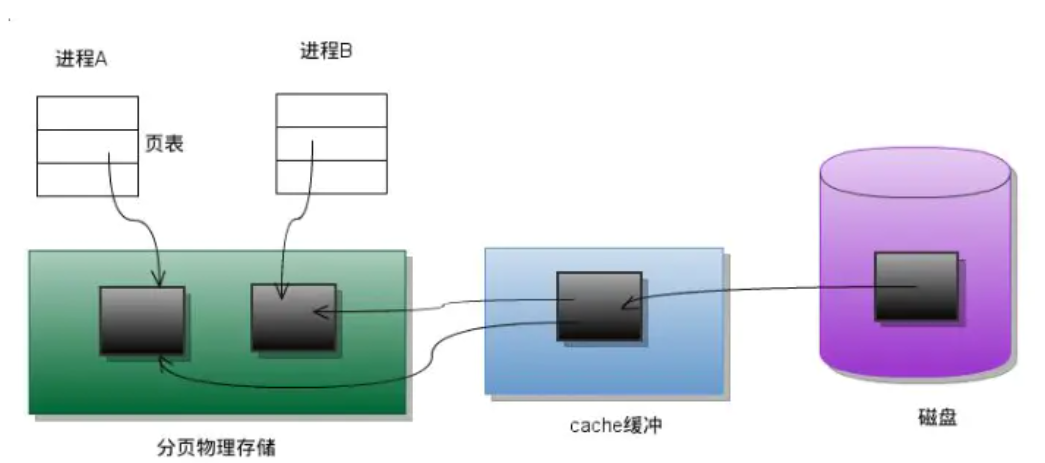
内存映射

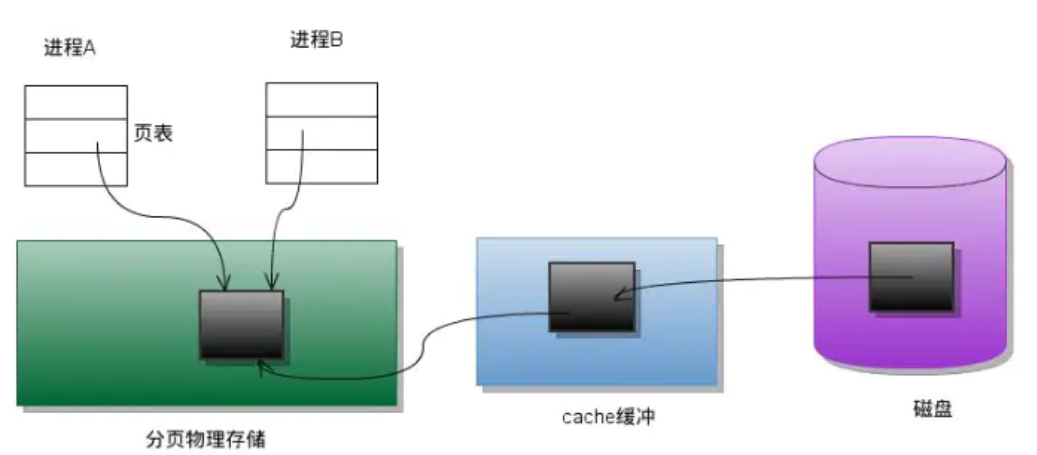
概念：

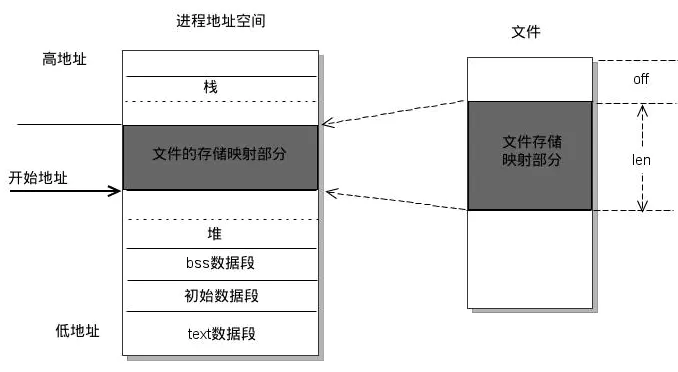
使一个磁盘文件与内存中的一个缓冲区相映射，进程可以像访问普通内存一样对文件进行访问，不必再调用read,write。

mmap()的****优点：****

****实现了用户空间和内核空间的高效交互方式****







函数定义：

void \*mmap(void \*addr, size\_t length, int prot, int flags, int fd, off\_t offset);

功能：创建共享内存映射

函数返回值：成功返回创建的映射区首地址，失败返回MAP\_FAILED（ ((void \*) -1) ），设置errno值

参数说明：

addr：指定要映射的内存地址，一般设置为 NULL 让操作系统自动选择合适的内存地址。

length：必须>0。映射地址空间的字节数，它从被映射文件开头 offset 个字节开始算起。

prot：指定共享内存的访问权限。可取如下几个值的可选：PROT\_READ（可读）, PROT\_WRITE（可写）, PROT\_EXEC（可执行）, PROT\_NONE（不可访问）。

flags：由以下几个常值指定：MAP\_SHARED（共享的） MAP\_PRIVATE（私有的）, MAP\_FIXED（表示必须使用 start 参数作为开始地址，如果失败不进行修正），其中，MAP\_SHARED , MAP\_PRIVATE必选其一，而 MAP\_FIXED 则不推荐使用。MAP\_ANONYMOUS（匿名映射，用于血缘关系进程间通信）

fd：表示要映射的文件句柄。如果匿名映射写-1。

offset：表示映射文件的偏移量，一般设置为 0 表示从文件头部开始映射。

注意事项：

（1） 创建映射区的过程中，隐含着一次对映射文件的读操作，将文件内容读取到映射区。

（2） 当MAP\_SHARED时，要求：映射区的权限应 <=文件打开的权限(出于对映射区的保护)，如果不满足报非法参数（Invalid argument）错误。

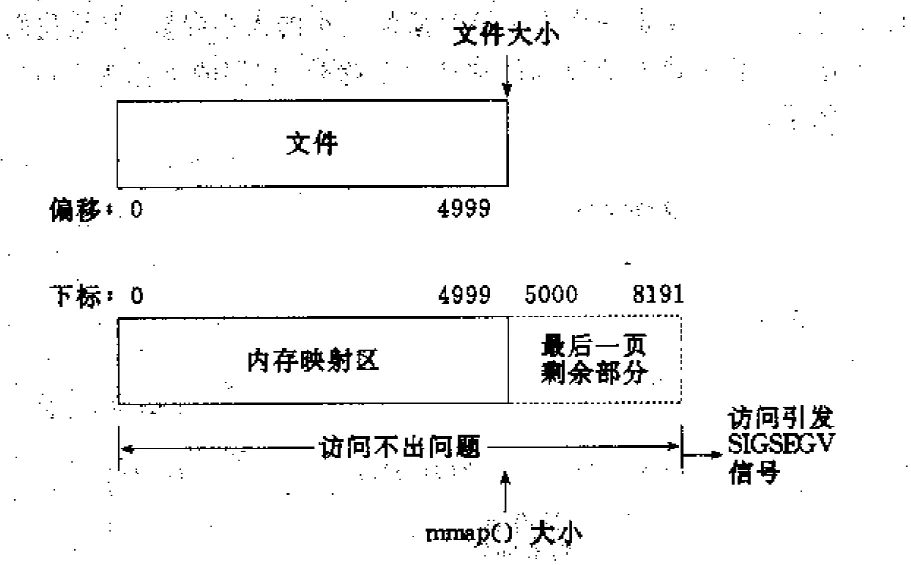
当MAP\_PRIVATE时候，mmap中的权限是对内存的限制，只需要文件有读权限即可，操作只在内存有效，不会写到物理磁盘，且不能在进程间共享。

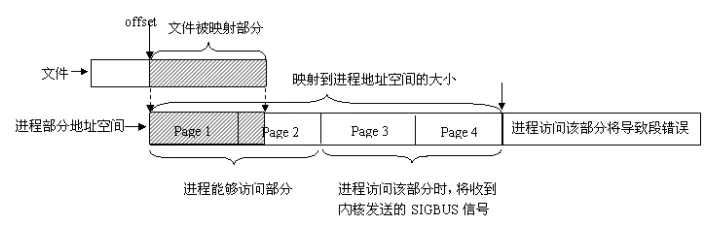
（3） 映射区的释放与文件关闭无关，只要映射建立成功，文件可以立即关闭。

（4） 用于映射的文件大小必须>0，当映射文件大小为0时，指定非0大小创建映射区，访问映射地址会报总线错误，指定0大小创建映射区，报非法参数错误（Invalid argument）

（5） 文件偏移量必须为0或者4K的整数倍（不是会报非法参数Invalid argument错误）.

（6）映射大小可以大于文件大小，但只能访问文件page的内存地址，否则报总线错误 ，超出映射的内存大小报段错误





（7）mmap创建映射区出错概率非常高，一定要检查返回值，确保映射区建立成功再进行后续操作。

mmap()映射的种类：

1基于文件的映射

2 匿名映射

适用于具有亲缘关系的进程之间，

释放内存映射

munmap函数

int munmap(void \*addr, size\_t length);

返回值：成功返回0，失败返回-1，并设置errno值。

函数参数：

addr：调用mmap函数成功返回的映射区首地址

length：映射区大小（即：mmap函数的第二个参数）

**System V共享内存**

**IPC 的key**



**ftok函数**

key\_t ftok(const char \*path, int id);

其中参数path是指定的文件名，这个文件必须是存在的而且可以访问的。id是子序号，它是一个8bit的整数。即范围是0~255。当函数执行成功，则会返回key\_t键值，否则返回-1。在一般的UNIX中，通常是将文件的索引节点取出，然后在前面加上子序号就得到key\_t的值

system V 共享内存使用步骤：

1创建/打开共享内存

2映射共享内存，即把指定的共享内存映射到进程的地址空间用于访问

3 读写共享内存

4 撤销共享内存映射

5 删除共享内存对象

查看共享内存命令ipcs

共享内存创建 – shmget

int shmget(key\_t key, int size, int shmflg);

共享内存映射

void \*shmat(int shmid, const void \*shmaddr, int shmflg);

第二个参数一般写NULL，表示自动映射

第三参数一般写0 ，表示可读写

共享内存撤销

int shmdt(void \*shmaddr);

撤销后，内存地址不可再访问。

共享内存控制

int shmctl(int shmid, int cmd, struct shmid\_ds \*buf);

shmctl(shmid, IPC\_RMID, NULL);删除共享内存