

Sistemas Operacionais

Gerência de Arquivos

Edson Moreno

edson.moreno@pucrs.br

<http://www.inf.pucrs.br/~emoreno>

Sumário

- Conceituação de arquivos
- Implementação do sistemas de arquivo

Introdução

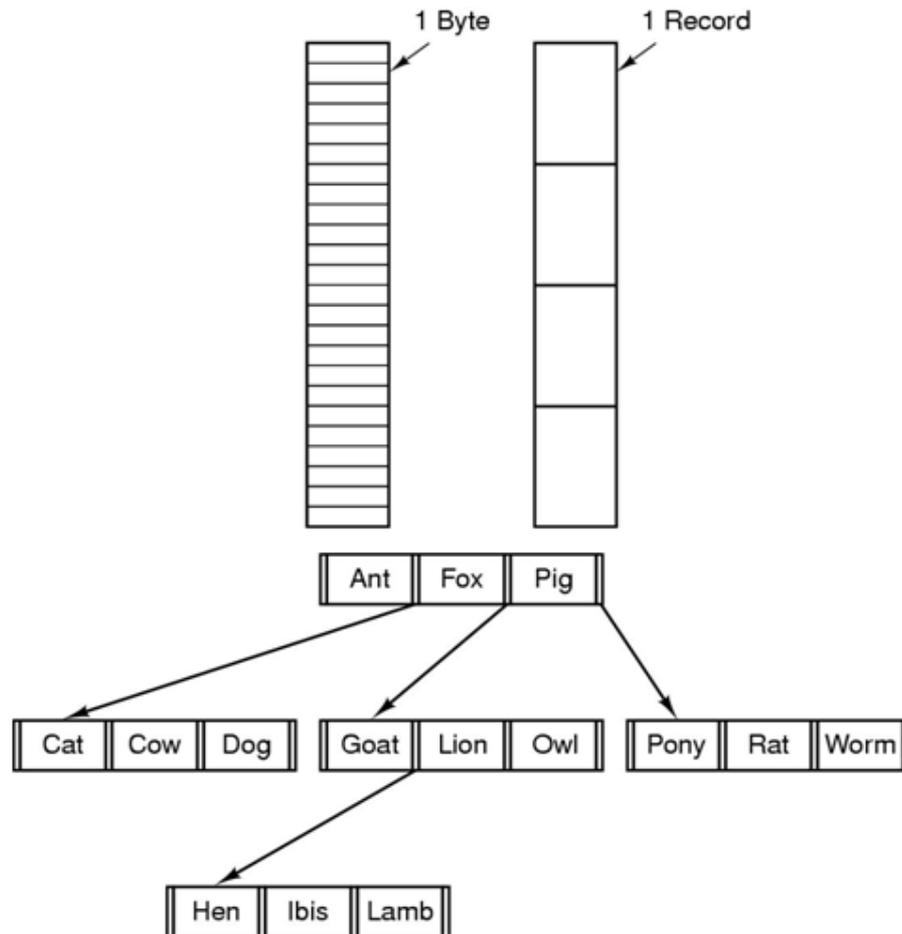
- Sistema de memória
 - Provê solução para a execução de processos
 - Disponibiliza espaço para a manipulação de dados
 - Mantidos enquanto processo existe
 - Não atende a necessidade de todas aplicações
 - Pode-se requerer além da capacidade da memória virtual
 - E.g. Um sistema corporativo
 - Pode-se precisar garantir persistência dos dados qdo processo termina
 - E.g. Um banco de dado
 - Pode-se precisar que diferentes processos acessem a mesma informação
 - E.g. Uma página web
- Solução
 - Emprego de sistema de arquivos
 - Permitir armazenar grandes quantidades de informações
 - Informação permanecem após término do processo
 - Permitir o acesso por distintos processos

Arquivos

- Entidades lógicas
 - Criadas por processos
 - Mapeada pelo S.O. em dispositivos físicos
- Características
 - Contém informações definidas pelo proprietário
 - Dados e/ou programas
 - Possui nome pelo qual é referenciado
 - Dependendo do SO (case-sensitive, tamanho, extensões)
 - Manipulação suportada pelo SO
 - Chamadas de sistema

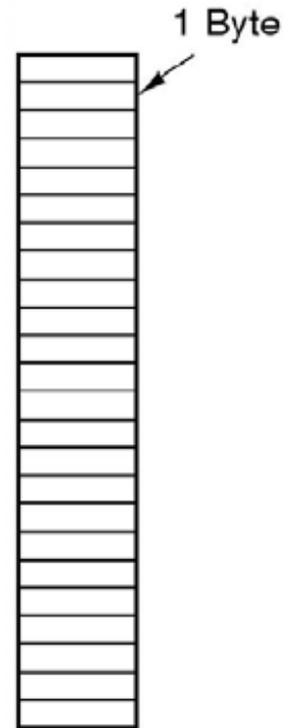
Arquivos

- Estrutura de arquivos
 - Seqüência de bytes
 - Seqüência de registros
 - Árvore



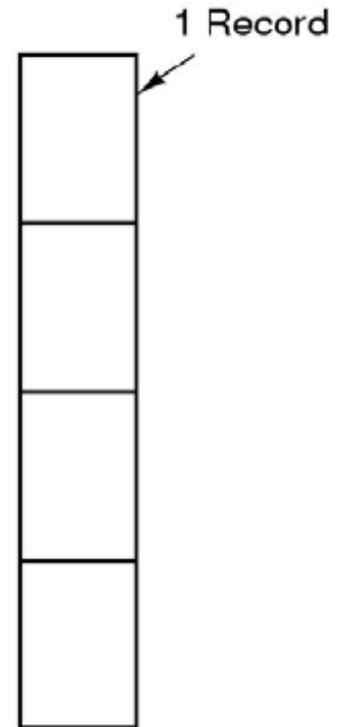
Arquivos

- Estrutura de arquivos
 - Seqüência de bytes
 - Sistema operacional não sabe o que o arquivo contém.
 - Aplicação em nível de usuário impoem o significado
 - Garante flexibilidade
 - Nomeação e conteúdo do arquivo definido pelo aplicativo
 - Abordagem assumida por SOs tal como Windows e Unix



Arquivos

- Estrutura de arquivos
 - Seqüência de registros
 - Formato definido internamente pelo SO
 - Arquivo definido como uma sequencia de registros de tamanho fixo
 - Operação de leitura / escrita opera sobre um registro
 - Historicamente
 - Sistemas operacionais baseados em registros de 80 caracteres pois trabalhava com cartão perfurado de 80 colunas

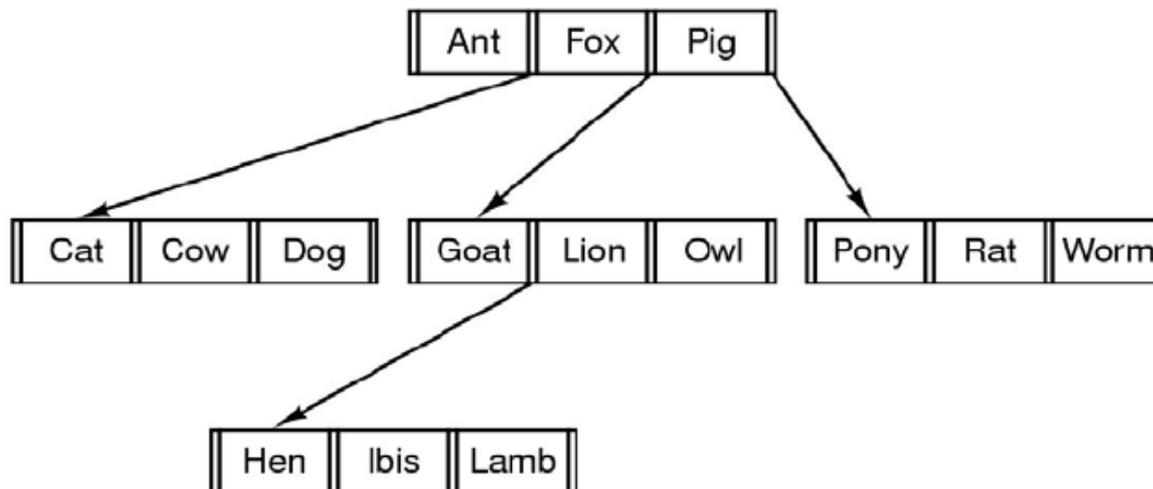


Arquivos

- Estrutura de arquivos

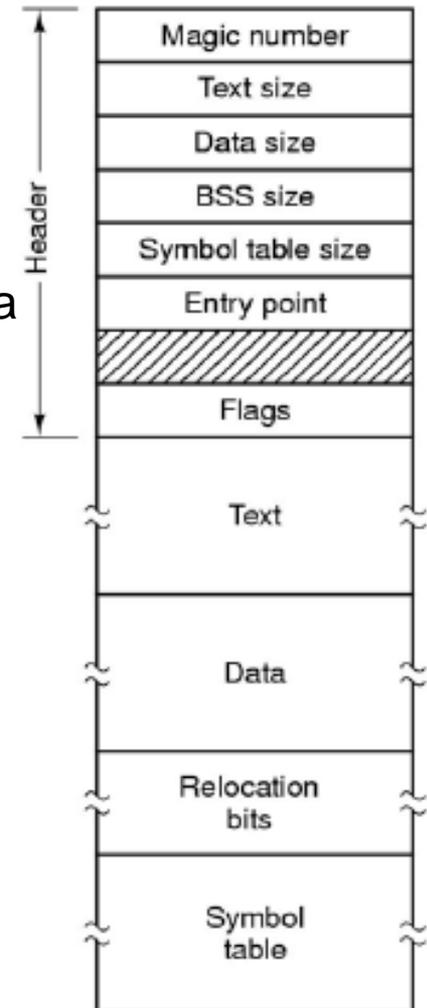
- Árvore

- Arquivo constituído de uma árvore de registros de tamanho variável
 - Busca de registros usando campo-chave
 - Campo reside em posição fixa do bloco
 - Ordenação da árvore feita pelo campo-chave garante busca mais rápida
 - Foco não está na busca do próximo registro, mas sim no registro que contenha uma chave específica



Arquivos

- Tipos de arquivo
 - Regulares (arquivos comuns)
 - Contém informações dos usuários
 - Podem ser ASCII ou binários
 - Arquivos binários normalmente contém estrutura interna conhecida
 - Exemplo: Arquivos executáveis : formato unix →
 - Diretórios
 - Arquivos do sistema
 - Mantém a estrutura do sistema de arquivos
 - Links
 - Utilizados para compartilhamento de arquivos
 - Especiais
 - Utilizados para acesso à dispositivos de E/S
 - Caracter: terminais, impressoras, rede
 - Bloco: disco



Arquivos

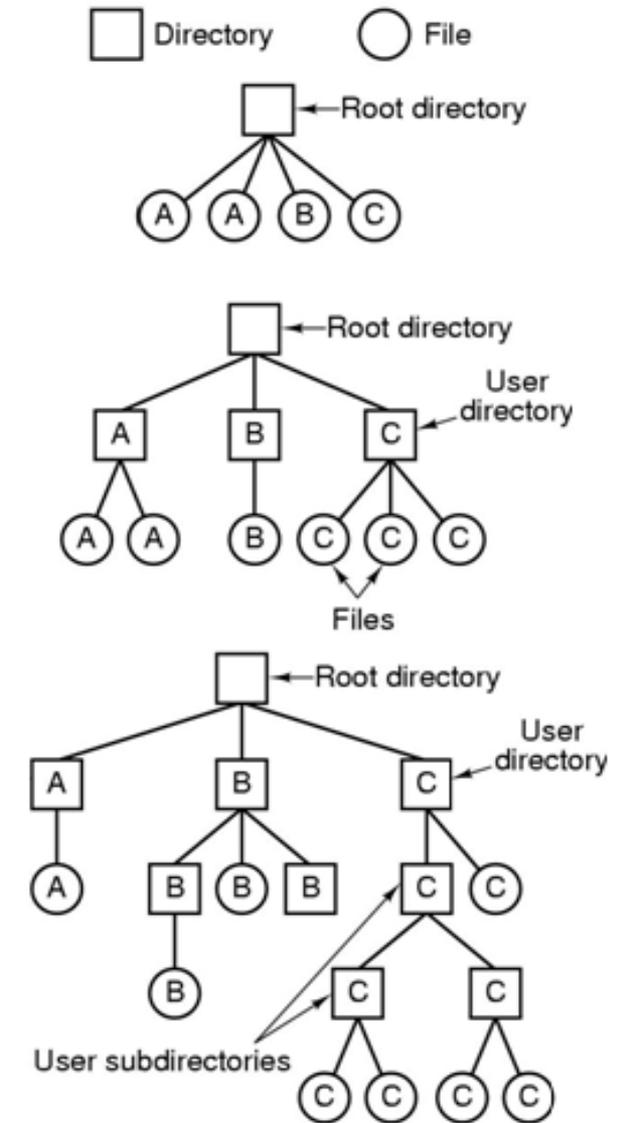
- Tipos de acesso aos arquivos
 - Dita como o sistema de arquivos recupera registros
- Acesso seqüencial
 - Leitura dos dados (bytes/registros) se dá a partir do início
 - Não permite acesso fora de ordem
 - Primeiros SOs só armazenavam arquivos em fitas magnéticas
- Acesso aleatório
 - Leitura dos dados (bytes/registros) pode ser feito fora de ordem
 - Essencial para sistemas tal como base de dados
 - Sistemas de arquivo atuais empregam este tipo

Diretórios

- Características
 - Contém informações que permitem acessar aos arquivos
 - Possui várias entradas
 - Uma por arquivo
- Organização de diretórios
 - Cada entrada contém o nome e os atributos do arquivo
 - Cada entrada contém o nome do arquivo e um ponteiro para uma estrutura de dados com os atributos

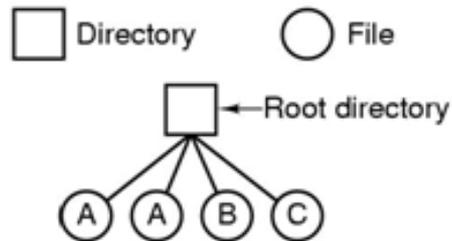
Diretórios

- Tipos de diretório:
 - Sistema de diretório em nível único
 - Sistema de diretório com dois níveis
 - Sistema de diretório hierárquico



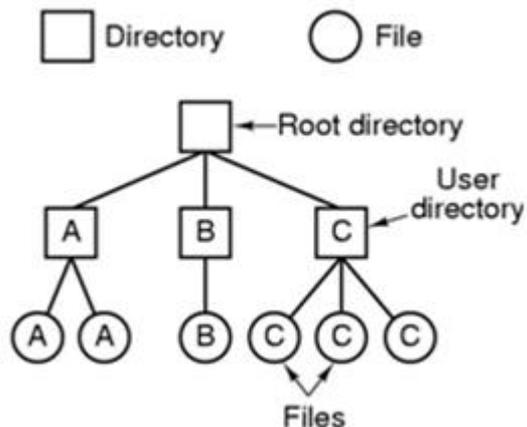
Diretórios

- Tipos de diretório:
 - Sistema de diretório em nível único
 - Implementação mais simples
 - Diretório raiz
 - Contém os arquivos de todos os usuários
 - Não permite que arquivos com mesmo nome...
 - Empregado em sistemas embarcados
 - Telefones, câmeras fotográficas, etc



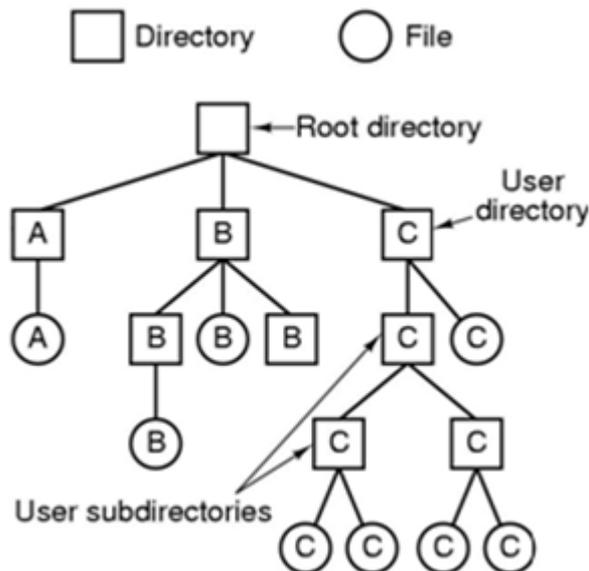
Diretórios

- Tipos de diretório:
 - Sistema de diretório em dois nível único
 - Cada usuário possui um diretório
 - Nomes dos arquivos tem de ser garantidos no diretório do usuário
 - Deve haver um diretório a mais, indexado pelo nome do usuário



Diretórios

- Tipos de diretório:
 - Sistema de diretório hierárquico
 - Adotado pela maioria do sistemas operacionais
 - Permite que cada usuário crie seus arquivos e (sub)diretórios
 - Cada arquivo possui um caminho, desde a raiz

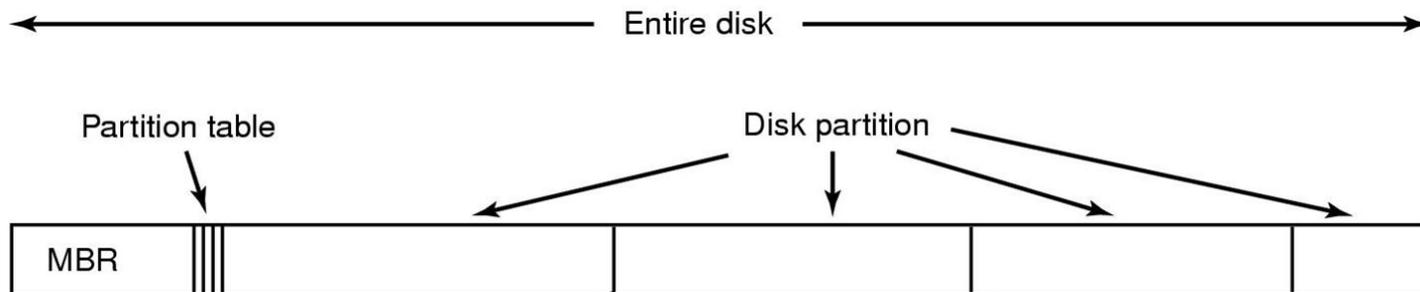


Sumário

- Conceituação de arquivos
- ***Implementação do sistemas de arquivo***

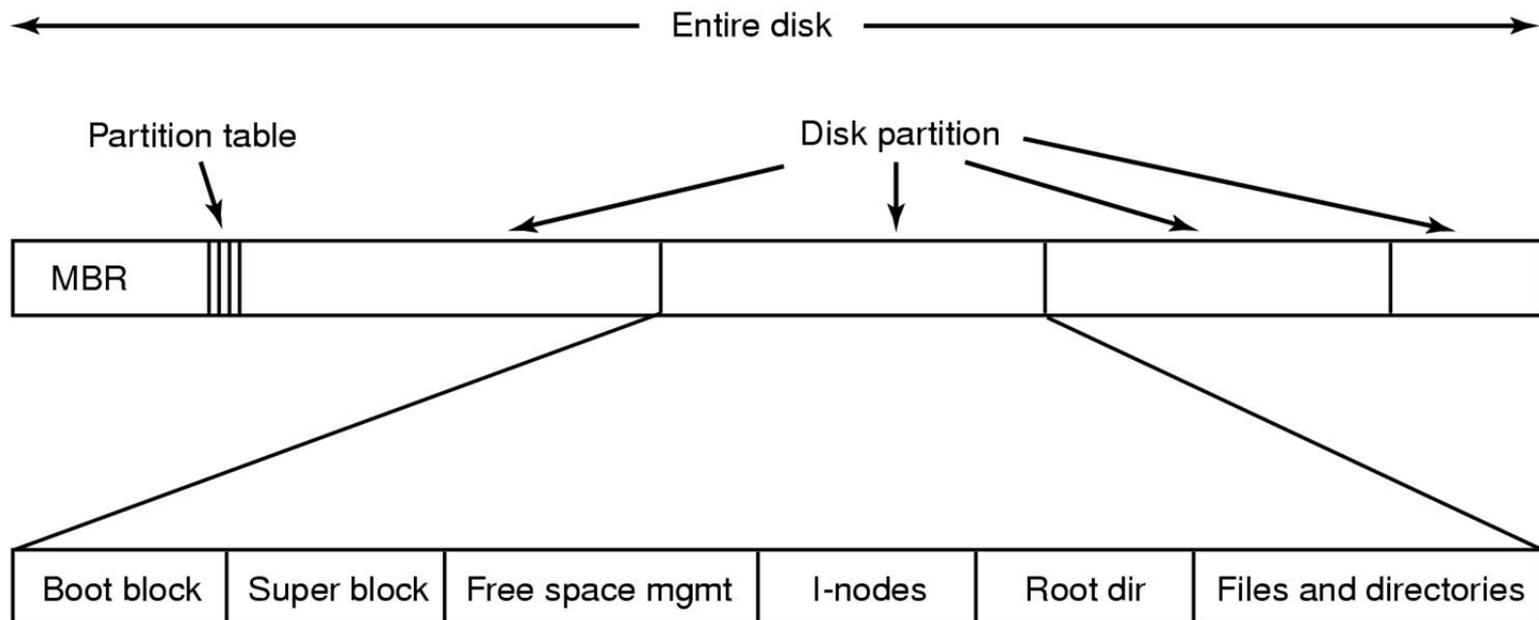
Sistema de Arquivos

- Visão do implementador
 - Garante a gerência dos arquivos pelo SO
 - Define como os arquivos serão estruturados / nomeados / acessados
- Caracterização
 - Sistema armazenado em disco
 - Específico para cada partição do disco



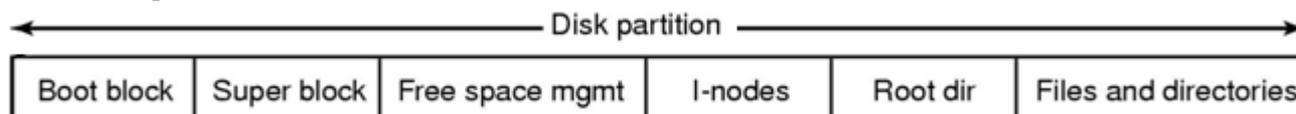
Sistema de arquivos

- *Master boot record (MBR)*
 - Presente no setor 0 do disco
 - Usado para inicialização do computador
 - BIOS lê e executa o MBR que
 - Localiza partição ativa na tabela de partições
 - Carrega o Sistema Operacional



Sistema de arquivos

- Componentes de *cada partição*
 - **Bloco de inicialização (boot block)**
 - Quando executado, deve carregar o SO
 - Presente em todas partições mesmo naquelas sem SO
 - **Superbloco**
 - Contem os principais parâmetros sobre o sistema de arquivo
 - “Número mágico” que identifica o sistema de arquivo
 - Número de blocos presentes no sistema de arquivos
 - Carregado na memória quando
 - Computador é inicializado
 - Utilizado pela primeira vez
 - **Gerencia de espaço livre** : Mapa de bits / lista de ponteiros
 - **Diretório raiz**: Define o topo da árvore
 - **Arquivos e diretórios**: restante da estrutura de dados da partição



Sistema de Arquivos

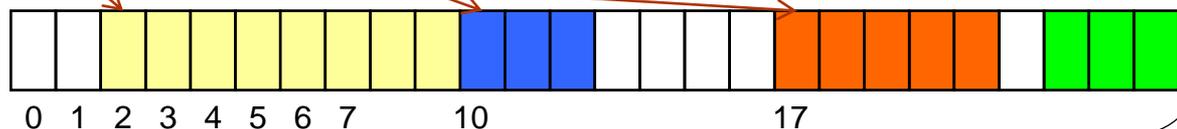
- Armazenamento de arquivos
 - Exige o controle da relação entre blocos de disco e arquivos
 - Variam de SO para SO
 - Tipos a serem explorados
 - Alocação contígua
 - Alocação por lista encadeada
 - Alocação por lista encadeada usando tabela de memória
 - i-node

Sistema de Arquivos

- Armazenamento de arquivos
 - Alocação contígua
 - Consiste em armazenar um arquivo em blocos contíguos
 - Implementação bem simples
 - Endereço do bloco onde inicia o arquivo
 - Número de blocos que são empregados
 - Problemas
 - Alocação de novos arquivos em espaços livres
 - Pode gerar fragmentação externa
 - E se o tamanho de arquivo armazenado precisar ser estendido

arquivo inicio #blocos

readme.txt	010	003
prova.doc	002	008
Aula.pdf	017	005

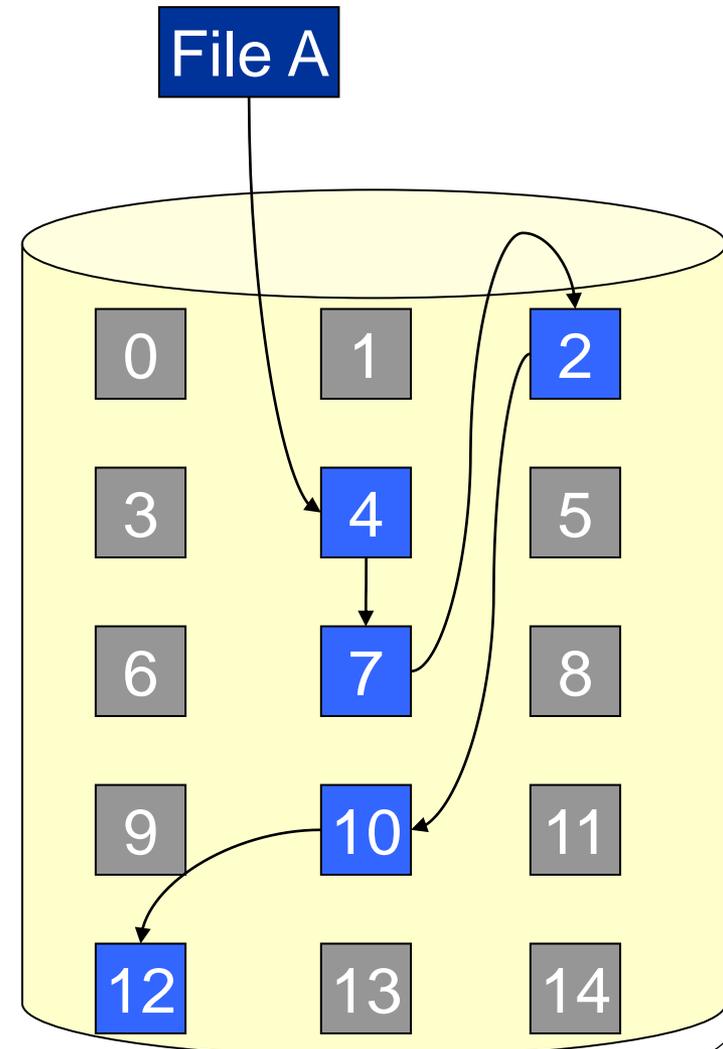
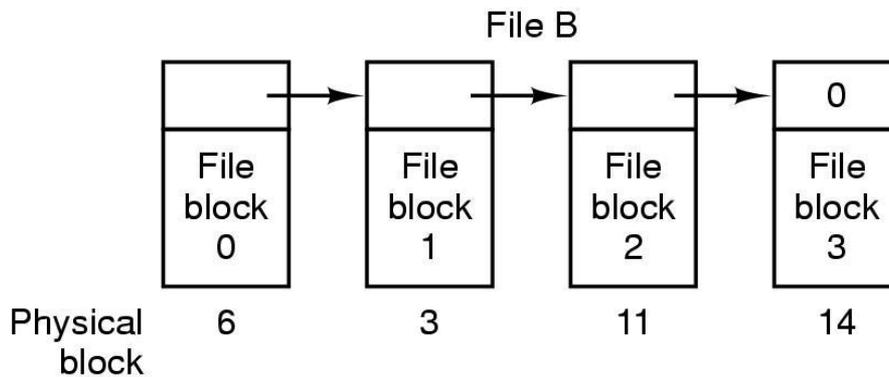
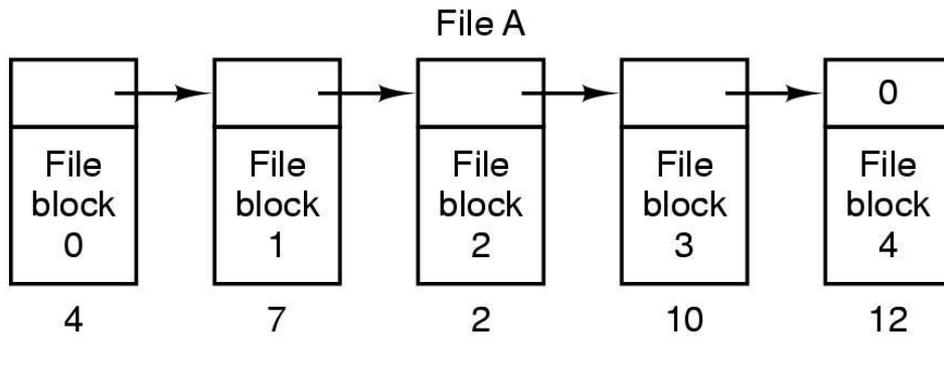


Sistema de Arquivos

- Armazenamento de arquivos
 - Lista encadeada
 - Só é preciso conhecer o 1o. Bloco
 - Primeira palavra utilizada como ponteiro para próximo bloco
 - Último bloco aponta para endereço nulo
 - Não causa fragmentação externa
 - Problemas
 - Acesso aleatório é lento pois os blocos tem de ser sequencialmente buscados
 - Para se chegar ao bloco n , deve-se passar por $n-1$ blocos antes
 - Vinculo está armazenado em disco
 - Bloco não é usado somente para dados
 - Exemplo
 - Bloco de 1kbytes e endereçamento de 16 bits
 - 1022 bytes armazenam informações
 - 2 bytes endereçam o próximo bloco

Sistema de Arquivos

- Armazenamento de arquivos
 - Lista encadeada

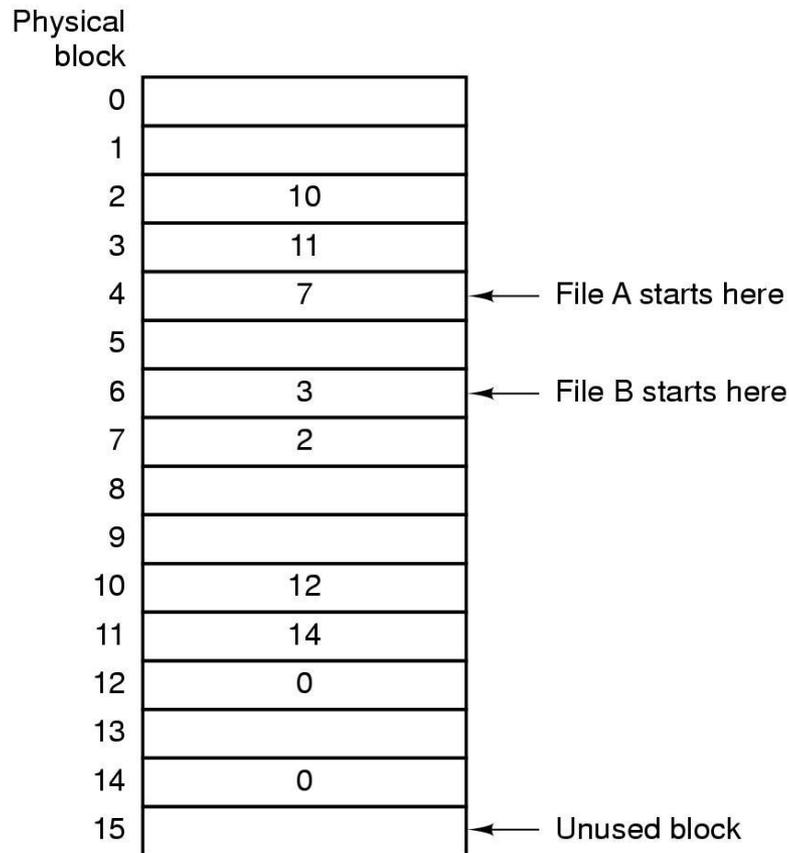


Sistema de Arquivos

- Armazenamento de arquivos
 - Lista encadeada com tabela na memória
 - Otimização do método baseado em lista encadeada
 - Cria tabela em memória (FAT: *File Allocation Table*)
 - Cada entrada da tabela contem o apontador de blocos de disco
 - Acelera o acesso aleatório
 - Identificação de um bloco não requer acesso a disco
 - Problemas
 - Para acesso, a FAT deve estar presente em memória a todo tempo
 - Dependendo do disco
 - A tabela pode ser muito grande para ser mantida em memória
 - Disco de 200GB, blocos de 1K, exige tabela com 200Mega entradas
 - Requer ao 28 bits para endereçar todas entradas (ocupará 4 bytes)
 - MS-DOS trata o problema usando blocos grandes (até 32K)

Sistema de Arquivos

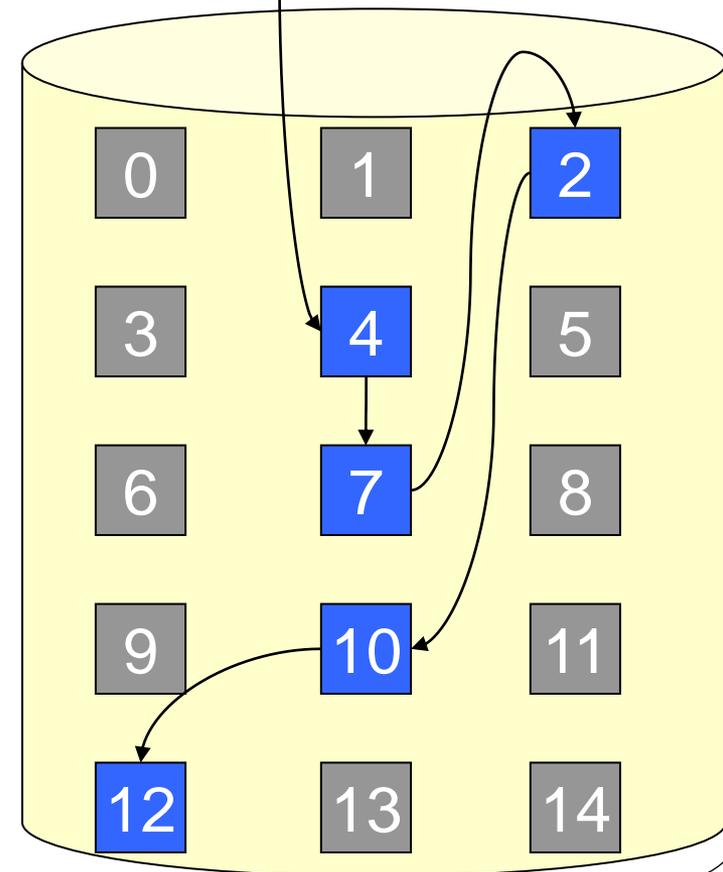
- Armazenamento de arquivos
 - Lista encadeada com tabela na memória



Diretório x

A	4
B	6

File A



Sistema de Arquivos

- Armazenamento de arquivos
 - i-nodes (nós índice)
 - Associada a cada arquivo no disco
 - Um arquivo possui uma estrutura de inode que relaciona
 - Atributos e endereços de blocos de arquivo em disco
 - A partir do i-node
 - Alcança-se todos blocos em disco que formam um arquivo
 - Vantagem sobre a FAT
 - FAT cresce linearmente com o tamanho do disco
 - Inodes requer espaço proporcional a quantidade de arquivos que podem estar abertos
 - Supondo que no máximo x arquivos possam estar abertos por vez
 - Supondo que a estrutura de um inode é igual a y bytes, logo
 - O espaço máximo será $x * y$ bytes
 - Estrutura somente fica em memória quando o arquivo é acessado

Sistema de Arquivos

- Armazenamento de arquivos

- i-nodes

- Composição de um i-node

- Atributos do arquivo

- Nome

- Permissões

- Tamanho

- Momento de

- Criação/acesso/alteração

- Proprietário ...

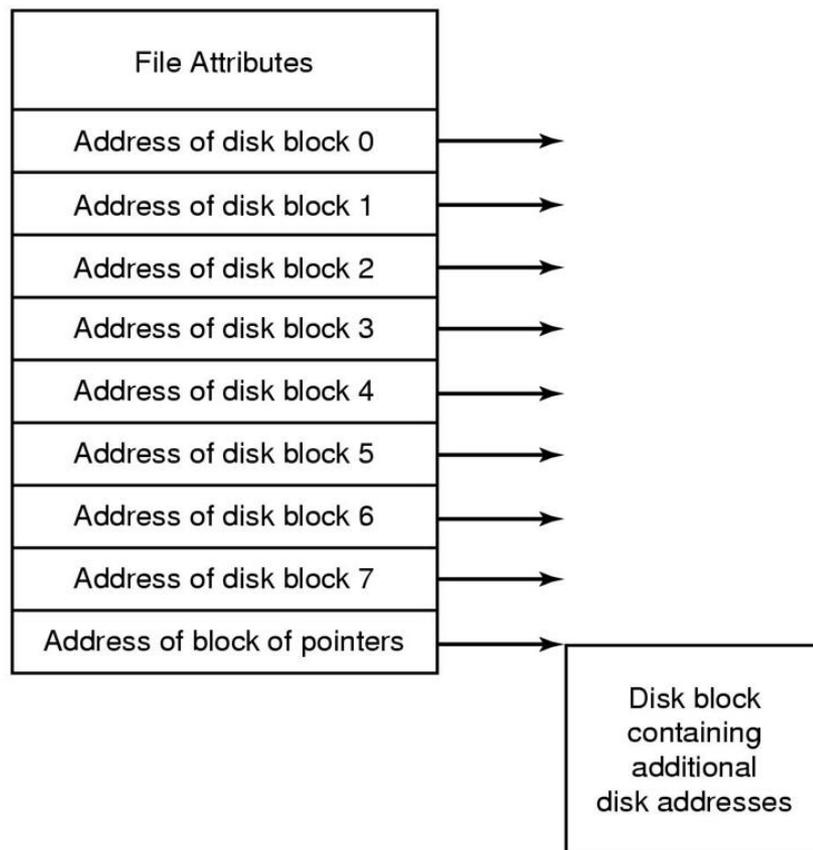
- Endereçamento dos dados

- Endereçamento direto

- Define a quantidade de dados diretamente alcançáveis pela estrutura

- Endereçamento indireto direto

- Utiliza de blocos de dados como blocos de endereçamento de dados



Sistema de Arquivos

- Exemplo
 - se
 - Bloco possui 1K e
 - Bloco é endereçado com 4 bytes
 - e
 - Uma estrutura de inode possui
 - Oito entradas de endereçamento direto
 - Uma de endereçamento indireto simples
 - Uma de endereçamento indireto duplo
 - Uma de endereçamento indireto triplo
 - Então
 - ...

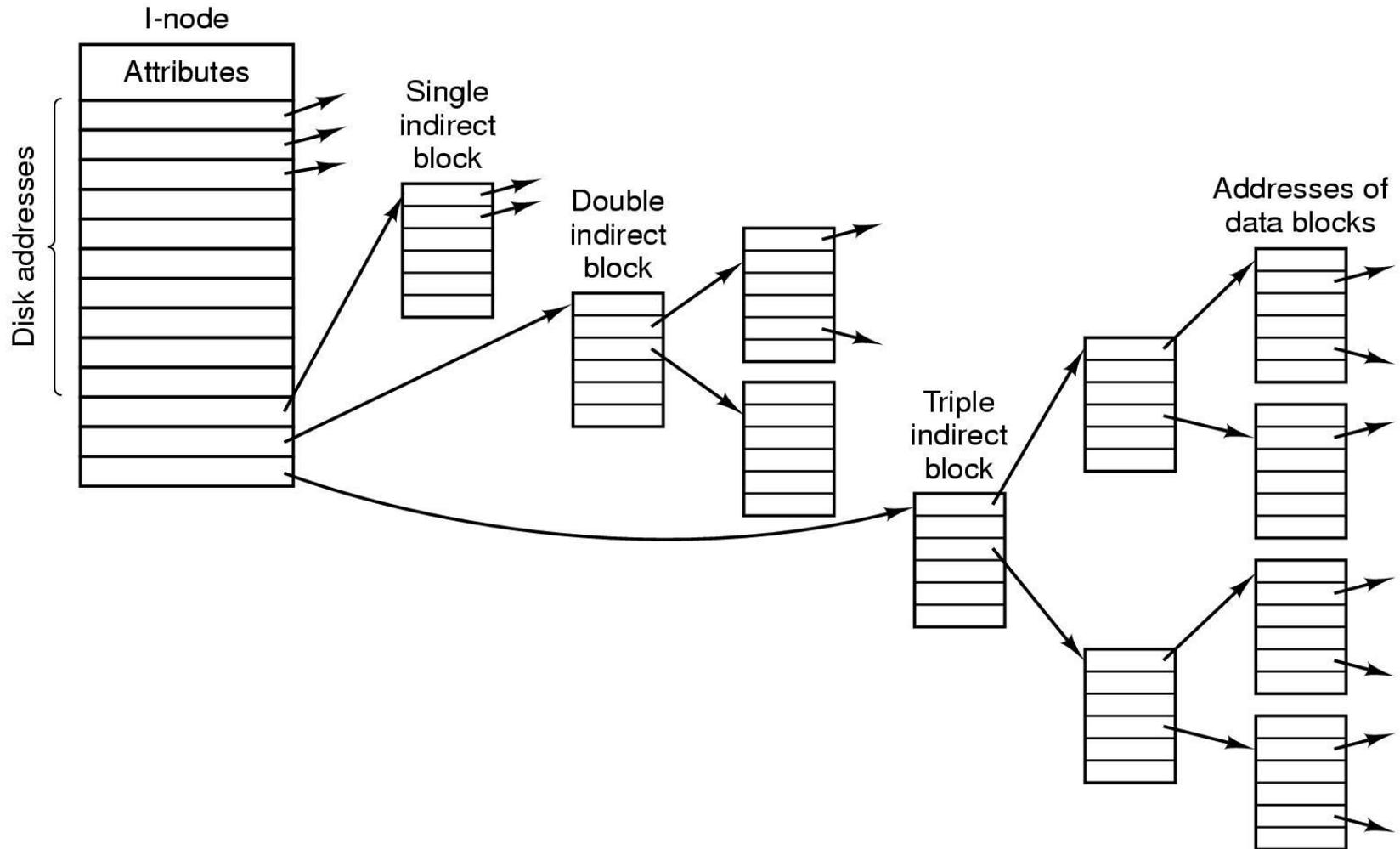
Sistema de Arquivos

● Exemplo

- Blocos de endereçamento direto permitem endereçar
 - 8 blocos de dados
- Blocos de endereçamento indireção simples
 - Endereça 1 bloco de dados que é transformado em bloco de endereço
 - Endereça até 1Kbyte / 4 bytes → 256 blocos de dados
- Blocos de endereçamento indireção duplo
 - Endereça 1 bloco de dados que é transformado em bloco de endereçamento simples, assim
 - Endereça até $256^2 = 65.536$ blocos
- Blocos de endereçamento indireção triplo
 - Endereça 1 bloco de dados que é transformado em bloco de endereçamento duplo, assim
 - Endereça até $256^3 = 16.777.216$ blocos
 - $256 * 256 * 256$ blocos = 16.777.216 blocos
- Somados, pode endereçar um arquivo de até 16.843.012 blocos ~ 17Gb

Sistema de Arquivos

- Exemplo i-nodes - UNIX

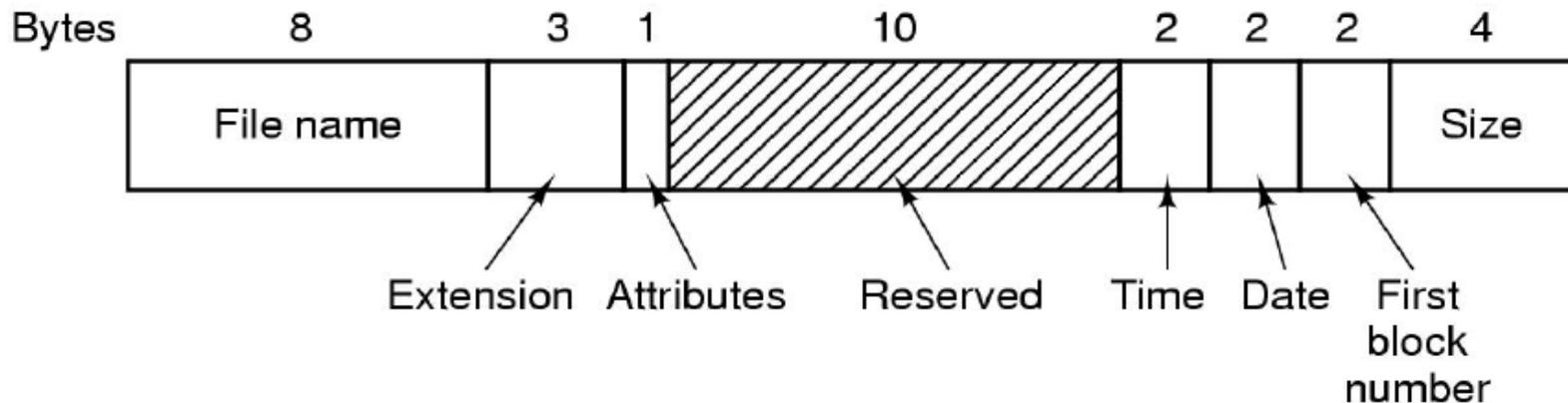


Sistema de arquivos

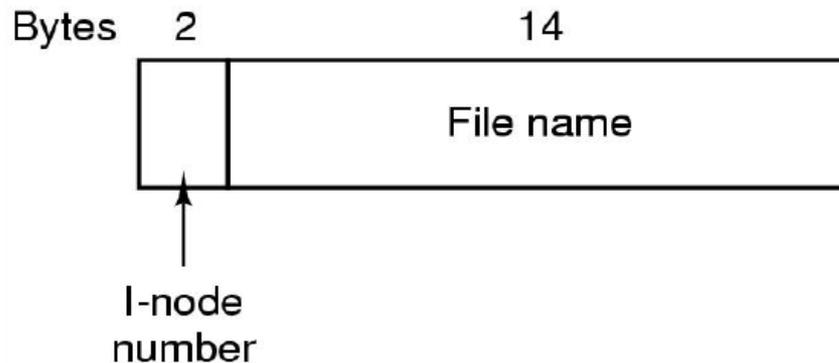
- Implementação de diretórios
 - Contém informações que permitem acessar os arquivos
 - As entradas fornecem informações para encontrar os blocos
 - Possui várias entradas, uma por arquivo:
 - nome
 - tipo; tamanho
 - proprietário; proteção
 - data de criação; data da última modificação
 - lista de blocos usados

Sistema de arquivos

- Entrada de diretório no DOS

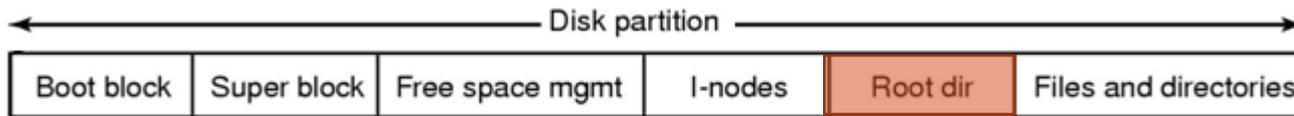


- No Unix (eg. System V)



Diretórios vs I-nodes

- Alcançando /usr/ast/mbox



Root directory

1	.
1	..
4	bin
7	dev
14	lib
9	etc
6	usr
8	tmp

Looking up
usr yields
i-node 6

I-node 6
is for /usr

Mode	size
	times
132	

I-node 6
says that
/usr is in
block 132

Block 132
is /usr
directory

6	.
1	..
19	dick
30	erik
51	jim
26	ast
45	bal

/usr/ast
is i-node
26

I-node 26
is for
/usr/ast

Mode	size
	times
406	

I-node 26
says that
/usr/ast is in
block 406

Block 406
is /usr/ast
directory

26	.
6	..
64	grants
92	books
60	mbox
81	minix
17	src

/usr/ast/mbox
is i-node
60