Instituto de Informática - UFRGS

Sistemas Operacionais

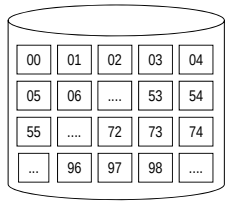
Sistema de arquivos

Organização do espaço em disco

Aula 24

Organização do espaço em disco

- Objetivo: armazenar arquivos de forma a acessá-los eficientemente
  - Alocar blocos suficientes para armazenar o arquivo
- Duas formas básicas:
  - Contigua (alocação contigua)
  - Não-contigua (alocação encadeada e alocação indexada)
- Princípio: disco é formado em blocos lógicos
  - Numerados sequencialmente
  - Bloco = *n* setores físicos



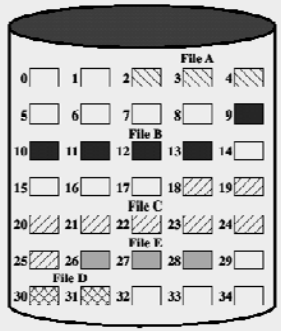
Instituto de Informática - UFRGS
A. Carissimi -5-juin-12

Alocação contígua

- Arquivo é uma seqüência de blocos lógicos contigüos alocados no momento da criação
- Endereços no disco são lineares
  - bloco lógico *i* e *i+1* são armazenados fisicamente em seqüência
  - Reduz a necessidade de *seek* já que blocos estão na mesma trilha
    - No pior caso necessita apenas a troca de cilindro
- Arquivo é descrito através de uma entrada na forma:
  - Bloco físico inicial
  - Tamanho do arquivo em blocos

Instituto de Informática - UFRGS
A. Carissimi -5-juin-12

Esquema alocação contígua



File Allocation Table

File Name	Start Block	Length
File A	2	3
File B	9	5
File C	18	8
File D	30	2
File E	26	3

Instituto de Informática - UFRGS
A. Carissimi -5-juin-12

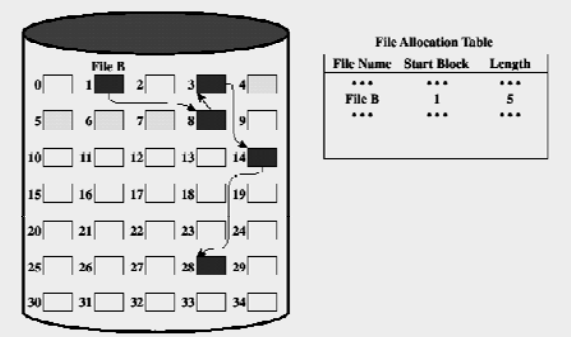
Problemas com alocação contígua

- Problema 1: encontrar espaço para um novo arquivo
  - Técnicas de gerência de memória
    - e.g.; *first-fit, best-fit, worst-fit*
  - Gera fragmentação externa
    - Necessidade de compactação
- Problema 2: determinar o espaço necessário a um arquivo
  - Arquivos tendem a crescer, e se não há espaço contígüo disponível?
    - Aborta execução do programa com erro
    - Recopia o programa para uma zona maior
  - Pré-alocar um espaço máximo para o arquivo
    - Fragmentação interna

Alocação encadeada

- Soluciona os problemas da alocação contígua
  - Relação a dimensionamento do tamanho e crescimento de arquivos
- Arquivo é uma lista encadeada de blocos
  - Cada bloco contém um ponteiro para o próximo bloco
- Arquivo é descrito em uma entrada na forma:
  - Bloco inicial do arquivo
  - Bloco final do arquivo ou tamanho do arquivo em blocos

Esquema de alocação encadeada

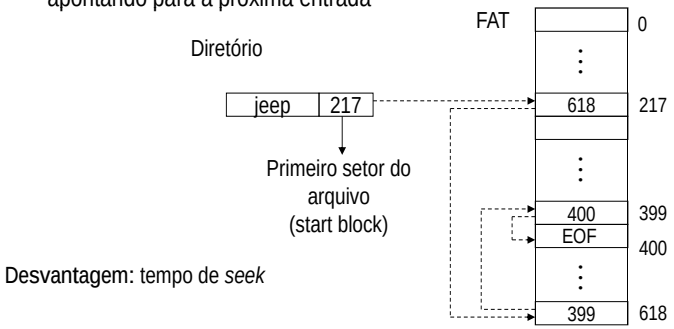


Prós e contras da alocação encadeada

- Elimina a fragmentação externa no disco
- Arquivos podem crescer indefinidamente
  - Não há necessidade de compactar o disco
- O acesso a um bloco *i* implica em percorrer a lista encadeada
  - Afeta o desempenho
  - Adequado para acesso seqüêncial a arquivos
- Confiabilidade
  - Erro provoca a leitura/escrita em bloco pertencente a outro arquivo

File Allocation Table (FAT)

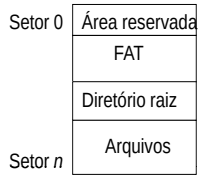
- Variação de alocação encadeada
  - Uma entrada na FAT para cada *bloco lógico* do disco (sistema de arquivos)
  - Composta por um ponteiro (endereço do bloco lógico)
  - Arquivo é descrito por uma seqüência de entradas na FAT, cada entrada apontando para a próxima entrada



Desvantagem: tempo de *seek*

Exemplo: sistema de arquivos FAT (MS-DOS)

- Organização lógica do disco:

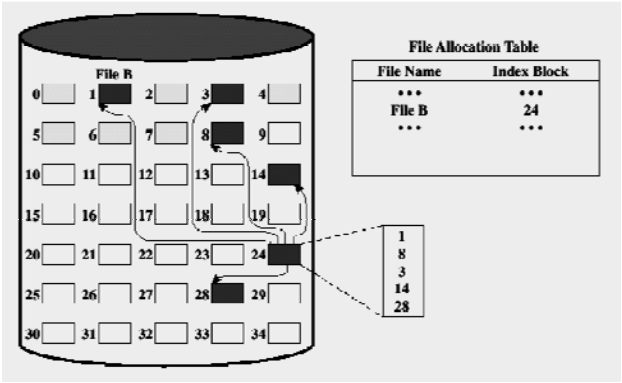


- Diretório raiz possui tamanho fixo em função da capacidade do disco
  - Cada entrada possui 32 bytes
- Tamanho da *File Allocation Table* (FAT) é proporcional a capacidade do disco
- Alocação é baseada em clusters (bloco lógico)
  - 2^n setores (depende da capacidade do disco)

Alocação indexada

- Busca resolver o problema de "ponteiros esparramados" pelo disco que a alocação encadeada provoca
- Um arquivo é descrito por um índice
  - Indice é mantido dentro de um bloco lógico
  - Lista de todos os blocos que compõem o arquivo
- Diretório possui um ponteiro para o bloco de índice de um determinado arquivo

Esquema de alocação indexada



Prós e contras da alocação indexada

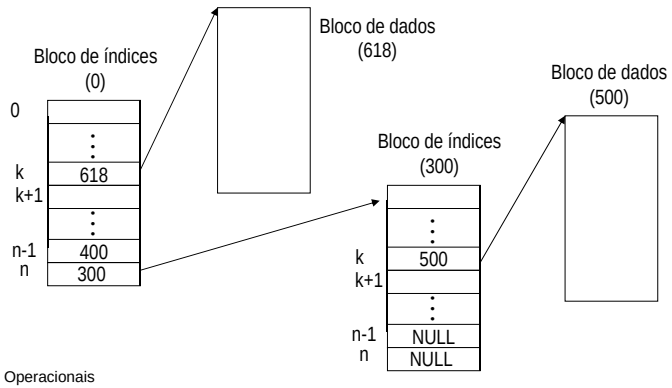
- Permite o acesso randômico a blocos independentes de sua posição relativa no arquivo
- Tamanho máximo do arquivo é limitado pela quantidade de entradas que o bloco lógico suporta
  - Muito pequeno (limita tamanho do arquivo)
  - Muito grande (desperdiça espaço em disco)

Variações em alocação indexada

- Buscam resolver o problema do tamanho do bloco de índices
- Três métodos básicos:
  - Encadeado
  - Multinível
  - Combinado

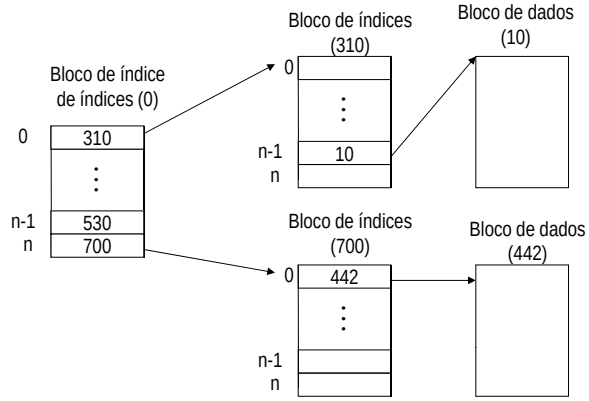
Método encadeado

- O bloco de índice mantém ponteiros para os blocos que compõem o arquivo com exceção da última entrada
  - Mantém um ponteiro para outro bloco onde índice continua



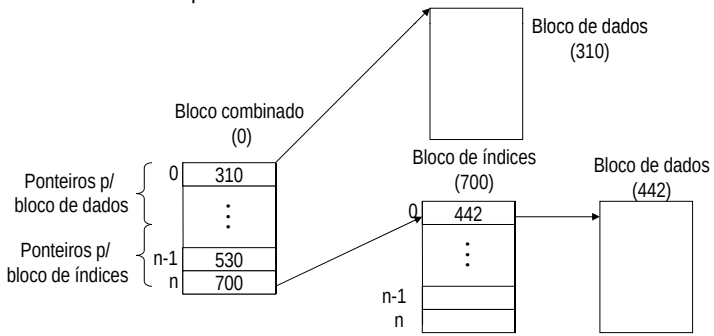
Método multinível

- Mantém um índice de índices
  - Não resolve completamente o problema de limite



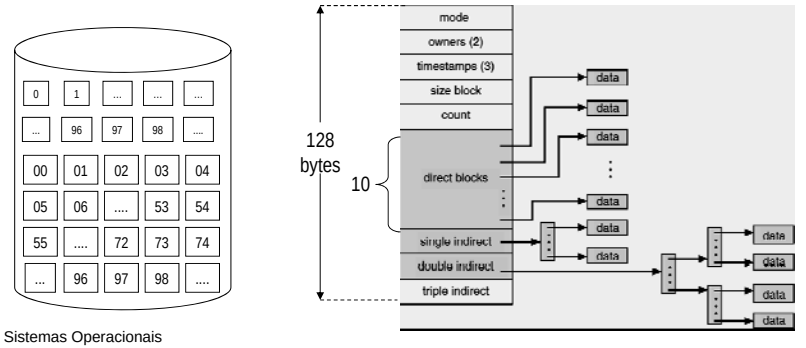
Método combinado

- Métodos encadeado e multinível em uma única estrutura de dados
- O que justifica essa combinação?
  - Acesso otimizado a blocos de dados: método indexado
  - Limite de arquivos: multinível



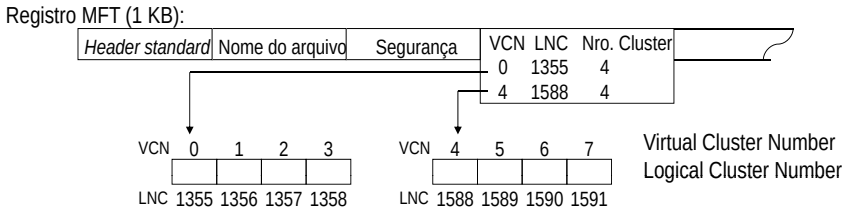
Exemplo: *i-node* (descritor de arquivos UNIX)

- Método combinado com dois tamanhos de blocos: um para índice e outro para dados
- Pré-definidos em uma tabela no disco (partição)
- Entrada do diretório associa nome simbólico a *i-node*



Exemplo: NTFS

- Descritores (registros) são armazenados no MFT
- 12 entradas iniciais são arquivos de controle (*metafiles*)
  - Demais entradas estão disponíveis para arquivos e ou diretórios
- Cada entrada é formada por um conjunto de pares {atributo, valor}
  - Se houver atributos que excedam a capacidade de uma entrada, uma nova entrada é encadeada a essa primeira



Leituras complementares

- R. Oliveira, A. Carissimi, S. Toscani; Sistemas Operacionais. Editora Sagra-Luzzato, 2001.
  - Capítulo 8, seção 8.3.1
- A. Silberchatz, P. Galvin; *Operating System Concepts*. 4th edition, 1994, Addison-Wesley.
  - Capítulo 11 seção 11.2