

FSO - 02/11/2018

Sumário:

Sistemas de ficheiros: objectivos, organização.
Nomes simbólicos, locais a um processo,
internos. Conversão de nomes simbólicos em
nomes locais. Diretorias

Bibliografia:

OSTEP Caps 39 e 40

Conceito de ficheiro

- Um espaço de endereçamento logicamente contíguo
 - Dados
 - números
 - texto
 - binário
 - Programas
- Informação guardada de forma permanente
- Uma parte do SO encarrega-se de gerir os ficheiros e os recursos em que estas residem – **suporte de sistemas de ficheiros** ou simplesmente **sistema de ficheiros**
- A gestão directa dos periféricos é da responsabilidade do componente **gestão de periféricos** do SO

Operações sobre um ficheiro

- Criar - creat
- Escrever - write
- Ler - read
- Reposicionar dentro do ficheiro – lseek
- Apagar - unlink
- Truncar
- Abrir - $\text{Open}(F_i)$ – Procurar no disco a entrada F_i , e mover o conteúdo da entrada para memória.
- Fechar - $\text{Close}(F_i)$ – mover o conteúdo da entrada F_i de memória para a estrutura de directorias em disco.

Programa que usa chamadas ao sistema relacionadas com ficheiros (1/2)

```
/* File copy program. Error checking and reporting is minimal. */

#include <sys/types.h>                /* include necessary header files */
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[]);    /* ANSI prototype */

#define BUF_SIZE 4096                /* use a buffer size of 4096 bytes */
#define OUTPUT_MODE 0700             /* protection bits for output file */

int main(int argc, char *argv[])
{
    int in_fd, out_fd, rd_count, wt_count;
    char buffer[BUF_SIZE];

    if (argc != 3) exit(1);          /* syntax error if argc is not 3 */
```

Programa que usa chamadas ao sistema relacionadas com ficheiros (2/2)

```
/* Open the input file and create the output file */
in_fd = open(argv[1], O_RDONLY); /* open the source file */
if (in_fd < 0) exit(2);          /* if it cannot be opened, exit */
out_fd = creat(argv[2], OUTPUT_MODE); /* create the destination file */
if (out_fd < 0) exit(3);        /* if it cannot be created, exit */

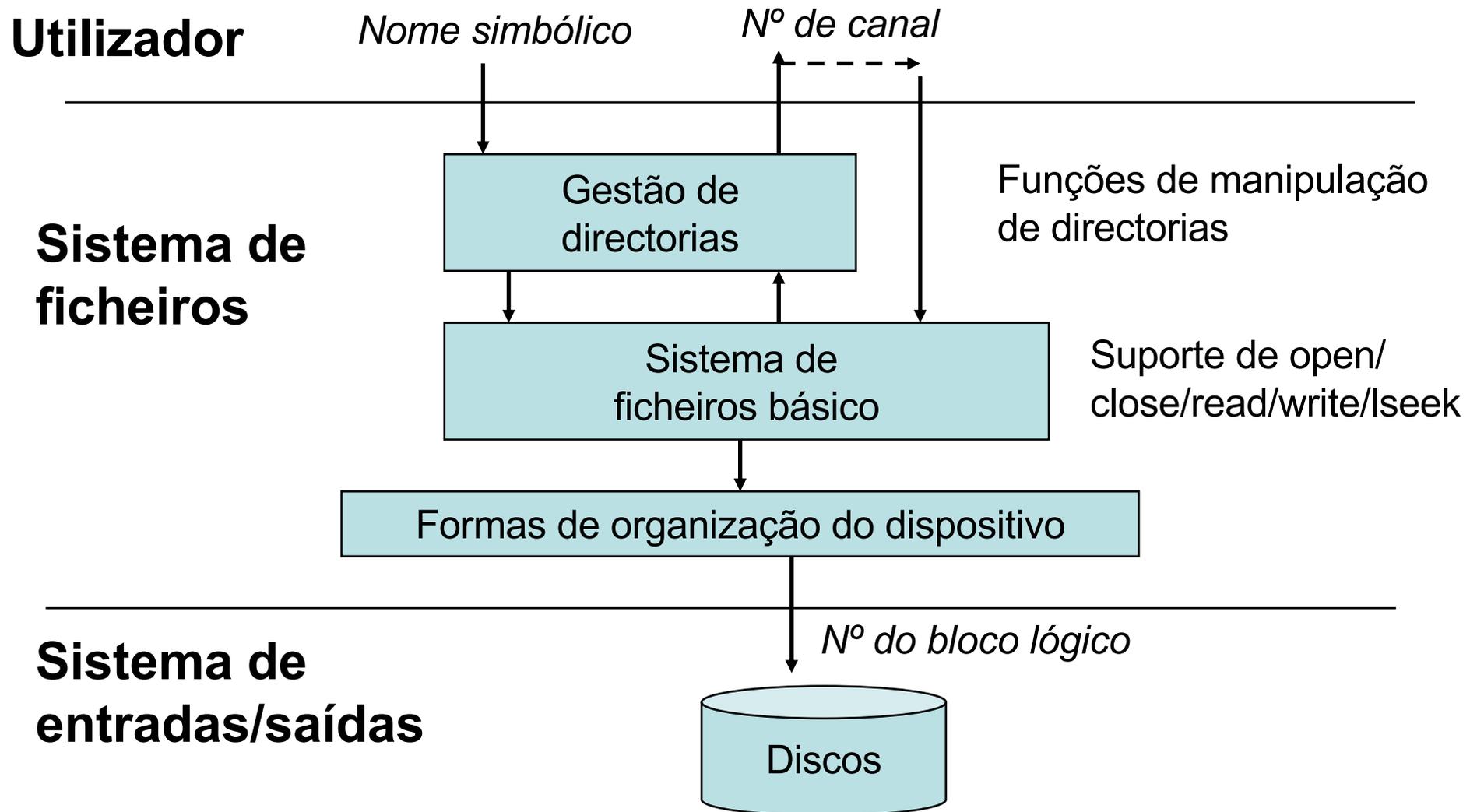
/* Copy loop */
while (TRUE) {
    rd_count = read(in_fd, buffer, BUF_SIZE); /* read a block of data */
    if (rd_count <= 0) break;                /* if end of file or error, exit loop */
    wt_count = write(out_fd, buffer, rd_count); /* write data */
    if (wt_count <= 0) exit(4);              /* wt_count <= 0 is an error */
}

/* Close the files */
close(in_fd);
close(out_fd);
if (rd_count == 0) /* no error on last read */
    exit(0);
else
    exit(5); /* error on last read */
}
```

Funções do gestor de ficheiros

- Apresentar uma ***visão lógica ou abstracta*** dos ficheiros e directorias escondendo aos utilizadores os detalhes dos dispositivos de armazenamento e das operações necessárias ao uso dos dispositivos
- Permitir um ***utilização eficiente*** dos dispositivos de armazenamento
- Suportar a ***partilha de ficheiros*** entre utilizadores. Isto inclui mecanismos de protecção.

Visão por níveis do sistema de ficheiros



Camadas do S.F.

Sistema de ficheiros
lógico

-conversão de nomes lógicos em nomes internos (directorias, ...), protecção e segurança

Sistema de ficheiros
básico

-conhecimento da organização do ficheiro: registos, "stream" de bytes ..., conversão deslocamento – bloco; gestão de blocos livres

- Transforma endereços na forma (nome interno, deslocamento) em endereços do tipo (disco lógico, nº de bloco lógico)

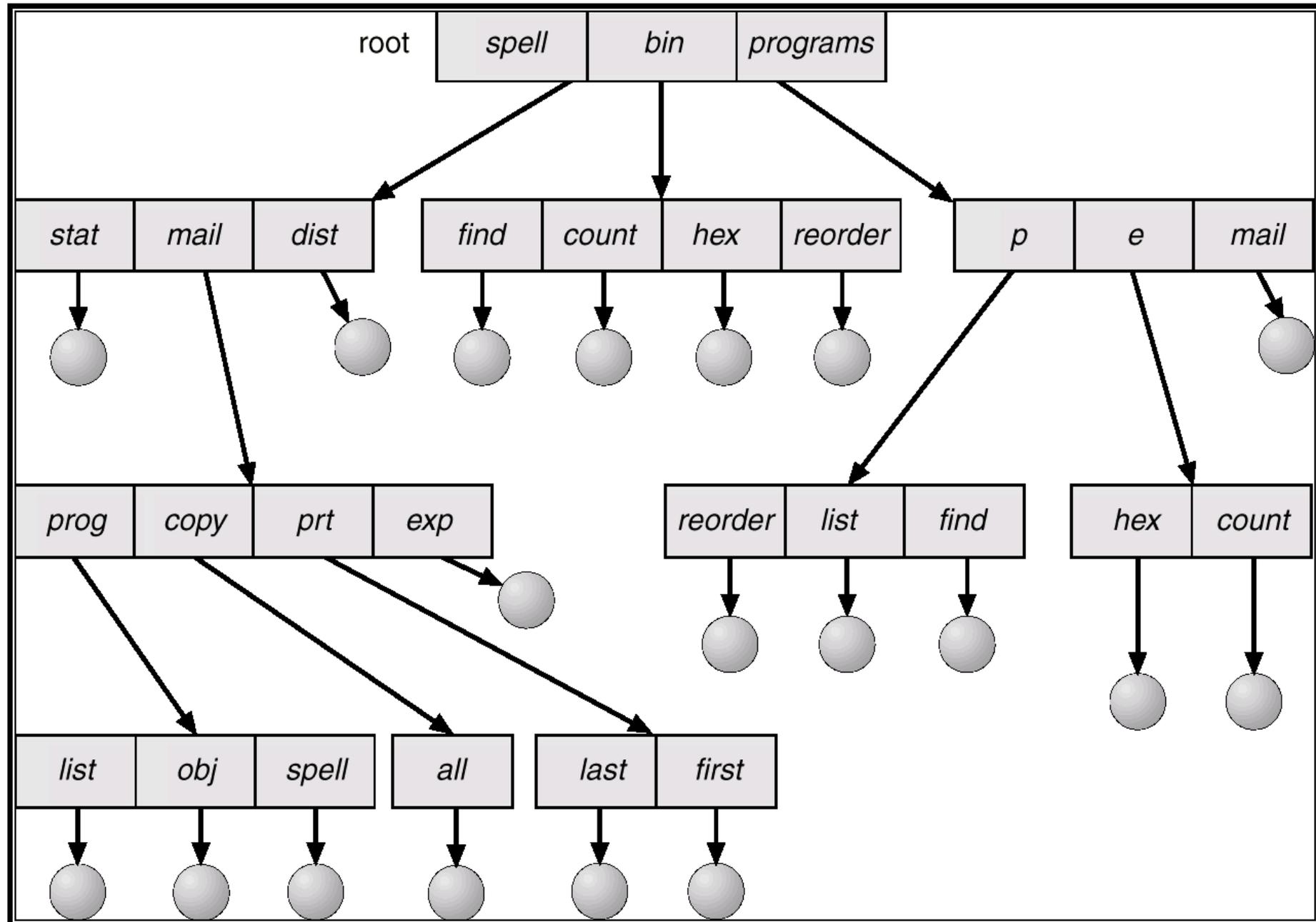
Gestão de periféricos

-Converte endereços recebidos para a forma recebe (disco, superfície, pista, sector);
Transfere dados entre o disco e a memória central

Nomes de um ficheiro

- Nome simbólico
 - Absoluto c:\x\y\z, /home/pm/xpto.c
 - Locais i = open("/home/pm/xpto.c", O_RDONLY)
- Nome interno
 - Índice numa tabela geral que contém todos os ficheiros guardados no volume onde está localizado o ficheiro

Directorias em árvore



Exemplo UNIX

- O sistema de ficheiros no UNIX suporta essencialmente duas entidades: ficheiros e directorias.
- As directorias são apenas ficheiros com um formato especial

Descritor de ficheiro (file control block) – UNIX i-node

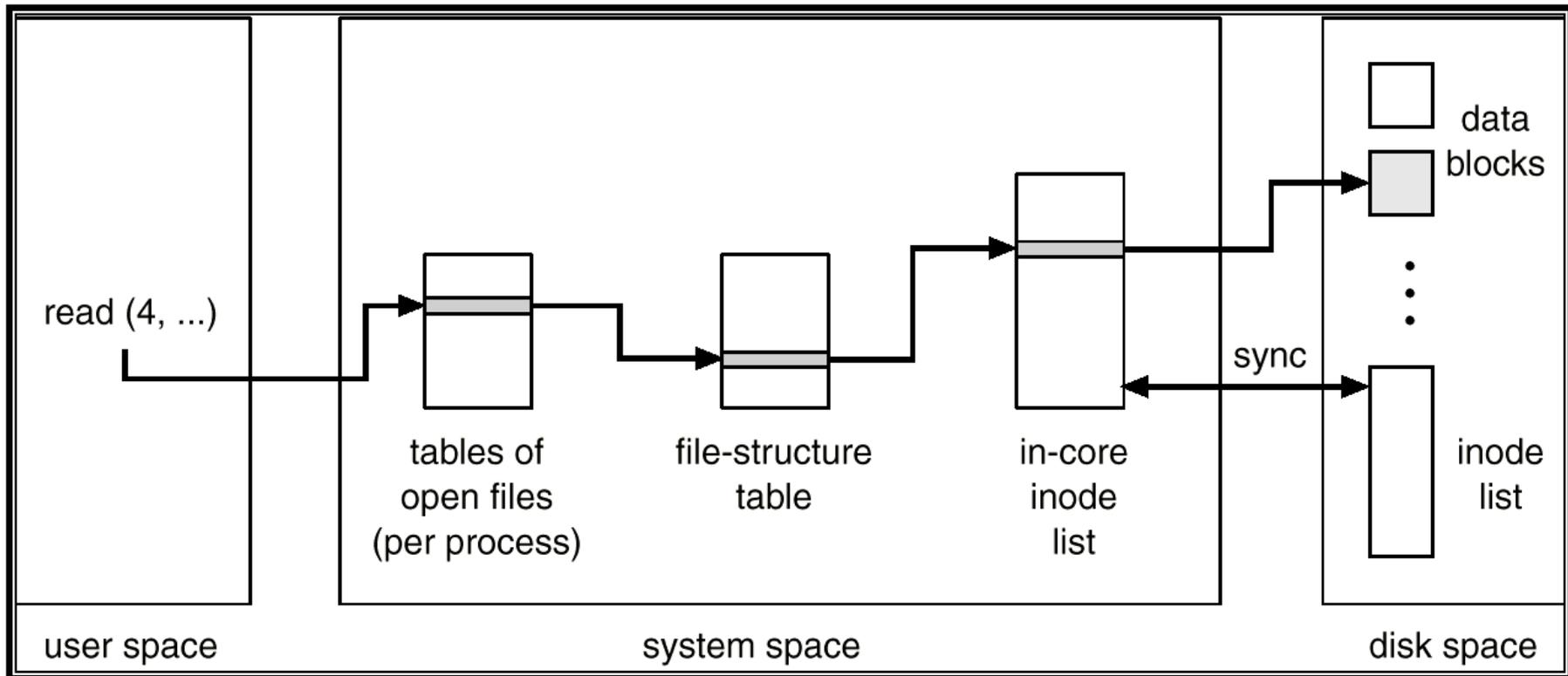
Comp. do ficheiro	Vetor com os blocos do disco que contêm os bytes do ficheiro	Outra informação
-------------------	--	------------------

- O *<inode number>* é usado para identificar um ficheiro.
 - Os *<inode numbers>* num sistema de ficheiros são numerados consecutivamente.

Correspondência entre o nome do canal e o I-node

- As chamadas ao sistema que se referem a ficheiros abertos indicam um índice na *tabela de canais abertos* do processo. Esta entrada referencia uma posição na *tabela geral de ficheiros abertos* do sistema
- Cada entrada destas aponta para um *i-node*.

A chamada read(4, ..., ...)



Directorias

- O campo tipo do i-node permite distinguir entre ficheiros normais (plain files ou ordinary files) e directorias.
- As entradas na directoria são de tamanho variável: cada entrada contém:
 - Comprimento da entrada
 - Nome do ficheiro
 - Número do i-node
- As directorias são usadas para converter nomes lógicos (“/usr/ast/mbox”) em nomes internos (i-nodes – inteiros).

Tabela geral de ficheiros existente num disco



Directorias

Só contem o nome e o índice na tabela geral de ficheiros existentes no volume

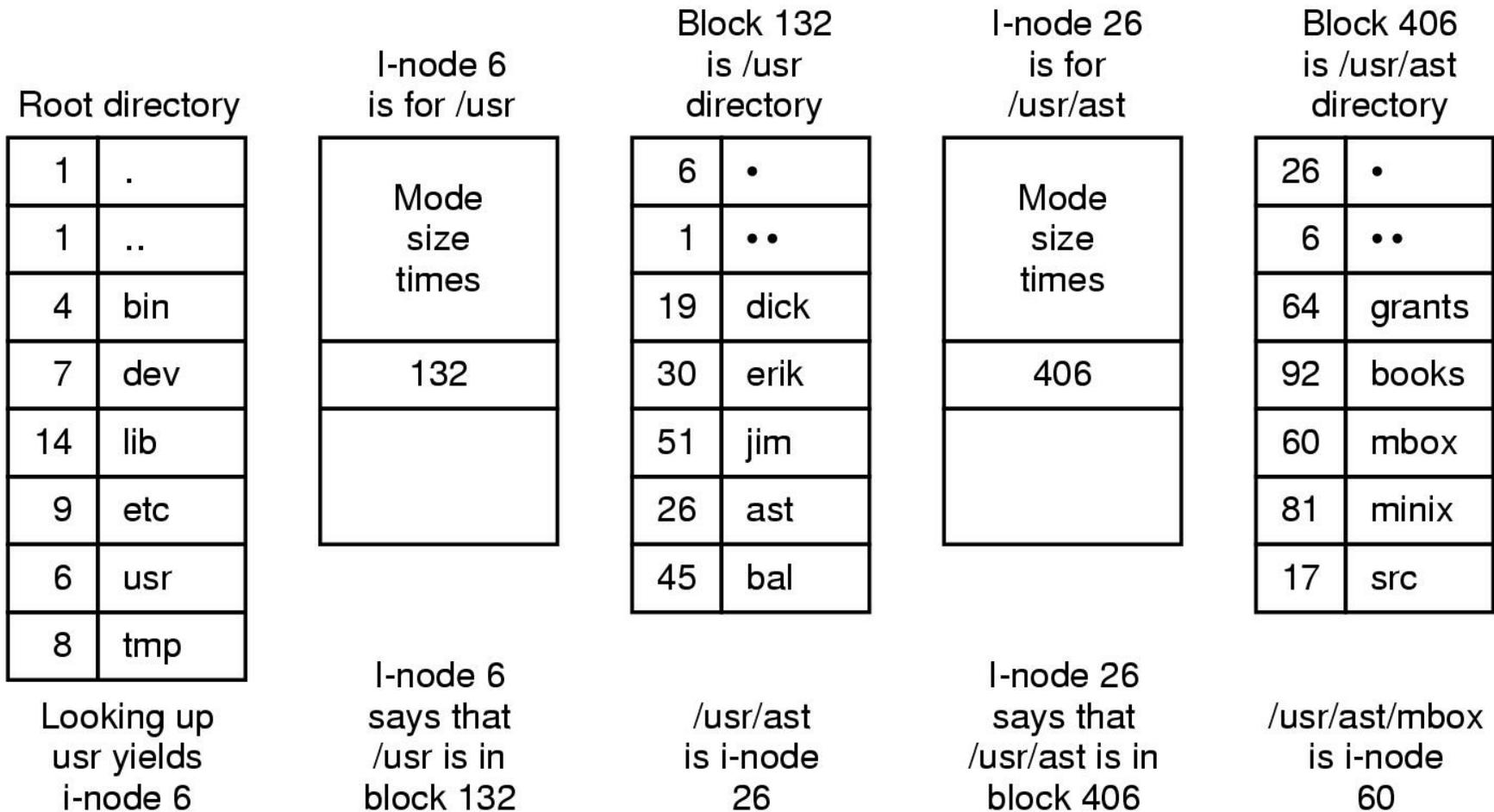
Tabela geral de ficheiros

Cada entrada contém a informação sobre cada ficheiro existente no disco: comprimento, datas de acesso, protecções, ..., blocos onde se situa o conteúdo do ficheiro (no UNIX tabela de i-nodes)

Conversão de nomes

- Primeiro, determina-se a directoria de partida:
 - Se o primeiro caracter é “/”, a directoria de partida é a directoria raiz.
 - Senão a directoria de partida é a directoria corrente.
- A procura continua até o último componente do nome ser alcançado e o i-node pretendido encontrado
- Quando o i-node é encontrado é copiado para memória e é criada uma entrada na tabela de ficheiros aberto com referências para o i-node inode.
- Alguns sistemas gerem uma cache que contém conversões nome- i-node recentes.

Conversão do nome em i-node



Passos realizados para converter o nome
/usr/ast/mbox