

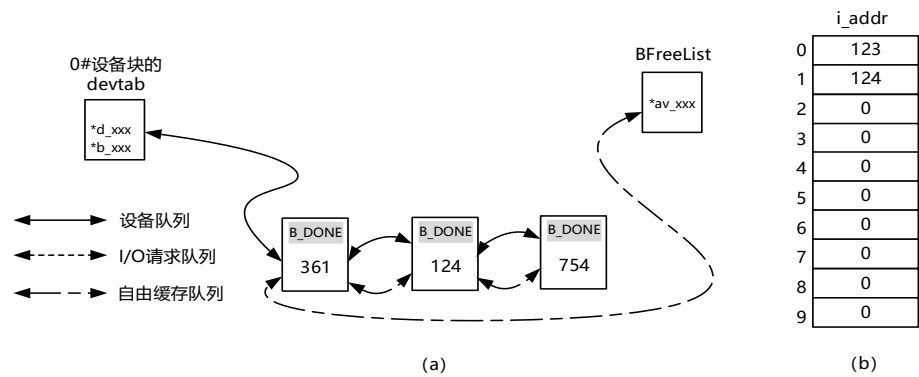
文件系统情景分析题 1

同济大学计算机系统课程例题

2023-12-29

完整的文件读写过程

```
int fd = open("/usr/ast/Jerry", 3); //以可读可写方式打开文件
char data[300];
seek(fd, 500, 0); //将文件读写指针定位到第500字节
int count = read (fd, data, 300); //从文件读300字节到data
count = write(fd, data, 300); //从 data 写 300 字节到文件
```



缓存队列 和 Jerry 文件的索引表

一、open 完成后 i_lastr == -1; seek 将 f_offset 调成 500

二、read 系统调用的执行过程：

```
count = read (fd, data, 300)
```

1、f_offset == 500，文件长度 750。read 系统调用从文件中读 250 字节至 data 数组：data[0]~data[249]。data 数组是局部变量，故后 50 个字符是乱码。

2、细节：要读 0#、1#逻辑块，分 2 次完成。

2.1 初始化 IO 参数：m_offset = f_offset = 500，m_base = data，m_count = 300

2.2 读 0#逻辑块：

当前逻辑块 $bn = m_offset / 512 = 0$ ，块内偏移量 500，读 12 字节至块结束。

查混合索引表，得当前块 bn 的物理块号 123，下一块的物理块号 124。

$bn = i_lastr + 1$ ，顺序读，执行预读操作 `breada (123, 124)`，当前块 123 缓存不命中，预读块 124 缓存命中，PA 为当前块构造 IO 请求块，睡眠等待 IO 完成。

123#块 IO 完成后，中断处理程序唤醒 PA 进程。PA 进程执行 `IOmove` 函数将缓存块中 500# ~ 511#字节复制到用户空间 `data[0]~data[11]`。缓存使用完毕，解锁。

修改 IO 参数， $m_offset = 512$ ， $m_base = data + 12$ ， $m_count = 300 - 12$ 非 0 并且文件没结束，继续读下一块。

2.3 读 1#逻辑块：

当前逻辑块 $bn = m_offset / 512 = 1$ ，块内偏移量 0，读 238 个字节。

查混合索引表，得当前块 bn 的物理块号 124。文件结束，没有下一块。

$bn = i_lastr + 1$ ，顺序读，`breada (124, 0)`。124#物理块缓存命中，锁住，复用这块缓存中的数据。`iomove` 将该缓存中 0#~237#字节复制到用户空间 `data[12] ~ data[249]`，解锁缓存。预读块号为 0，表示无需执行预读操作。

修改 IO 参数， $m_offset = 512 + 238 ==$ 文件长度 i_size 。读操作结束，修改读写指针 $f_offset = 750$ 。返回实际读入的字节数 250。

三、write 系统调用的执行过程：

```
count = write(fd, data, 300)
```

1、 $f_offset == 750$ ，文件长度 750。`write` 系统调用将 `data` 数组中的 300 个字节追加写在文件尾部。写操作完成后，文件长度增加至 1050。

2、细节：分 2 次写入，要为 2#逻辑块分配新数据块。

2.1 初始化 IO 参数： $m_offset = f_offset = 750$ ， $m_base = data$ ， $m_count = 300$

2.2 写 1#逻辑块：

当前逻辑块 $bn = m_offset / 512 = 1$ ，块内偏移量 238，写 274 字节至块结束。

查混合索引表，得当前块 bn 的物理块号 124。

先读后写，这是因为写入的字节数 274 不足 512 字节。124#物理块缓存命中，先读操作只需锁住该缓存块。

`iomove` 将用户空间中的数据 `data[0] ~ data[273]` 写入这块缓存。

修改 IO 参数， $m_offset = 1024$ ， $m_base = data + 274$ ， $m_count = 300 - 274$ 。

写至 1#逻辑块底部，异步写回磁盘。

m_count 非 0，还要继续写。写操作导致文件长度增加， $i_size = 1024$ 。

2.3 写 2#逻辑块:

当前逻辑块 $bn = m_offset/512 = 2$, 块内偏移量 0, 此次写 26 个字节。

混合索引表中 2#逻辑块的物理块号是 0, 系统为其分配新数据块 new, 登记:
 $i_addr[2] = new$ 。

先读后写, 这是因为写入的字节数 26 不足 512 字节。系统为新物理块 new 分配缓存块, 启动 IO 操作、同步读入磁盘数据块 new。

iomove 将用户空间中的数据 $data[274] \sim data[299]$ 写入这块缓存。

修改 IO 参数, $m_offset = 1050$, $m_count = 0$ 。。。

未写至 1#逻辑块底部, 延迟写。缓存块打脏标记, 释放。

写操作导致文件长度增加, $i_size = 1050$ 。 m_count 是 0, write 系统调用结束。返回实际写入文件的字节数 300。