Sistemas Operacionais
parte II

Professor: Emerson Rogério de Oliveira Jr.

O material apresentado aqui (conjunto de slides) foi fortemente baseado nos slides de aula da disciplina de Sistemas Operacionais, da UFRGS, elaborado pelos professores Rômulo Silva de Oliveira, Alexandre da Silva Caríssimi e Simão Sirineo Toscani, de acordo com o livro destes mesmos professores

OLIVEIRA, R. S.; CARÍSSIMI, A. S. e TOSCANI, S. S. Sistemas Operacionais, 3ª ed. Ed. Sagra-Luzzatto. 2004.

Sumário

- Entrada e saída
- Sistema de arquivos

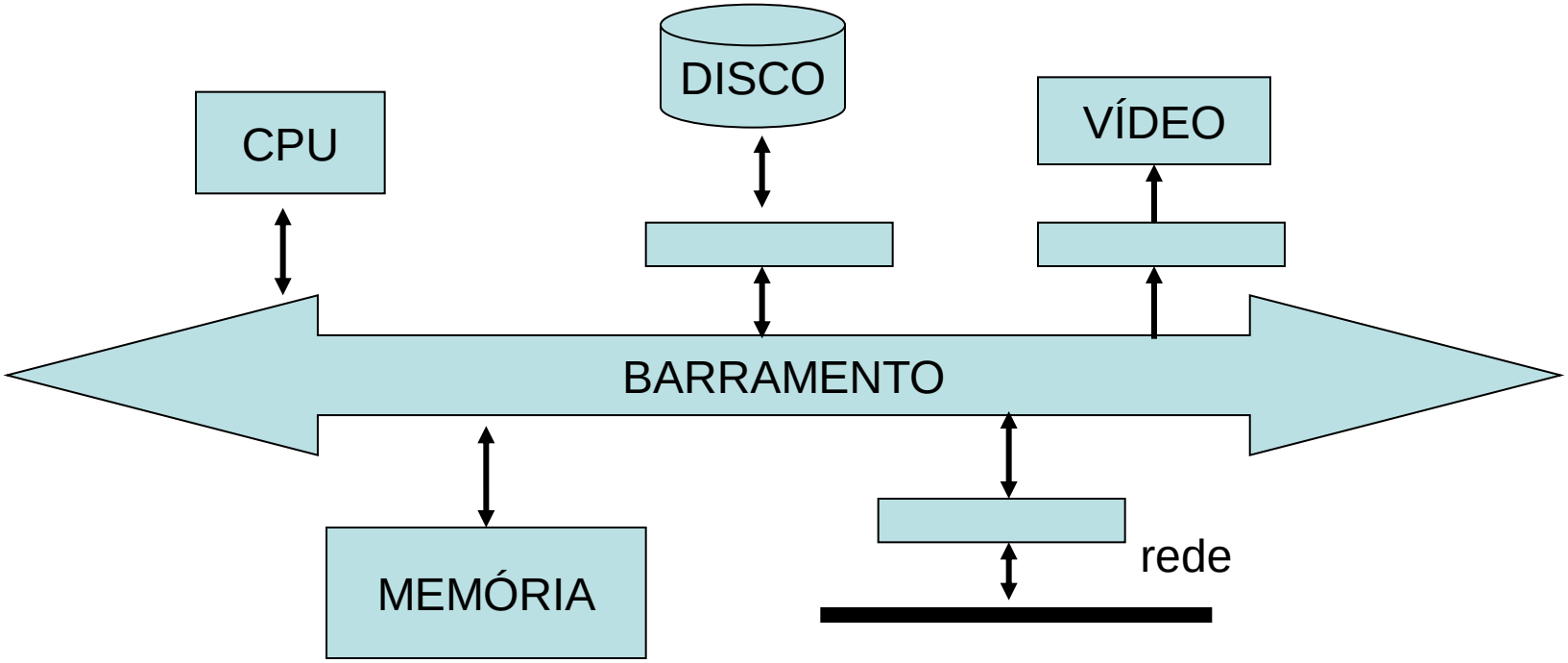
Entrada e saída

Princípios básicos de hardware

- Periférico é um dispositivo conectado a um computador, através de uma interface, de forma a possibilitar sua interação com o mundo externo
- As interfaces são conectadas aos barramentos internos de um computador
  - Elemento chave na coordenação da transferência de dados
  - Interfaces se utilizam de um processador dedicado à realização e controle das operações de entrada e saída
    - controladoras

Arquitetura de entrada e saída

- Um dispositivo de entrada e saída possui uma parte mecânica e outra eletrônica



Dispositivos de entrada e saída

- Classificados como
  - Orientado a caractere
    - Unidade de transferência é o caractere
      - Teclado, interface serial
  - Orientado a bloco
    - Unidade de transferência é um bloco de caracteres (fixo)
      - Disco, fitas DAT
- Esquema de classificação não é perfeito, pois alguns dispositivos não se enquadram nestas situações
  - Relógio, memória de vídeo mapeada em espaço de E/S

Dispositivos de entrada e saída

- Podem ser classificados de acordo com o tipo de entidade que interagem
  - Com usuário
    - Vídeo, teclado, mouse, etc.
  - Com dispositivos eletrônicos
    - Discos, fitas, controladores, etc.
  - Com dispositivos remotos
    - Modem, interfaces de rede
- Apresentam características próprias
  - Taxa de transferência de dados
  - Complexidade de controle
  - Unidade de transferência
  - Representação de dados (codificação)
  - Tratamento de erros

Tipos de conexão e transferência de dados

- Em função da interconexão física das interfaces com os periféricos, podem ser classificadas em dois tipos
  - Interface serial
  - Interface paralela
- Interface serial
  - Apenas uma linha para transferência de dados (bit a bit)
- Interface paralela
  - Mais de uma linha para transferência de dados
    - *n* x 8 bits

Como controladoras e sistema operacional interagem?

- A controladora é programada via registradores de configuração
  - Recebem ordens do processador
  - Fornecem estados de operação
  - Leitura e escrita de dados do periférico
- Registradores são "vistos" como posições de memória
  - E/S mapeada em espaço de E/S
  - E/S mapeada em espaço de memória

Mapeamento em espaço de memória e em espaço de entrada e saída

- Espaço de endereçamento
  - Conjunto de endereços de memória que o processador consegue endereçar
  - Definido no projeto de processador
    - Pode existir um único espaço de endereçamento
    - Pode existir um espaço de endereçamento dedicado à entrada e saída
- Instruções específicas para acessar um ou outro espaço de endereçamento
  - mov, in, out

Mapeamento em espaço de memória

- Um único espaço de endereçamento
- No projeto do computador, se reserva uma parte de sua área de endereçamento para acesso a periféricos (controladoras)
- Instruções de acesso à memória do tipo **mov end, dado** podem tanto referenciar uma posição real de memória como um registrador associado a um periférico de entrada / saída
- Exemplo: processadores Motorola

Mapeamento em espaço de entrada/saída

- O processador possui duas áreas distintas de endereçamento
  - Espaço de memória: acessado via instruções de acesso de memória (*mov*)
  - Espaço de E/S: acessado via instruções de acesso específico (*in, out*)
- No projeto de um computador usando tal processador, é possível utilizar os dois tipos de mapeamento para acesso a periféricos de entrada/saída
- Exemplo: processadores Intel

Técnicas para realização de E/S

- E/S programada
  - Toda interação entre o processador e o controlador é feita pelo programador
    - Envio de comando para a controladora
    - Espera pela realização do comando
  - Controladora de E/S atualiza *bits* de estado da operação
  - Processador espera o término da operação (*busy waiting*)
  - Desvantagem
    - Desperdício de tempo do processador para verificar continuamente o estado de uma operação de E/S

Técnicas para realização de E/S

- E/S orientada a interrupções (*interrupt driven*)
  - Evita desperdício de tempo
  - Processador é interrompido quando o módulo de E/S está pronto
  - Processador é responsável por iniciar uma operação de E/S
  - Desvantagem
    - Processador atua como um intermediário na transferência, pois cada palavra escrita ou lida passa pelo processador

Técnicas para realização de E/S

- E/S através de acesso direto à memória
  - DMA (*Direct Access Memory*)
  - Transfere diretamente um bloco de dados entre a memória e o módulo de E/S
  - O mecanismo de interrupção é utilizado para sinalizar final de tarefa
  - Processador é envolvido com a tarefa de E/S apenas no começo e no final da transferência

O problema da gerência de entrada e saída

- Entrada e saída é extremamente lenta se comparada com a velocidade de processamento e de acesso à memória principal
- Multiprogramação possibilita que processos executem enquanto outros esperam por operações de entrada e saída
- Procedimento de *swapping* é entrada e saída
- Eficiência no tratamento de entrada e saída é importante

Objetivos da gerência de entrada e saída

- Eficiência
- Esconder detalhes de acesso para camadas mais baixas
- Disponibilizar comandos de alto nível como *read, write, open* e *close*
- Um subsistema de entrada e saída é necessário
  - Os periféricos são diversificados
  - Objetivo é padronizar rotinas de acesso de E/S
  - Organizado em camadas
    - E/S nível de usuário
    - E/S independente do dispositivo
      - *Drivers* dos dispositivos

Software de E/S

- Síncrono (bloqueante)
  - Processo é suspenso até a conclusão da operação
  - Programação e uso simples
- Assíncrono (não bloqueante)
  - Processo continua sua execução enquanto E/S é realizada
  - Mecanismos empregados
    - Interrupção para sinalizar conclusão da operação
    - Emprego de *multithreading*

Drivers de dispositivos

- Camada inferior do software de E/S
- Responsável pela gerência do dispositivo físico de E/S
- Código específico a um dispositivo
- Recebe solicitações da camada de gerência de dispositivo
- Responsável pelo tratamento de erros

Entrada e saída independente do dispositivo

- Implementa procedimentos e funções gerais a todos os dispositivos de E/S
- Fornece uma visão lógica do dispositivo através de dados genéricos que representam classes de dispositivos
- Principais serviços
  - Escalonamento de E/S
  - Denominação
  - Bufferização
  - Cache de dados
  - Alocação e liberação
  - Direitos de acesso
  - Tratamento de erros

Entrada e saída independente do dispositivo

- Atribuição uniforme do nome independente do dispositivo
- O unix é um exemplo
  - Nome do dispositivo é um string
  - Discos (dispositivos) fazem parte do sistema de arquivos

Entrada e saída a nível de usuário

- "visão" que o usuário possui dos dispositivos de entrada e saída
  - Interface de programação associada a bibliotecas de entrada e saída
  - Interface de comandos (ls, cp, mv)
- Construído sobre primitivas do sistema operacional
  - Chamadas de sistema

Disco magnético: o periférico mais importante

- Porque permite
  - Armazenamento de dados
  - Suporte à implementação de memória virtual
  - Tolerância a falhas (RAID)
- Um disco de plástico ou metal recoberto de material magnético
- Dados são gravados e recuperados através do cabeçote de leitura e gravação
  - Escrita:
    - o cabeçote é submetido a uma tensão que gera um campo magnético que magnetiza o disco com diferentes padrões de polaridade
  - Leitura:
    - a variação do campo magnético gerado pela rotação do disco induz uma corrente no cabeçote de leitura

Disco magnético: o periférico mais importante

- Disco é organizado em uma seqüência de círculos concêntricos
  - Trilhas, separadas por *gaps*
  - Duas tecnologias
    - CAV (*Constant Angular Velocity*)
      - O número de bits por trilha é constante
    - CLV (*Constant Linear Velocity*)
      - O número de bits por trilha depende se ela é mais interna ou mais externa – CDROM
  - Transferência de dados é feita em blocos (bloco é menor que trilha)
  - Trilha é subdividida em setores

Exemplo de um disco magnético

- Disco 4.1 Gigabytes
  - 255 cabeças
  - 63 setores de 512 bytes
  - 525 cilindros
  - Capacidade: 255x63x512x525
- O sexagésimo quarto setor mantém um mapa de setores da trilha

Desempenho do disco

- Para ler ou escrever dados é necessário que o cabeçote de leitura e gravação esteja posicionado na trilha e no início do setor desejado
- Três tempos envolvidos
  - Tempo de posicionamento (*seek time*)
    - Tempo necessário para posicionar o cabeçote na trilha desejada (aprox. 5 a 10 ms – ano 2000)
  - Tempo de latência rotacional
    - Tempo necessário para atingir o início do setor desejado
    - Discos rígidos: 5400rpm a 10000rpm)
  - Tempo de transferência
    - Tempo para a escrita ou leitura dos dados

Escalonamento do disco

- Tratar E/S em disco de forma eficiente
  - Minimizar o tempo de posicionamento (*seek*)
- Algoritmos
  - FCFS
  - SSTF
  - SCAN
  - C-SCAN
- Exemplo
  - Disco organizado em 200 trilhas
  - Posição inicial do cabeçote: trilha 53
  - Requisições:98-183-37-122-14-124-65-67

FCFS – *First Come First Served*

- Acessa na ordem em que as requisições são solicitadas

- Qual o deslocamento obtido?
  - 640 trilhas

SSTF – *Shortest Seek Time First*

- Seleciona a requisição com o menor tempo de *seek* em relação à posição atual do cabeçote de leitura e gravação

- Qual o deslocamento obtido?
  - 236 trilhas

SCAN

- O movimento do cabeçote de leitura e gravação inicia em uma extremidade do disco e se movimenta à outra extremidade (até à última requisição)
  - Executa as requisições na ordem desta varredura
  - Ao chegar no outro extremo, inverte o sentido desta varredura
- Conhecido como "algoritmo do elevador"
- Qual o deslocamento obtido?
  - 208 trilhas (53 até 14 e depois até 183 = (53-14)+(183-14))

C-SCAN

- Variação do algoritmo SCAN
- Procedimento idêntico ao SCAN, porém as requisições são atendidas apenas em um sentido da varredura
  - Ao final da varredura o cabeçote é posicionado no início do disco
- Qual o deslocamento obtido?
  - 183 trilhas
- Uma variação do C-SCAN é o C-LOOK
  - O cabeçote de leitura e gravação não necessita atingir o extremo do disco para voltar ao início do disco

Outras variações de SCAN

- N-step-SCAN
  - Divide a fila de requisições em um certo número de filas de tamanho N
  - Cada fila é atendida separadamente utilizando SCAN
  - Novas requisições são inseridas em filas diferentes da que está sendo atendida no momento da chegada destas requisições
- FSCAN
  - Baseada em duas filas
  - Uma fila recebe todas as novas requisições enquanto a outra é empregada para atender às requisições já feitas

Observações dos algoritmos de escalonamento

- SSTF é o método comumente empregado
- SCAN e C-SCAN apresentam um melhor desempenho em discos que possuem um grande número de acessos
- Fatores importantes
  - Quantidade e tipo de requisições
  - Organização de arquivos e diretórios no disco (sistema de arquivos)
- O algoritmo de escalonamento deve ser escrito como um módulo separado do sistema operacional para permitir sua substituição de forma fácil

Sistema de arquivos

Introdução

- O sistema de arquivos é a parte mais visível do sistema operacional
- Cria um recurso lógico a partir de recursos físicos através de uma interface coerente e simples, fácil de usar
- Mecanismo para armazenamento e acessoa dados e programas
- Duas partes básicas
  - Arquivos
    - Armazenamento de dados e de programas
  - Diretórios
    - Organização e informações sobre arquivos
- Fornece mecanismos para usuários manipularem arquivos e diretórios
- Otimiza o acesso aos dados

Introdução

- Fornece suporte a operações de E/S a diferentes dispositivos
  - Interface única para usuário e funções de E/S
- Suporta vários usuários (multiprogramação)
- Cada usuário deve ser capaz de
  - Criar, apagar, ler e alterar arquivos
  - Controlar as permissões de acesso a seus arquivos
  - Estruturar o sistema de arquivos de acordo com suas necessidades
  - Transferir dados entre arquivos
  - Realizar *backups* e recuperar arquivos em caso de problemas
  - Nomear os arquivos de forma simbólica

Operações básicas sobre arquivos

- Arquivo é um tipo abstrato de dados sobre o qual pode-se efetuar as operações de
  - Criação (*create*)
  - Escrita (*write*) e leitura (*read*)
  - Reposicionamento no arquivo (*file seek*)
  - Remoção (*delete*)
  - Abertura (*open*) e fechamento (*close*)
  - Truncagem (*truncate*)
  - Renomeação (*rename*)
  - *Appending*
  - etc.
- Correspondem a chamadas de sistema

Controle de acesso

- Importante controlar o acesso aos arquivos
  - Segurança e confidencialidade
  - Evitar acessos indevidos a arquivos
  - Sistema de identificação do usuário (login + senha)
  - Solução típica: lista de acesso e grupo
- Lista de acesso
  - Associar a cada arquivo e/ou diretório uma lista de acesso que determina que tipos de acesso são permitidos para cada usuário
  - Problema: o tamanho da lista
  - Solução simples:
    - Criar classes de usuário
      - Proprietário, grupo, universo
    - Criar tipos de acesso
      - *Read, write, modify, execute*

Exemplo: UNIX

- Cada objeto oferece 3 bits (*rwx*) para três domínios diferentes (*owner, group* e *others*)
- Flexível
  - Quando um usuário pertence a vários grupos, ele é identificado por um grupo primário e os demais grupos estão cadastrados em */etc/groups*
- Estrutura interna de arquivos
  - Forma como os dados estão dispostos no arquivo
  - Cada tipo de arquivo possui uma estrutura interna apropriada à sua finalidade
  - Alguns sistema operacionais suportam nomes de arquivos onde o tipo é indicado
    - Windows (.doc, .pdf, .xls, etc.)

Métodos de acesso

- Forma pela qual o conteúdo de um arquivo é acessado
- Estrutura lógica de um arquivo em função da forma de acesso
  - Registro: coleção de campos que mantém uma relação entre si
- Métodos básicos de acesso
  - Acesso sequencial
    - Acesso feito através de primitivas *read* e *write*
    - Cada *read* retorna ao processo os dados seguintes àqueles que foram lidos na chamada anterior
    - Método não adequado quando tem-se um número muito grande de acessos e atualizações de arquivos

Métodos de acesso

- Acesso relativo
  - Provê uma chamada de sistema específica para indicar o ponto em que um arquivo deve ser lido/escrito
  - Implementado através da abstração de "posição corrente no arquivo"
- Outros tipos de acesso
  - Seqüencial indexado
  - Indexado
  - Direto
  - *Hash*
  - etc.

Implementação de arquivos

- Para cada arquivo tem-se uma estrutura de dados
- Descritor de arquivo é um registro que mantém informações sobre o arquivo
  - Nome do arquivo
  - Tamanho em bytes
  - Data e hora da criação, do último acesso, da última modificação
  - Identificação do usuário que criou o arquivo
  - Listas de controle de acesso
  - Local do disco físico onde o conteúdo do arquivo foi colocado
  - etc

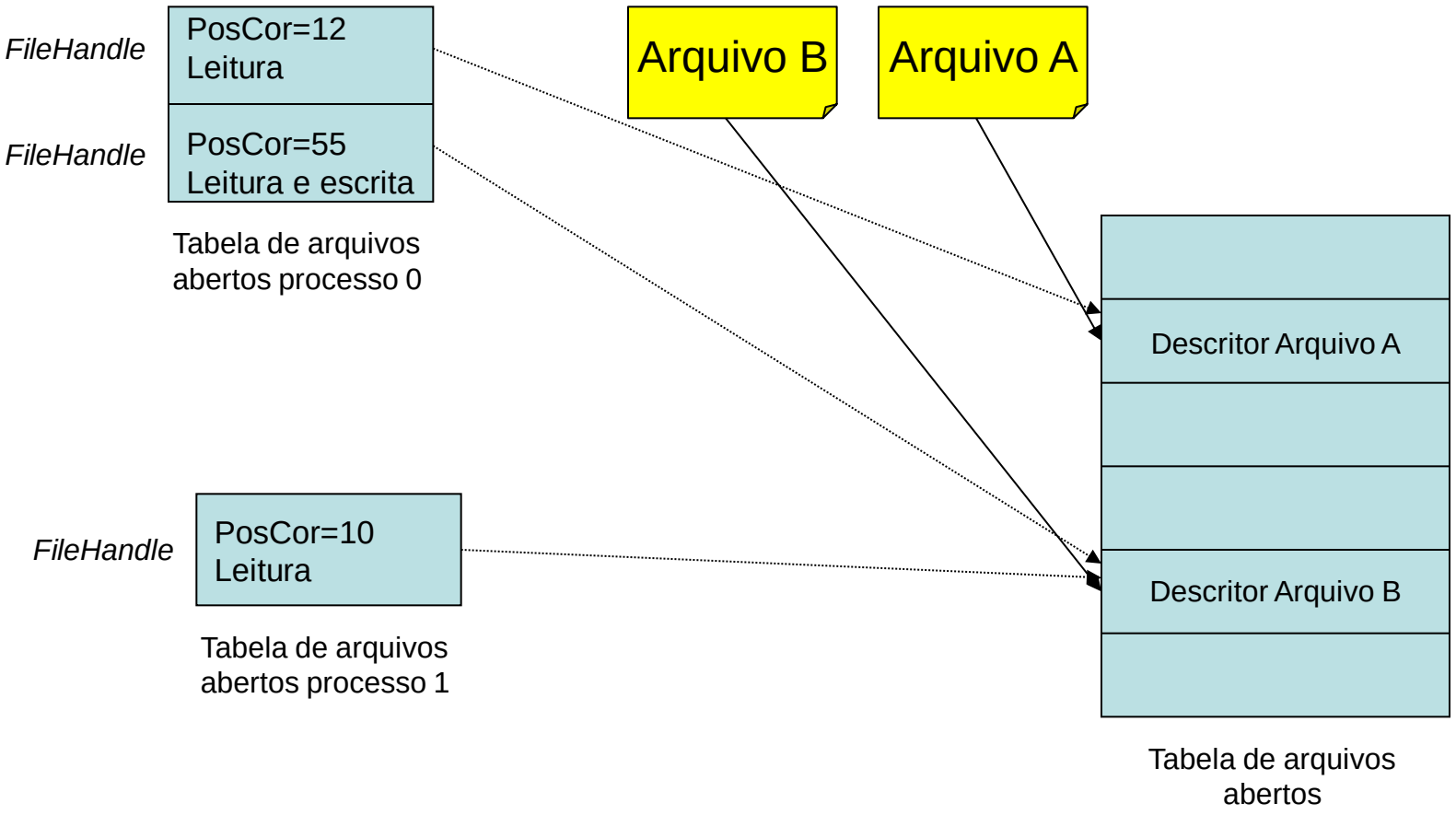
Tabelas de descritores de arquivos

- São armazenadas no próprio disco lógico (partição)
- Desempenho ruim
  - Acesso lento para ler o descritor de arquivos
  - Solução: manter o descritor em memória enquanto o arquivo estiver em uso
- Sistema de arquivos mantém os descritores de arquivos em memória em uma estrutura de dados do sistema operacional
  - Tabela de descritores de arquivos abertos - TDAA

Tabelas de arquivos abertos por processo

- Dois tipos de informações
  - Não variam conforme o processo que está acessando o arquivo
    - Ex. tamanho do arquivo
  - Dependem do processo que está acessando o arquivo
    - Ex. posição corrente
- Informações dependentes do processo são armazenadas em uma tabela à parte, mantida pelo processo – TAAP
  - Ex. posição corrente no arquivo, tipo de acesso e apontador para a entrada correspondente na TDAA
- Entrada na TAAP serve para referenciar o arquivo
  - *File handle*

Emprego conjunto das tabelas TAAP e TDAA



Gerenciamento do dispositivo de armazenamento

- Gerenciamento:
  - Relação número de setores do disco que compõem um bloco
    - Não necessita ser 1:1
  - Alocação de blocos no disco
  - Recuperação de blocos liberados
  - Localização de dados no disco
- Como alocar espaço em disco de forma que os arquivos sejam armazenados de forma eficiente e que permitam acesso rápido
- Três métodos:
  - Contíguo
  - Encadeado
  - indexado

Alocação contígua

- Arquivo é uma seqüência de blocos contíguos alocados no momento da criação
- Endereços no disco são lineares
  - Bloco lógico *i* e *i+1* são armazenados em seqüência, fisicamente
  - Reduz a necessidade de *seek*, já que blocos estão na mesma trilha
- Arquivo é descrito através de uma entrada na forma
  - Bloco físico inicial
  - Tamanho do arquivo em blocos

Problemas com alocação contígua

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

Alocação encadeada

- Soluciona problemas da alocação contígua
- Alocação é baseada em uma unidade de tamanho fixo (bloco)
  - Tipo paginação
- 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

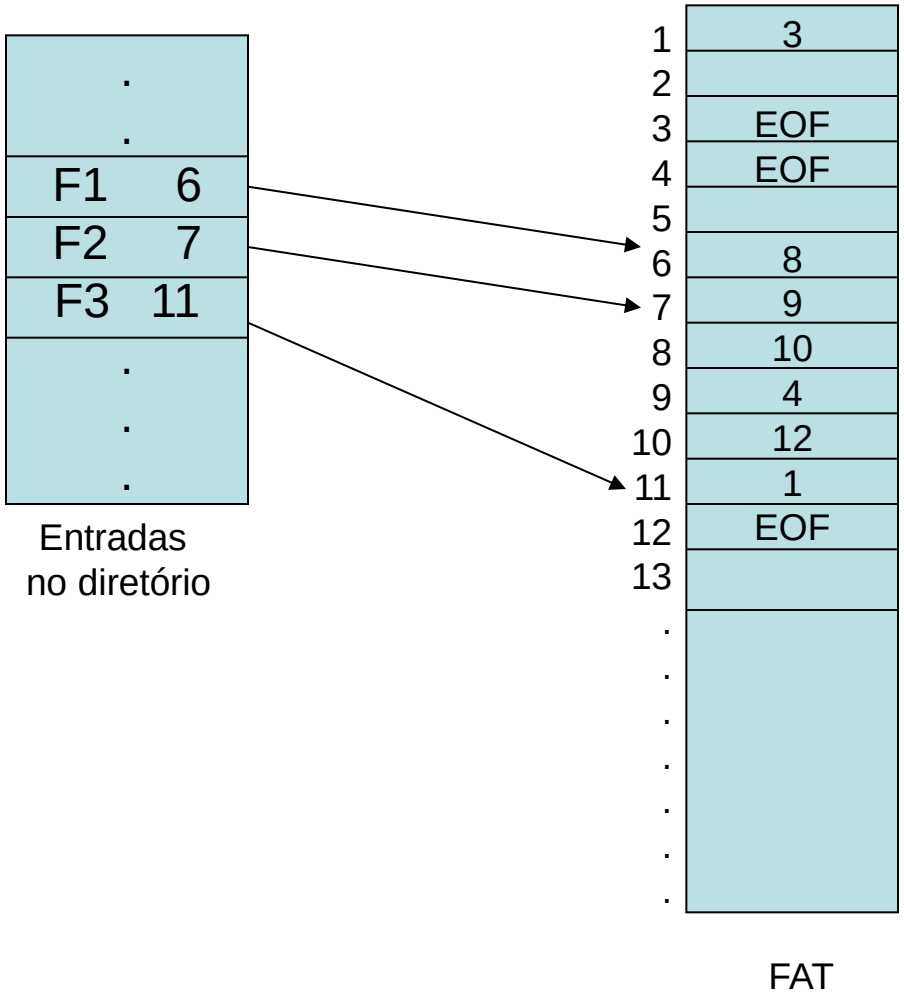
Prós e contras da alocação encadeada

- Elimina a fragmentação externa, porém gera fragmentação interna
- Arquivos podem crescer indefinidamente, já que não há uma relação física entre blocos físicos e lógicos
- O acesso a um bloco *i* implica percorrer a lista encadeada
  - Afeta o desempenho
- Adequado para acesso seqüencial a arquivos
- Espaço necessário no bloco para armazenamento de ponteiros
  - Ex. para acessar 4 GBytes, necessita 2^{32} ponteiros (4 bytes), ou seja, 0.78% de um setor de 512bytes
- Confiabilidade
  - Erro provoca a leitura / escrita em bloco pertencente a outro arquivo

FAT

- *File Allocation Table* (FAT)
  - Esquema adotado pelo DOS
- Unidade de alocação é o cluster (conjunto de setores)
  - Arquivo é formado por um conjunto de clusters
- A FAT é uma tabela de encadeamento de clusters
  - Uma entrada na FAT para cada *cluster* do disco (sistema de arquivos)
  - Composta por um ponteiro (endereço de um cluster)
  - Arquivo é descrito por uma seqüência de entradas na FAT, cada entrada apontando para a próxima entrada (alocação encadeada)

FAT

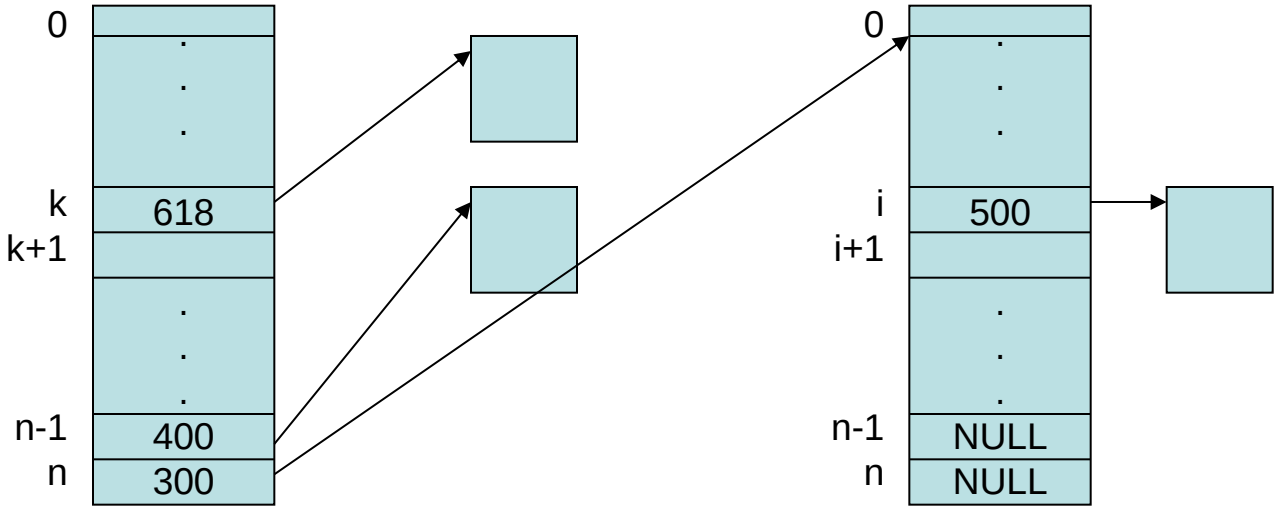

.
.
F1 6
F2 7
F3 11
.
.
.
Entradas
no diretório
1 3
2
3 EOF
4 EOF
5
6 8
7 9
8 10
9 4
10 12
11 1
12 EOF
13
FAT

Alocação indexada

- Busca resolver o problema de ponteiros esparramados pelo disco, que a alocação encadeada provoca
- Mantém, por arquivo, um índice de blocos que o compõe
- O índice é mantido em um bloco
- Diretório possui um ponteiro para o bloco onde está o índice associado a um determinado arquivo
- Vantagem
  - Permite acesso randômico a blocos independentes de sua posição relativa no arquivo
- Desvantagem
  - Tamanho máximo do arquivo é limitado pela quantidade de entradas suportadas pelo bloco

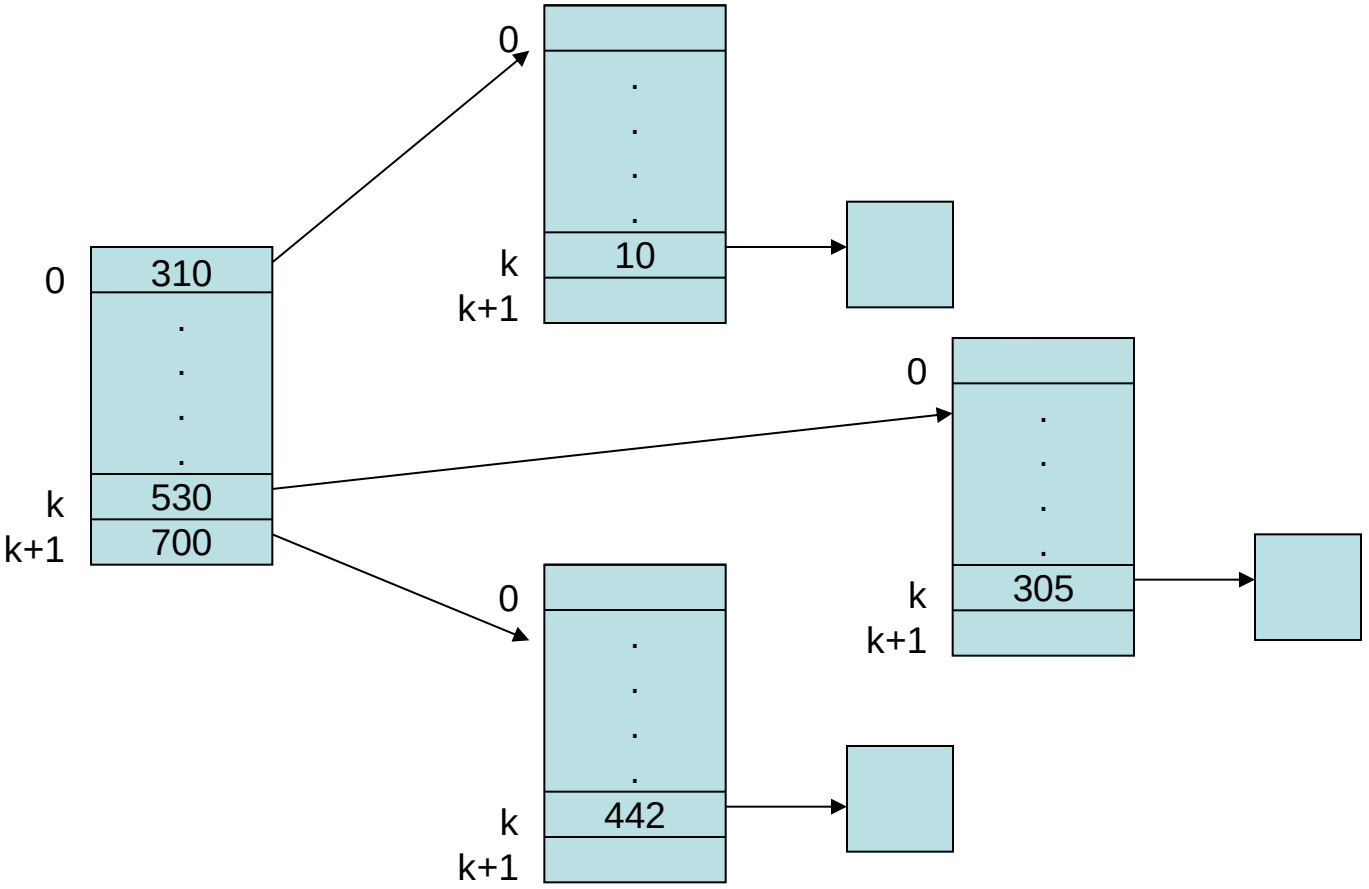
Variações em alocação indexada

- Três métodos:
  - Encadeado
  - Multinível
  - Combinado
- Método encadeado
  - A tabela de índice mantém ponteiros para os blocos que compõem o arquivo, à exceção da última entrada, que mantém um ponteiro para outro bloco, onde a tabela de índices continua.



Método multinível

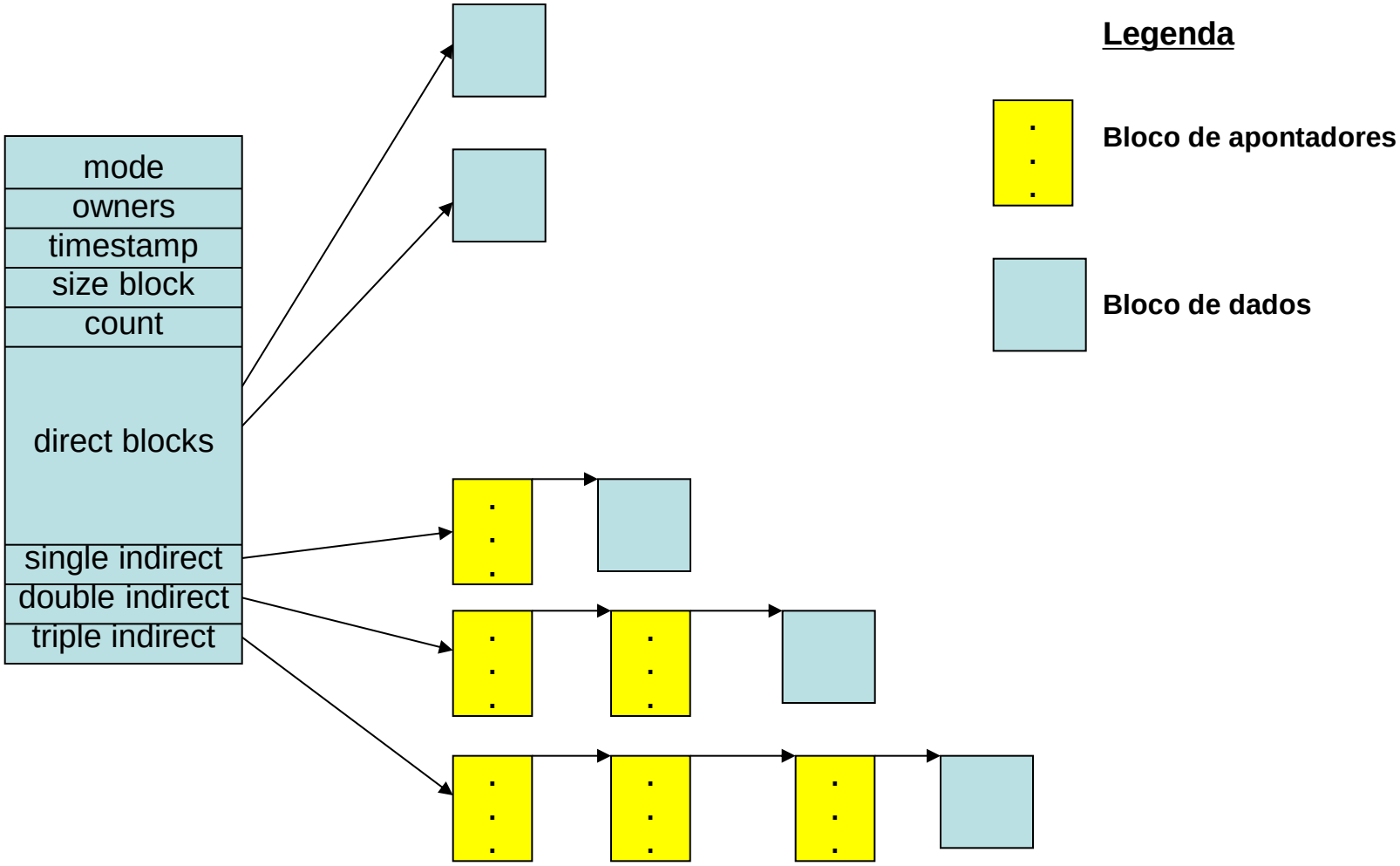
- Mantém uma tabela de tabela de índices
  - Não resolve completamente o problema do limite



Método combinado

- Combina os métodos encadeado e multinível em uma única estrutura de dados
- Essencialmente, cria diferentes níveis de indireção na indexação
  - Apontadores direto
    - Endereçam diretamente blocos de dados
  - Apontadores de indireção simples
    - Apontam para blocos que possuem apontadores diretos
  - Apontadores de indireção dupla
    - Apontam para blocos que possuem apontadores de indireção simples
  - E assim por diante...

Método combinado: estruturas de i-node do UNIX



Método combinado: UNIX

- Para acesso direto, o UNIX reserva 10 endereços diretos de blocos no I-node
  - Desta forma, se o bloco possuir um tamanho de 1 Kbyte, um determinado arquivo poderá crescer até ocupar um tamanho de 10 Kbytes, ocupando os blocos de acesso direto, apenas
- Se o arquivo desejar crescer mais, então será utilizado o número de bloco para acesso *single indirect*
  - Isto significa que esta posição dentro do I-node será ocupada por um endereço de bloco de ponteiros para blocos de dados e não para um simples bloco de dados
  - Assim o arquivo (do exemplo) poderá crescer até um tamanho de 266 Kbytes

Método combinado: UNIX

- Da mesma forma, se o arquivo desejar crescer mais, então será utilizado o número de bloco para acesso *double indirect*
  - Esta posição será ocupada pelo endereço de um bloco de ponteiros, onde cada entrada apontará para outro bloco de ponteiros que, por sua vez, cada entrada apontará para um bloco de dados
  - Assim o arquivo poderá crescer até um tamanho de 64.2 Mbytes
- Pode-se utilizar ainda o *triple indirect* do I-node
  - Gera três níveis de indireção
  - Desta forma, o arquivo do exemplo poderá chegar ao seu tamanho máximo com 16.6 Gbytes

Comparação entre os tipos de alocação

- Alocação contígua
  - Só armazena endereço do primeiro bloco
  - Dados são contíguos
  - Acesso randômico é possível
- Alocação encadeada
  - Armazena endereço do primeiro bloco
  - Problema de *seek*
  - Não recomendado para acesso randômico
- Alocação indexada
  - Análise mais complexa
  - Depende da estrutura de índice e do tamanho de arquivos

Múltiplos sistemas de arquivos

- Evolução tecnológica
  - Diferentes sistemas de arquivos
- Desafio
  - fazer o sistema operacional suportar simultaneamente diferentes sistemas de arquivos
- Cada partição possui um único sistema de arquivos
- Tabela com descritores virtuais de arquivos abertos
  - Descritor virtual
    - Apontador para uma estrutura "tipo de sistema de arquivos"
    - Apontador para o descritor real do arquivo
- Descritor do sistema de arquivos
  - ou a estrutura de tipo, mantém uma lista de apontadores para rotinas que implementam a interface padrão para cada sistema de arquivos

Organização da cache de disco

- Manter na memória blocos do disco
- A cache de disco está na memória e é controlada pelo sistema operacional
- Em uma requisição de E/S verifica se bloco está na cache
  - Se sim: realiza o acesso a partir da cópia em memória
  - Se não: realiza o acesso a partir do disco e carrega o bloco para a cache
- Políticas de atualização da cache
  - Posterga ao máximo
  - Atualiza a cada intervalo de tempo
  - Atualiza imediatamente no disco
  - Atualiza imediatamente apenas informações sensíveis à consistência do sistema de arquivos

Política de substituição

- O que fazer quando um novo bloco deve ser inserido na cache e não há espaço livre?
  - Usa-se LRU (*Least Recently Used*)
  - Facilmente implementada com uma lista duplamente encadeada
  - Quando o bloco é acessado ele é removido de sua posição na lista e colocado no início
  - Todo bloco novo (acessado pela primeira vez) também é inserido no início da lista
  - O bloco menos recentemente usado é o último da lista

Gerência de espaço livre

- Necessidade de alocar espaço livre do disco a arquivos
- Usa-se um mapa de bits
  - Forma simples de gerenciar o espaço em disco
  - A cada bloco do disco está associado um bit indicando se o bloco está livre ou ocupado
- Lista de blocos livres
  - Os blocos livres estão em uma lista
  - A lista é mantida no próprio disco
    - Problema: o tamanho da lista
    - Solução alternativa: manter uma lista de áreas livres e não de blocos livres

Diretório

- Forma de organização de arquivos no disco
- Sistema de arquivos oferece
  - Partição
  - Diretório
- Partição divide um disco físico em discos lógicos
  - Cada partição é autocontida, isto é, todas as informações para acesso aos arquivos das partições estão contidas na própria partição
    - Diretórios e subdiretórios
    - Descritores de arquivos da partição
    - Blocos de dados
    - Lista de blocos livres

Diretório

- Um setor (bloco) especial do disco informa quais são as partições e quais parcelas do disco que a partição ocupa
- Mas não resolve a organização de arquivos

- Diretório é uma estrutura de dados que contém informações sobre arquivos
  - Atributos
  - Localização
  - Propriedades
- Duas formas de diretórios
  - Linear
  - Em árvore

Diretório linear

- Mais simples
- O diretório corresponde a uma lista de todos os arquivos do disco
- Desvantagem
  - Problema de nomeação a agrupamento
  - Dois ou mais usuários não podem ter arquivos com o mesmo nome
- Diretório linear a dois níveis
  - Existe um diretório principal que contém uma entrada para cada usuário cadastrado no sistema
  - Entrada corresponde a um subdiretório

Diretório em árvore

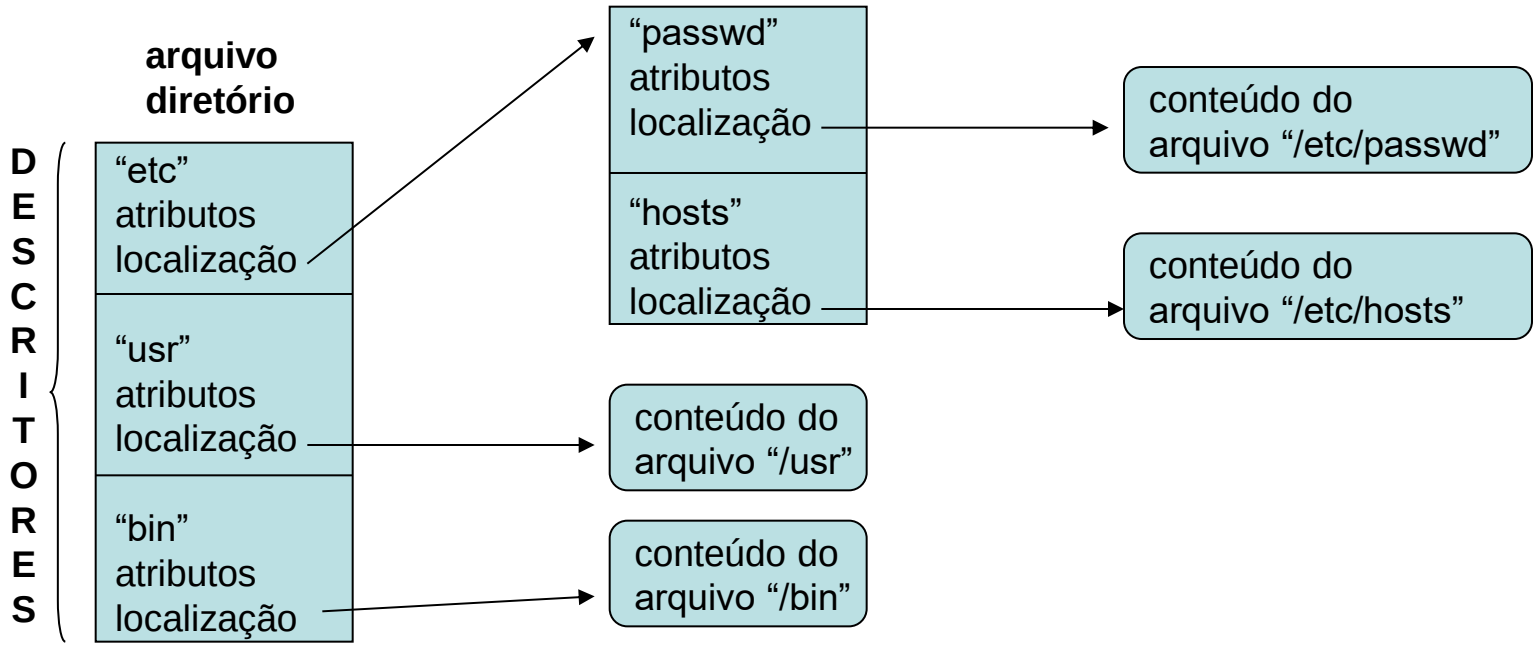
- Permite aos usuários criarem seus próprios subdiretórios
- Qualquer arquivo pode ser identificado através de seu *path*
  - Conceito de:
    - Diretório corrente
    - Caminho absoluto
    - Caminho relativo
- Vantagens
  - Procura eficiente por arquivos
  - Possibilidade de agrupamento de arquivos
- Desvantagens
  - Compartilhamento de arquivos
    - Questão é: copiar ou não copiar arquivos a compartilhar?

Diretório estruturado em grafo

- Estrutura em árvore que resolve o problema do compartilhamento de arquivos
- Pode ser implementado usando *aliases*
- *link* é uma forma comum de *alias*
  - Ponteiro para outro arquivo ou subdiretório
  - *link* simbólico
- *link* é uma entrada na estrutura de diretório
- Remover um *link* implica apenas remover a sua entrada na estrutura de diretório, não o arquivo que o aponta

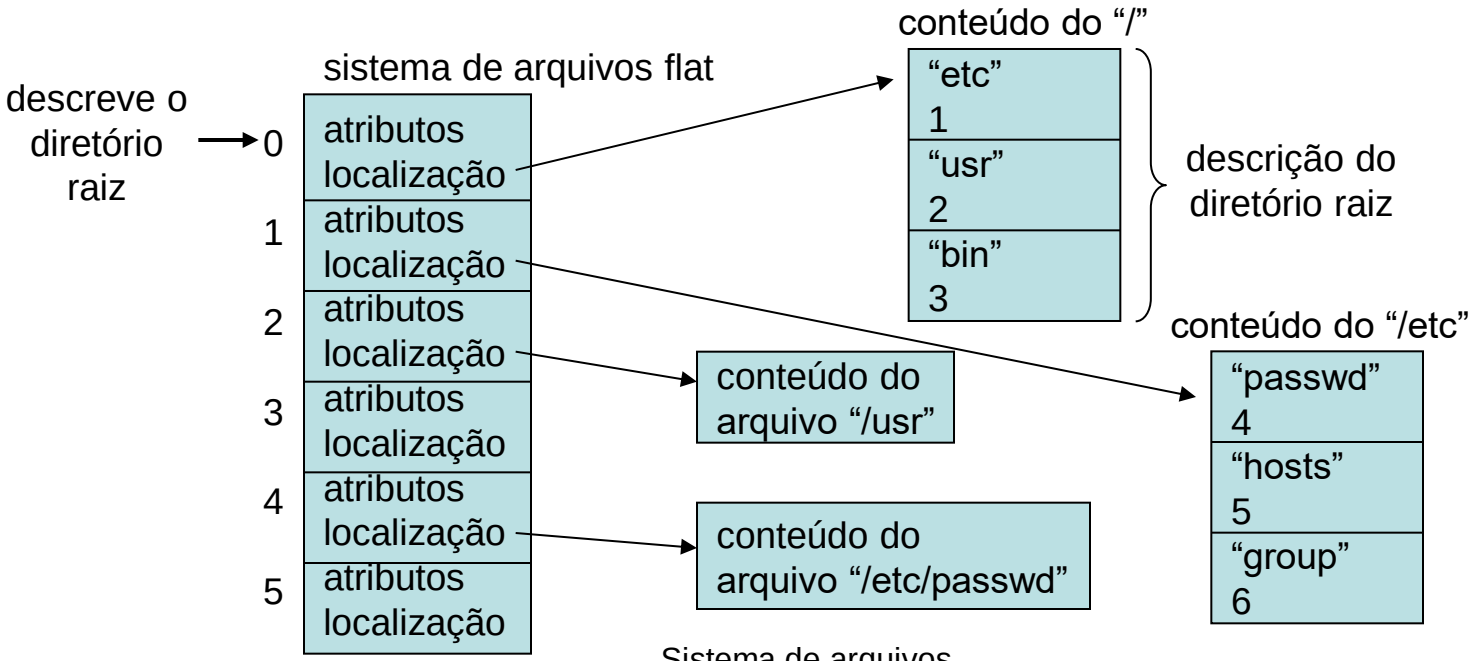
Implementação de diretórios

- Através de descritores de arquivos
  - Conteúdo de um arquivo diretório corresponde aos descritores dos arquivos e subdiretórios contidos neste diretório



Implementação de diretórios

- Através de vetores de descritores
  - Composto por um conjunto de blocos da partição exclusivos para o armazenamento de descritores de arquivos e de subdiretórios
  - Cada descritor é identificado pelo número de partição e pela posição nesse vetor
  - O primeiro descritor descreve o diretório raiz



Implementação de diretórios

- Através de tabelas
  - Um diretório é uma tabela
  - Formas de implementação mais utilizadas
    - Lista não ordenada
    - Lista ordenada
    - Tabela de dispersão (tabela *hash*)
- As vantagens e desvantagens residem no paradigma
  - Simplicidade *versus* desempenho

Referências Bibliográficas

DEITEL, H. M.; DEITEL P. J. e CHOFFNES, D. R. Sistemas Operacionais, 2 ed. Pearson Prentice Hall, 2005.

OLIVEIRA, R. S.; CARÍSSIMI, A. S. e TOSCANI, S. S. Sistemas Operacionais, 3 ed. Ed. Sagra-Luzzatto. 2004.

TANENBAUM, A. S. Sistemas Operacionais Modernos. 2 ed. Pearson Prentice Hall, 2003.