

D.

参数	值
CO	0x3
CI	0x5
CT	0xd
高速缓存命中? (是/否)	是
高速缓存字节返回	0x1d

9.5 解决这个题目将帮助你很好地理解内存映射。请自己独立完成这道题。我们没有讨论 open、fstat 或者 write 函数，所以你需要阅读它们的帮助页来看看它们是如何工作的。

code/vm/mmapcopy.c

```
1  #include "csapp.h"
2
3  /*
4   * mmapcopy - uses mmap to copy file fd to stdout
5   */
6  void mmapcopy(int fd, int size)
7  {
8      char *bufp; /* ptr to memory-mapped VM area */
9
10     bufp = Mmap(NULL, size, PROT_READ, MAP_PRIVATE, fd, 0);
11     Write(1, bufp, size);
12     return;
13 }
14
15 /* mmapcopy driver */
16 int main(int argc, char **argv)
17 {
18     struct stat stat;
19     int fd;
20
21     /* Check for required command-line argument */
22     if (argc != 2) {
23         printf("usage: %s <filename>\n", argv[0]);
24         exit(0);
25     }
26
27     /* Copy the input argument to stdout */
28     fd = Open(argv[1], O_RDONLY, 0);
29     fstat(fd, &stat);
30     mmapcopy(fd, stat.st_size);
31     exit(0);
32 }
```

code/vm/mmapcopy.c

9.6 这道题触及了一些核心的概念，例如对齐要求、最小块大小以及头部编码。确定块大小的一般方法是，将所请求的有效载荷和头部大小的和舍入到对齐要求（在此例中是 8 字节）最近的整数倍。比如，malloc(1)请求的块大小是 4+1=5，然后舍入到 8。而 malloc(13)请求的块大小是 13+4=17，舍入到 24。

请求	块大小 (十进制字节)	块头部 (十六进制)
malloc(1)	8	0x9
malloc(5)	16	0x11
malloc(12)	16	0x11
malloc(13)	24	0x19

9.7 最小块大小对内部碎片有显著的影响。因此，理解和不同分配器设计和对齐要求相关联的最小块大小是很好的。很有技巧的一部分是，要意识到相同的块可以在不同时刻被分配或者被释放。因此，最小块大小就是最小已分配块大小和最小空闲块大小两者的最大值。例如，在最后一个子问题中，