Nem todos os arquivos devem sofrer backup
 - Executáveis, arquivos de E/S, temporários
- Nem todos os arquivos precisam ser totalmente copiados
 - Cópias incrementais
 - Deve haver pelo menos uma cópia total
- Nem sempre o sistema pode ser copiado
 - Copiar arquivos que estão sendo usados e alterados é complicado

Backup (2)



- A segurança do backup é importante
 - Danos
 - Roubos
- Cópias físicas
 - Copiar um disco inteiro, não importa o que está escrito
- Cópias lógicas
 - Programa de backup lê diretórios e pastas específicas

- Operações de escrita deixam sistema temporariamente inconsistente
 - Criação de arquivo
 - Deleção de arquivo
- Contagem de estruturas
 - $i\text{-nodes} + \text{blocos livres} = n^{\circ} \text{ total de blocos}$
 - $N^{\circ} \text{ de diretórios} = N^{\circ} \text{ de ligações}$
- Detecção de anomalias
 - Diretórios com muitos subdiretórios/arquivos
 - Permissões estranhas

- Disco demora até um milhão de vezes a mais do que a memória primária
 - Apenas melhora, querido
- Técnicas empregadas para melhorar o desempenho
 - Cache de blocos
 - Leitura antecipada em blocos
 - Redução do movimento do braço do disco
 - Desfragmentação de disco

Cache de blocos



- Tabela em memória dos blocos mais usados
 - Política de cache
 - Política de atualização
 - i-nodes são pouco referenciados, mas devem ser escritos rapidamente
 - Discos podem ser removidos
 - Caches de escrita direta

Leitura antecipada de blocos



- Leitura sequencial de arquivos facilita a previsão de blocos que serão necessários
 - Ler partes posteriores dos arquivos conforme partes anteriores estão sendo processados

Redução do movimento do braço do disco



- Alguns blocos são lidos sequencialmente
 - Se estiverem próximos, o braço se movimentará menos
- I-nodes exigem 2 buscas em disco (1 para o i-node e 1 para o arquivo)
 - I-nodes no início do disco
 - Trajeto i-node -> arquivo médio é de $\frac{1}{2}$ disco
 - I-nodes no meio do disco
 - Trajeto i-node -> arquivo médio é de $\frac{1}{4}$ de disco
 - I-nodes espalhados pelo disco
 - Trajeto i-node -> arquivo ainda menor

Desfragmentação de disco



- Arquivos mudam de tamanho
 - Nascem
 - Crescem
 - São removidos
- Alterações nos arquivos geram fragmentação externa
 - Tempo de busca aumenta
- Operacionais diferentes têm políticas diferentes
 - Windows
 - Arquivos são escritos de forma contígua
 - Linux
 - Arquivos possuem espaço entre si para crescerem
- SSD?

Sistemas de Arquivos

Pedro Cruz

EEL770 – Sistemas Operacionais