

2.1.4. Fuse block

为了更好地简单理解读写请求后面要做的事情，可以先看看下面这张图。

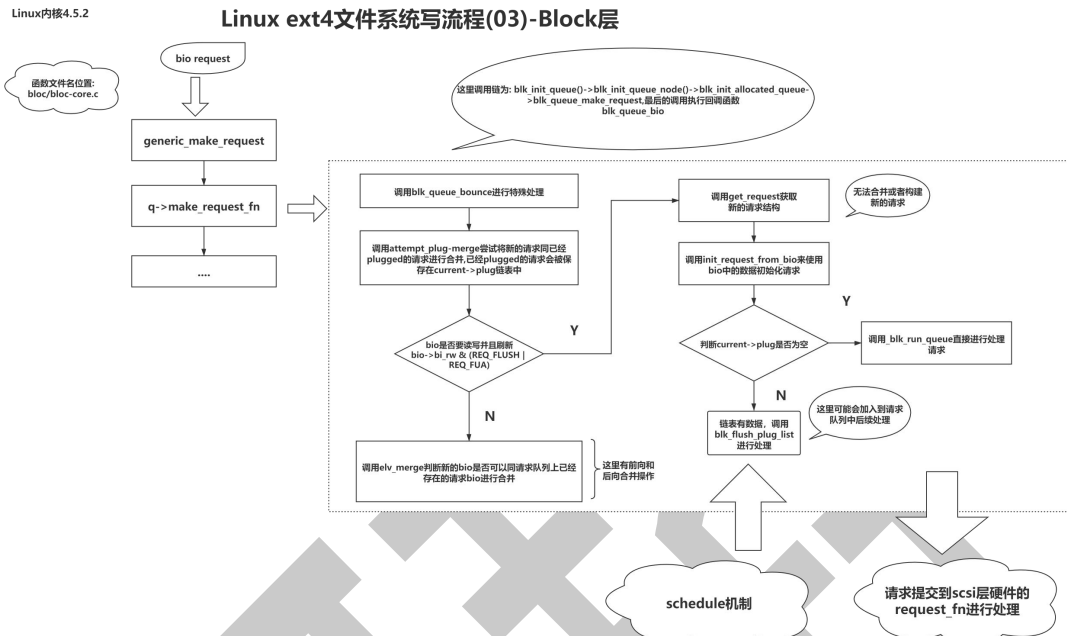


图 2.1.3-1 fuse block 层原理

对于一个 IO 请求，除了前面提到的把数据写入到缓存以外，其实还有一个叫直接 IO 的内容，那么关于 IO 我们常常还有听到同步与异步 IO 的内容，而异步 IO 则只能使用直接 IO 来实现的。另外对于一个 bio 请求来说，这里还会进行前向与后向的合并，之后会放入到调度队列里面，等待调度算法来进行调度处理。

最后，在经历了一系列的原理的理解之后，我们使用 strace 命令来了解一下，在 linux 系统中创建一个文件时使用的函数调用吧。

```
1. # strace touch 1.txt
2. execve("/usr/bin/touch", ["touch", "1.txt"], 0x7ffe4906edc8 /* 24 vars */) = 0
3. brk(NULL) = 0x13ef000
4. mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7ff5b1044000
```

```

5.  access("/etc/ld.so.preload", R_OK)          = -1 ENOENT (No such file or directory)
6.  open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
7.  fstat(3, {st_mode=S_IFREG|0644, st_size=25560, ...}) = 0

8.  mmap(NULL, 25560, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7ff5b103d000
9.  close(3)                                     = 0
10. open("/lib64/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
11. read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\3\0>\0\1\0\0\0`&\2\0\0\0\0"..., 832) = 832
12. fstat(3, {st_mode=S_IFREG|0755, st_size=2156352, ...}) = 0
13. mmap(NULL, 3985920, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7ff5b0a56000
14. mprotect(0x7ff5b0c1a000, 2093056, PROT_NONE) = 0
15. mmap(0x7ff5b0e19000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1c3000) = 0x7ff5b0e19000
16. mmap(0x7ff5b0e1f000, 16896, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7ff5b0e1f000
17. close(3)                                     = 0
18. mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7ff5b103c000
19. mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7ff5b103a000
20. arch_prctl(ARCH_SET_FS, 0x7ff5b103a740) = 0
21. mprotect(0x7ff5b0e19000, 16384, PROT_READ) = 0
22. mprotect(0x60d000, 4096, PROT_READ)        = 0
23. mprotect(0x7ff5b1045000, 4096, PROT_READ) = 0
24. munmap(0x7ff5b103d000, 25560)               = 0
25. brk(NULL)                                   = 0x13ef000
26. brk(0x1410000)                             = 0x1410000
27. brk(NULL)                                   = 0x1410000
28. open("/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
29. fstat(3, {st_mode=S_IFREG|0644, st_size=106176928, ...}) = 0
30. mmap(NULL, 106176928, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7ff5aa513000
31. close(3)                                     = 0
32. open("1.txt", O_WRONLY|O_CREAT|O_NOCTTY|O_NONBLOCK, 0666) = 3
33. dup2(3, 0)                                  = 0

```

34. close(3)	= 0
35. utimensat(0, NULL, NULL, 0)	= 0
36. close(0)	= 0
37. close(1)	= 0
38. close(2)	= 0
39. exit_group(0)	= ?
40. +++ exited with 0 +++	

这里有常见的 mmap 和 brk 是和内存管理分配有关的函数，open 是文件打开的函数等，其中 open 函数中的参数 O_CREAT 是创建并打开一个新文件,关于这些不同的函数的作用与参数意义，可以根据需要时去查阅接口文档。

章节语:

这一章我们主要是理解了一些 linux 文件系统概念，一个正常的读写请求所经过的层次结构，而这里面涉及到内存管理，进程调度等模块，本章也只是非常粗糙简略地讲解了 VFS 和缓存的一些内容，而这个章节的内容，也是为了方便理解后续 glusterfs fuse 中的一些参数做准备的，因为 glusterfs fuse 的底层也是很多我们所熟悉的系统调用，而且 glusterfs fuse 是一个用户文件系统。