

RAID 廉价冗余磁盘阵列.

使用多个磁盘 构建更快. 更大. 更可靠的磁盘系统

RAID 内部: 微控制器.

易失性存储器 (DRAM).

非易失性存储器. 磁盘缓冲.

专用逻辑电路 - 奇偶校验

评估 RAID 容量. 可靠性. 性能

RAID 0 条带化 (无冗余. 容量. 性能好. 但. 可靠性差).

RAID 1 镜像 { RAID 1+0 先做镜像后分块. 允许坏掉磁盘但不能坏一对磁盘
RAID 0+1 先分块后镜像. 允许坏掉个盘但只能在一个 RAID 中

RAID 4 奇偶校验 容量 $N-1$. 只能容忍一个磁盘故障.

RAID 5 旋转奇偶校验.

	RAID 0	RAID 1	RAID 4	RAID 5
容量	N	$\frac{N}{2}$	$N-1$	$N-1$
可靠性	0	1	1	1
吞吐量	顺读 $1S$ 顺写 $1S$ 随读 NR 随写 NR	$\frac{N}{2}S$ $\frac{N}{2}S$ NR $\frac{N}{2}R$	$(N-1)S$ $(N-1)S$ $(N-1)R$ $\frac{1}{2}R$	$(N-1)S$ $(N-1)S$ $N \cdot R$ $\frac{1}{4}R$
延迟	D	D	D	D
带宽	D	D	$>D$	$>D$

文件系统 API

创建文件.

`int fd = open("文件名", 参数);` 如 `O_CREAT` 创建.
`O_WRONLY` 只读.
`O_TRUNC` 强制删除已有内容.
 返回值: 文件描述符

读写文件

`read(文件描述符, 指向缓冲区, 缓冲区大小);`
`write(文件描述符, 数据, 数据长度);`
`off_t fseek(int fd, int whence, off_t offset);`
 文件描述符, 偏移量, 按搜索法偏移.

`fsync()` 立即写入. `off_t fsync(int fd);`

重命名

`rename(char* old, char* new);`

获取文件相关信息, stat(), fstat().

删除文件

unlink unlink()

创建目录

mkdir()

读取目录

opendir()

readdir()

closedir()

删除目录

rmdir(), 目录需为空

硬链接

link(), 不能链接目录, ln

符号链接 (软链接), ln -s

创建并挂载文件系统, mkfs

mount -t

ext3 基于磁盘的文件系统

proc 文件系统系统, 访问当前进程信息

tmpfs 临时文件的文件系统

AFS 分布式文件系统

文件系统实现

文件物理结构: 连续分配

每个文件占据磁盘上一组连续的地

特点: 访问容易, 简单

问题: 为新文件找空间较难, 文件很难增大

链接分配

链表组织

优点: 只需名称即可置, 文件创建增长容易

缺点: 不能随机访问

(FAT: 文件分配表)

索引分配

优点: 既能顺序存取, 又能随机存取, 文件可动态增长, 插入删除, 充分利用存储空间
缺点: 地址列表增大开销

位图: 每个位用指示对应的块是空闲 (0) 还是正在使用 (1)