第九章 虚拟内存

9.11

A. `00 0010 0111 1100`

B.

参数	值
VPN	0x09
TLB索引	0x01
TLB标记	0x02
TLB命中	No
缺页	No
PPN	0x17

C. `0101 1111 1100`

D.

参数	值
字节偏移	0X0
缓存索引	0XF
缓存标记	0X17
缓存命中	nO
返回字节	-

9.12

A. `00 0011 1010 1001`

B.

参数	值
VPN	0x0e
TLB索引	0x2
TLB标记	0x3
TLB命中	No

参数	值
缺页	No
PPN	0x11

C. `0100 0110 1001`

D.

参数	值
字节偏移	0X01
缓存索引	0Xa
缓存标记	0X11
缓存命中	No
返回字节	-

9.13

A. `00 0000 0100 0000`

B.

参数	值
VPN	0x01
TLB索引	0x1
TLB标记	0x00
TLB命中	No
缺页	No
PPN	0x28

C. `1010 0000 0001`

D.

参数	值
字节偏移	0X1
缓存索引	0X0
缓存标记	0X28
缓存命中	No
返回字节	-

9.14

```c
#include <stdio.h>
#include <sys/stat.h>
#include <sys/mman.h>
// 用于使用 open 函数
#include <sys/fcntl.h>

int main(){
        struct stat stat;
        char *bufp;
        int fd = open("hello.txt", O_RDWR, 0);
        fstat(fd, &stat);
        bufp = mmap(NULL, stat.st_size, PROT_WRITE, MAP_SHARED, fd, 0);
        *bufp = 'J';
        munmap(bufp, stat.st_size);
        return 0;
}
```

9.18

```c
#include "csapp.h"
#include "memlib.h"
#include "mm.h"
#include <assert.h>
#include <stdio.h>

unsigned int blk_head(void *blk) {
        unsigned int *header = (unsigned int *)blk - 1;
        return *header;
}

unsigned int prev_blk_tail(void *blk) {
        unsigned int *prev_tail = (unsigned int *)blk - 2;
        return *prev_tail;
}

int blk_alloc(void *blk) {
        return blk_head(blk) & 0x1;
}

int blk_prev_alloc(void *blk) {
        return (blk_head(blk) & 0x2) >> 1;
}

int rand_num(void) {
        return rand() % 10000;
}

int main(int argc, char *argv[]) {
```

```
        void *blk_1 = mm_malloc(rand_num());
        void *blk_2 = mm_malloc(rand_num());
        void *blk_3 = mm_malloc(rand_num());
        mm_free(blk_2);

        assert(blk_alloc(blk_1));
        assert(!blk_alloc(blk_2));
        assert(blk_alloc(blk_3));

        assert(blk_prev_alloc(blk_1));
        assert(blk_prev_alloc(blk_2));
        assert(!blk_prev_alloc(blk_3));

        assert(blk_head(blk_1) != prev_blk_tail(blk_2));
        assert(blk_head(blk_2) == prev_blk_tail(blk_3));

        return 0;
}
```