选择

1. 下列关于虚存和缓存的说法中，**正确**的是：

A. TLB 是基于物理地址索引的高速缓存

B. 多数系统中，SRAM 高速缓存基于虚拟地址索引

C. 在进行**线程**切换后，TLB条目绝大部分会失效

D. 多数系统中，在进行进程切换后，SRAM 高速缓存中的内容不会失效

1. 阅读下列代码并回答选项。（已知文件“input.txt”中的内容为“12”，头文件没有列出）

void \*Mmap(void \*addr, size\_t length, int prot, int flags,

int fd, off\_t offset);

int main(){

int status;

int fd = Open("./input.txt", O\_RDWR);

char\* bufp = Mmap(NULL, 2, PROT\_READ | PROT\_WRITE ,

MAP\_PRIVATE, fd, 0);

*if* (Fork()>0){

*while*(waitpid(-1,&status,0)>0);

\*(bufp+1) = '1';

Write(1, bufp, 2); // 1: stdout

bufp = Mmap(NULL, 2, PROT\_READ, MAP\_PRIVATE, fd, 0);

Write(1, bufp, 2);

}

*else*{

\*bufp = '2';

Write(1, bufp, 2);

}

close(fd);

return 0;

}

在shell中运行该程序，正常运行时的终端输出应为

1. 221112 B. 222121 C. 222112 D. 221111

大题（15分）

IA32体系采用**小端法**、32位虚拟地址和两级页表。两级页表大小相同，页大小都是4 KB = 212 Byte，结构也相同。TLB 采用**直接映射**，4位组索引。TLB 和页表每一项格式如图所示：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 31 12 | 11 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Address of 4KB page frame | Ignored | G | P  A  T | D | A | P  C  D | P  W  T | U  /  S | R  /  W | P |

部分位的含义如下：

0 (P): 1表示存在，0表示不存在

1 (R/W)：1表示可写，0表示只读

2 (U/S)：1表示内核模式，0表示用户模式

当系统运行到某一时刻时，TLB**有效位为1**的条目如下（未列出部分都是无效的）：

|  |  |  |
| --- | --- | --- |
| 索引（十进制） | TLB 标记 | 内容 |
| 0 | 0x0400 | 0x0ec91313 |
| 3 | 0x02ff | 0x5d2bac01 |
| 5 | 0xd551 | 0x019fa42d |
| 11 | 0x55a6 | 0xfdd3c66b |
| 13 | 0x5515 | 0xb591926b |

一级页表的基地址为0x00e66000，物理内存中的部分内容如下（均为十六进制）：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 地址 | 内容 | 地址 | 内容 | 地址 | 内容 |
| 00615000 | 21 | 00615001 | 2d | 00615002 | ee |
| 00615003 | c0 | 006154d0 | ff | 006154d1 | a0 |
| 00e66001 | a1 | 00e66002 | a4 | 00e66003 | 67 |
| 00e66004 | 21 | 00e66005 | 57 | 00e66006 | 61 |
| 00e66007 | 00 | 2167e000 | 42 | 2167e001 | 67 |
| 2167e002 | 9a | 2167e003 | 7c | c0ee2000 | 6f |
| c0ee2001 | d5 | c0ee2002 | 7e | c0ee24d0 | 48 |
| c0ee24d1 | 83 | c0ee24d2 | ec | c0ee2d21 | 11 |
| c0ee2d22 | 6b | c0ee2d23 | 82 | c0ee2d24 | 8a |

1. 将 cache 清空。访问一个在主存中的虚拟地址，TLB命中，**没有**触发缺页异常，这一过程中，需要访问物理内存（主存）\_\_\_\_\_次（3分）。具体来说，如果该虚拟地址为 y = 0xd5515213，y 地址所具有的**实质权限**是\_\_\_\_\_\_\_\_\_（多选题，选对才得分2分）。

①可写 ②只读 ③用户模式权限 ④内核模式权限

1. 不考虑第一小问，将 cache 清空。访问一个在主存中的虚拟地址，TLB不命中，**没有**触发缺页异常，这一过程中，需要访问物理内存（主存）\_\_\_\_\_次（3分）。具体来说，如果该虚拟地址为x = 0x004004d0，那么x 对应的二级页表起始地址是\_\_\_\_\_\_\_\_\_\_\_\_\_\_（填写16进制，例如0x00123000，2分），x 地址上单字节的**内容**是\_\_\_\_\_\_（填写16进制，例如0x00，1分）。
2. 考虑下面计算矩阵和向量乘法代码：

1 int \*mat\_vec\_mul(int \*\*A, int \*x, int n)

2 {

3 int i, j;

4 int \*y = (int \*)malloc(n \* sizeof(int));

5 *for* (i = 0; i < n; i++)

6 *for* (j = 0; j < n; j++)

7 y[i] += A[i][j] \* x[j];

8 *return* y;

9 }

⑴ 在64位 Linux 机器中运行该代码，输入**同一组合法的参数**后，每次运行返回的向量的值都**不一样**，修复这一错误有一种简单的方法，将第\_\_\_\_行改为\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。（第二空填写C代码，每空1分，共2分）

可能用到的函数：

void \*memcpy(void \*dest, const void \*src, size\_t n);

void \*memset(void \*s, int c, size\_t n);

void \*calloc(size\_t nelem, size\_t elemsize);

void \*realloc(void \*ptr, size\_t size);

⑵ 在进行前一问的测试**之前**，还在64位 Linux 机器上进行过如下用户代码测试（输入 mat\_vec\_mul的参数都**非零**）：

int \*y = mat\_vec\_mul(A, x, n);

int z = y[0];

结果发生了段错误。通过gdb调试发现mat\_vec\_mul函数内并没有发生段错误，但是在初始化变量z时发生了段错误，后来发现是参数的输入有问题。写出出现这种错误的**充分必要条件**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_（1分）

1. Double fault：Intel处理器中有一种特殊的异常，被称为double fault。此异常发生表明调用某个**故障（fault）**A的处理程序后又触发了另一个故障B。正常情况下，故障B会有相应异常处理程序来处理，因此两个故障B和A可以被顺序解决。但是如果处理器无法正常处理故障B，或是处理了之后依然无法处理故障A，就会产生double fault，并终止（abort）。假设除了缺页异常处理程序外，其他异常处理程序**都不会产生新的故障**。如果在某次**缺页故障**时产生了double fault，其原因可能是\_\_\_\_\_\_\_\_\_\_（不定项选择，都选对才得分，1分）

① 运行缺页故障处理程序时， CPU上的权限位是内核态，但所执行代码段 U/S=0

② 运行缺页故障处理程序时，CPU接收到了键盘发送的Ctrl + C信号

③ 缺页故障处理程序没有加载到主存中

备选题

1. 虚拟内存为内存的使用和管理提供了简化，这样的简化**没有**体现在

A. 编译器将C文件编译为目标文件的过程

B. 链接器生成完全链接的可执行文件的过程

C. 加载器向内存中加载可执行文件和共享对象文件的过程

D. 不同进程共享相同物理页面的过程

1. 下列选项中**错误**的是
2. 在使用虚拟地址空间的的系统中，程序引用的页面总数**必须不超过**物理内存总大小。
3. **主存中**的每个**有效**字节都有至少一个选自虚拟地址空间的**虚拟地址**和一个选自物理地址空间的**物理地址**。
4. 当在程序中正常调用malloc函数时，操作系统会分配出相应大小（例如k个）的**连续虚拟页面**，并且将它们映射到物理内存中**任意位置**的k个**物理页面**。
5. **不同**进程的多个**虚拟**页面可以映射到**同一个**共享**物理**页面上。
6. 以下关于动态内存分配的说法中，**错误**的是

A. 可以通过调用 sbrk(0) 获取当前进程中堆的顶部地址

B. 如果向一个已经 free 的指针写入数据，**一定会**触发异常

C. 如果使用 malloc(0x10) 获取一个指针，然后写入 0x200 字节大小的数据，**不一定**会触发异常

D. 使用 mmap 也是动态分配内存的方法之一

1. 以下关于虚拟内存的说法**错误**的是

A. 虚拟内存一般不需要来自应用程序开发者的干涉

B. 虚拟地址空间可以比物理内存更小

C. 连续的虚拟内存**总是**映射到连续的物理内存

D. 目标文件中的.bss段映射到全是二进制零的匿名文件