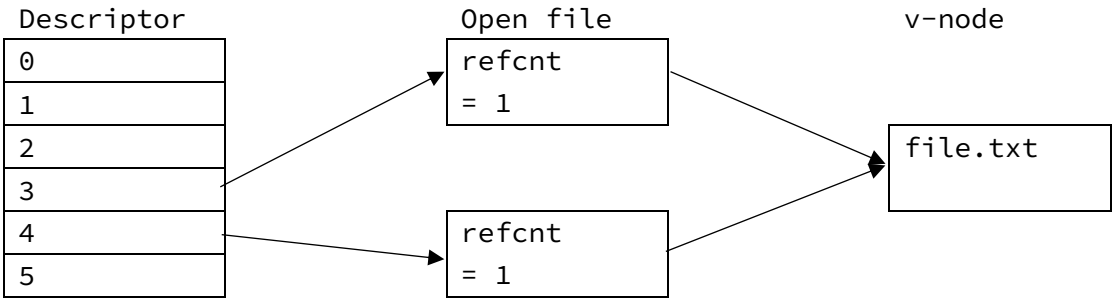
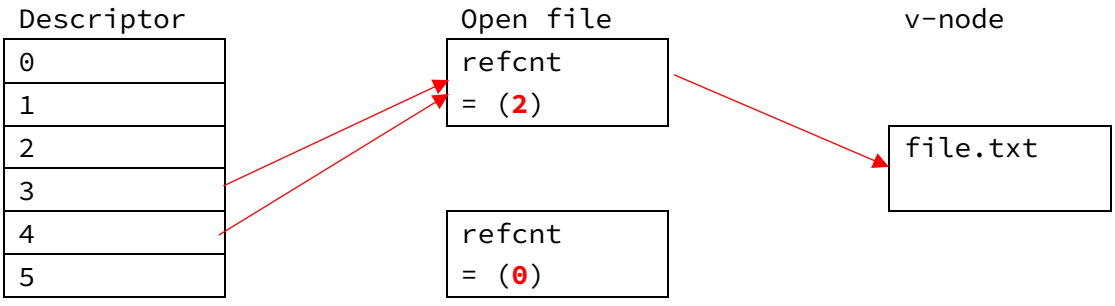
1. 假设磁盘上有空文件 file.txt。程序运行过程中的所有系统调用均成功。

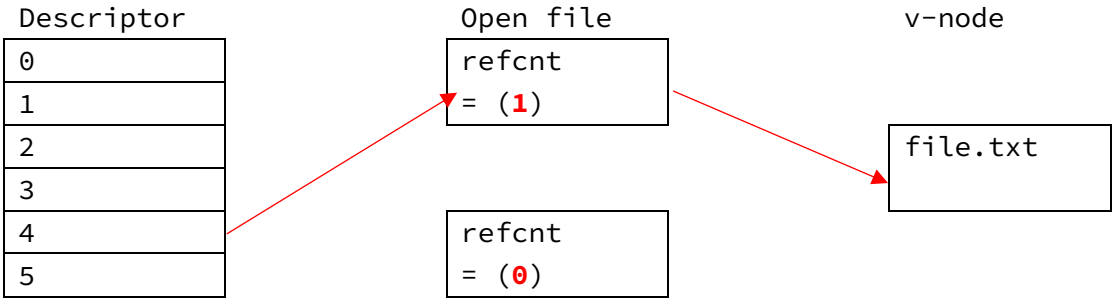
```
int main() {
   int fd1 = open("file.txt", O_RDWR|O_CREAT, S_IRUSR|S_IWUSR);
   int fd2 = open("file.txt", O_RDWR|O_CREAT, S_IRUSR|S_IWUSR);
   printf("%d %d\n", fd1, fd2);
   // A
   write(fd1, "012", 3);
   write(fd2, "ab", 2);
   dup2(fd1, fd2);
   // B
   write(fd1, "123", 3);
   write(fd2, "cd", 2);
   close(fd1);
   // C
   close(fd2);
   return 0;
}
```

已知在程序执行到 A 处时，画出 Linux 三级表的结构如下：



(1)请画出程序在执行到 B,C 处时 Linux 三级表的结构





(2)程序结束时，标准输出上的内容是_**3 4**_，file.txt 中的内容是_**ab2123cd**_.

2. 判断以下说法的正确性

(√)	目录(directory)是一种特殊的文件，包含一组链接(link)，每个链接将一个文件名映射到一个文件。
(×)	关闭一个已经关闭的描述符时，不会出错
(√)	在进程调用 fork()之后，子进程会继承父进程的描述符表(file descriptor table)，也会继承 stdio 的缓冲区
(√)	在编写网络程序时，应该使用 Unix I/O 而不是标准 I/O

3. 假设某进程有且仅有五个已打开的文件描述符：0~4，分别引用了五个不同的文件，尝试运行以下代码：

```
dup2(3,2); dup2(0,3); dup2(1,10); dup2(10,4); dup2(4,0);
```

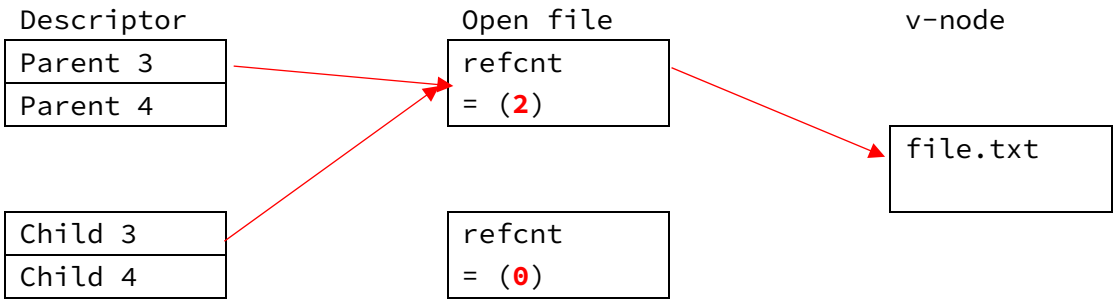
关于得到的结果，说法正确的是：**A**

A. 运行正常完成，现在有四个描述符引用同一个文件
B. 运行正常完成，现在进程共引用四个不同的文件
C. 由于试图从一个未打开的描述符进行复制，发生错误
D. 由于试图向一个未打开的描述符进行复制，发生错误

4. 假设磁盘上有空文件 file.txt。程序运行过程中的所有系统调用均成功。缓冲区足够大，且 stdout 只有在关闭文件、换行与 fflush 的情况下才会刷新缓冲区。

```
int main() {
    pid_t pid; int child_status;
    int fd1 = open("file.txt", O_RDWR|O_CREAT, S_IRUSR|S_IWUSR);
    if ((pid = fork()) > 0) {            // Parent
        printf("P:%d ", fd1);
        write(fd1, "123", 3);
        waitpid(pid, &child_status, 0);
    } else {                             // Child
        printf("C:%d ", fd1);
        write(fd1, "45", 2);
    }
    close(fd1); return 0;
}
```

(1)在子进程关闭 fd1 前，画出 Linux 三级表的结构如下



(2)程序结束时，标准输出上的内容是_____，file.txt 中的内容是_____.

C:3 P:3 ;12345 或 45123 注意:C 一定在 P 前输出

5. 根据本课程介绍的 Intel x86-64 存储系统，填写表格中某一个进程从用户态切换至内核态时，和进程切换时对 TLB 和 cache 是否必须刷新 **A**

A. ①不必刷新 ②不必刷新 ③刷新 ④不必刷新
B. ①不必刷新 ②不必刷新 ③不必刷新 ④不必刷新
C. ①刷新 ②不必刷新 ③刷新 ④刷新
D. ①刷新 ②不必刷新 ③不必刷新 ④刷新

6. 下列关于虚存和缓存的说法中，正确的是： **D**
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

7. 对于虚拟存储系统,一次访存过程中,下列命中组合不可能发生的是 **D**.
A.TLB 未命中，Cache 未命中，Page 未命中
B.TLB 未命中,Cache 命中， Page 命中
C.TLB 命中， Cache 未命中，Page 命中
D.TLB 命中， Cache 命中， Page 未命中

8. 关于写时复制（copy-on-write，COW）技术的说法，不正确的是： **D**
A. 写时复制既可以发生在父子进程之间，也可以发生在对等线程之间
B. 写时复制既需要硬件的异常机制，也需要操作系统软件的配合
C. 写时复制既可以用于普通文件，也可以用于匿名文件
D. 写时复制既可以用于共享区域，也可以用于私有区域

9. 假设有一台 64 位的计算机的物理页块大小是 8KB,采用三级页表进行虚拟地址寻址,它的虚拟地址的 VPO(Virtual Page Offset,虚拟页偏移)有 13 位,问它的虚拟地址的 VPN(Virtual Page Number,虚拟页号码)有多少位?**C**
A. 20
B. 27
C. 30
D. 33
本题考查对页表组成的理解.页块大小为 8KB,即 2^13 byte.在 64 位机器上,一个页表条目为 8byte.故共有页表条目 2^10 项,故每一级页表可以表示 10 位地址.因此三级页表存储，共需要 10*3=30 位.
10. 进程 P1 通过 fork()函数产生一个子进程 P2.假设执行 fork()函数之前，进程 P1 占用了 53 个(用户态的)物理页,则 fork 函数之后,进程 P1 和进程 P2 共占用 _____个(用户态的)物理 页;假设执行 fork()函数之前进程 P1 中有一个可读写的物理页,则执行 fork()函数之后,进程 P1 对该物理页的页表项权限为 **B**.
A. 53,读写
B. 53,只读
C. 106,读写
D. 106,只读

11. Intel 的 IA32 体系结构采用二级页表,称第一级页表为页目录(Page Directory),第二级页表为页表（Page Table）。页面的大小为 4KB，页表项 4 字节。以下给出了页目

录与若干页表中的部分内容，例如，页目录中的第 1 个项索引到的是页表 3，页表 1 中的第 3 个项索引到的是物理地址中的第 5 个页。则十六进制逻辑地址 8052CB 经过地址转换后形成的物理地址应为十进制的（ **B**）。

页目录		页表 1		页表 2		页表 3	
VPN	页表号	VPN	页号	VPN	页号	VPN	页号
1	3	3	5	2	1	2	9
2	1	4	2	4	4	3	8
3	2	5	7	8	6	5	3

A. 21195
B. 29387
C. 21126
D. 47195

4KB=2^12,所以页内地址有 12 位.4KB/4B=1K,所以页目录和每个页表中的页表项数为 1K 个.因此,在 32 位的虚拟地址中,最低的 12 位为页内地址(Offset),最高的 10 位为页目录的虚拟地址(Dir),中间 10 位为页表的虚拟地址(Table).
十六进制逻辑地址 8052CB 转换为二进制后为 100000000101 001011001011,Dir 为 10,即 10 进制的 2,在表中对应到页表 1.
Table 为 101,即 10 进制的 5,在表中对应到物理页面 7.因此,物理地址应为 7 的二进制 111 和 Offset 的拼合,即 111001011001011,转换为十进制为 29387,答案为 B.

12. 假定整型变量 A 的虚拟地址空间为 0x12345cf0,另一整形变量 B 的虚拟地址 0x12345d98,假定一个 page 的长度为 0x1000 byte,A 的物理地址数值和 B 的物理地址数值关系应该为: **B,在同一页中**
A.A 的物理地址数值始终大于 B 的物理地址数值
B.A 的物理地址数值始终小于 B 的物理地址数值
C.A 的物理地址数值和 B 的物理地址数值大小取决于动态内存分配策略
D.无法判定两个物理地址值的大小

13. 下列与虚拟内存有关的说法中哪些是不对的？ **D**
A.操作系统为每个进程提供一个独立的页表,用于将其虚拟地址空间映射到物理地址空间.
B.MMU 使用页表进行地址翻译时,虚拟地址的虚拟页面偏移与物理地址的物理页面偏移是相同的.
C.若某个进程的工作集大小超出了物理内存的大小,则可能出现抖动现象.
D.动态内存分配管理,采用双向链表组织空闲块,使得首次适配的分配与释放均是空闲块数量的线性时间.

14. 在 Core i7 中,关于虚拟地址和物理地址的说法,不正确的是:**B**
A.VPO = CI + CO
B.PPN = TLBT + TLBI
C.VPN1 = VPN2 = VPN3 = VPN4
D.TLBT + TLBI = VPN

15. 已知某系统页面长 8KB,页表项 4 字节,采用多层分页策略映射 64 位虚拟地址空间.若限定最高层页表占 1 页,则它可以采用多少层的分页策略?

A.3 层 B.4 层 C.5 层 D.6 层

C.由题意,64 位虚拟地址的虚拟空间大小为 264.页面长为 8KB,页表项 4 字节,所以一个页面可存放 2K 个表项.由于最高层页表占 1 页,也就是说其页表项个数最多为 2K 个,每一项对应一页,每页又可存放 2K 个页表项,依次类推可知,采用的分页层数为:5 层.

第五题（10 分）

以下程序运行时系统调用全部正确执行，且每个信号都被处理到。请给出代码运行后所有可能的输出结果。

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <signal.h>
int c = 1;
void handler1(int sig) {
    c++;
    printf("%d", c);
}

int main() {
    signal(SIGUSR1, handler1);
    sigset_t s;
    sigemptyset(&s);
    sigaddset(&s, SIGUSR1);
    sigprocmask(SIG_BLOCK, &s, 0);

    int pid = fork()?fork():fork();
    if (pid == 0) {
        kill(getppid(), SIGUSR1);
        printf("S");
        sigprocmask(SIG_UNBLOCK, &s, 0);
        exit(0);
    } else {
        while (waitpid(-1, NULL, 0) != -1);
        sigprocmask(SIG_UNBLOCK, &s, 0);
        printf("P");
    }
    return 0;
}
```

答: **共 3 种： S2PS2P SS2P2P S2SP2P**

得分

第六题（15 分）虚拟内存地址转换

为了提升虚拟内存地址的转换效率，降低遍历两级页表结构所带来的地址转换开销，英特尔处理器中引入了大页 TLB，即一个 TLB 项可以涵盖整个 4MB 对齐的地址空间（针对 32 位模式）。只要设置页目录页中页目录项（PDE）的大页标志位，即可让 MMU 识别这是一个大页 PDE，并加载到大页 TLB 项中。大页 PDE 中记录的物理内存页面号必须是 4MB 对齐的，并且整个连续的 4MB 内存均可统一通过该大页 PDE 进行地址转换。

在 32 位的 Linux 系统中，为了方便访问物理内存，内核将地址 0~768MB 间的物理内存映射到虚拟内存地址 3GB~3GB+768MB 上，并通过大页 PDE 进行进行该区间的地址转换。任何 0~768MB 的物理内存地址可以直接通过加 3G（0xC0000000）的方式得到其虚拟内存地址。在内核中，除了该区间的内存外，其他地址的内存通常都通过普通的两级页表结构来进行地址转换。

假设在我们使用的处理器中有 2 个大页 TLB 项，其当前状态如下：

索引号	TLB 标记	页面号	有效位
0	0xC4812	0x04812	1
1	0xC9C33	0x09C33	1

有 4 个普通 TLB 项，当前的状态如下：

索引号	TLB 标记	页面号	有效位
0	0xF8034	0x04812	1
1	0xF8033	0x09812	1
2	0xF4427	0x12137	1
3	0xF44AE	0x17343	1

当前页活跃的目录页（PD）中的部分 PDE 的内容如下：

PDE 索引	页面号	其他标志	大页位	存在位
786	0x04800	…	1	1
807	0x09C00	…	1	1
977	0x09C33	…	0	1
992	0x09078	…	0	1

注：普通页面大小为 4KB，并且 4KB 对齐。每个页面的页面号为其页面起始物理地址除以 4096 得到。大页由连续 1024 个 4KB 小页组成，且 4MB 对齐。

1．分析下面的指令序列，

```
movl $0xC4812024, %ebx
movl $128, (%ebx)
movl $0xF8034000, %ecx
movl $36(%ecx), %eax
```

请问，执行完上述指令后，eax 寄存器中的内容是（）；在执行上述指令过程中，共发生了（）次 TLB miss？同时会发生（）次 page fault？

注：不能确定时填写“--”。

2．请判断下列页面号对应的页面中，哪些一定是页表页？哪些不是？哪些不确定？

页面号	是否为页表页（是/不是/不确定）
0x04800	
0x09C33	
0x09812	

3．下列虚拟地址中哪一个对应着够将虚拟内存地址 0xF4427048 映射到物理内存地址 0x14321048 的页表项（）？

（A）0x09C33027　　（B）0xC9C3309C

（C）0xC9C33027　　（D）0x09C3309C

通过上述虚拟地址，利用 movl 指令修改对应的页表项，完成上述映射，在此过程中，是否会产生 TLB miss？（）（回答：会/不会/不确定）

修改页表项后，是否可以立即直接使用下面的指令序列将物理内存地址 0x14321048 开始的一个 32 位整数清零？为什么？

```
movl $0xF4427048, %ebx
movl $0, (%ebx)
```

答：

答案：

第 1 小题（各 1 分）

（1）128；（2）0；（3）0；

两个虚拟地址映射的是同一块物理内存；因此读出的就是写入的；此过程中全部

TLB 命中，因而既无 TLB miss，也不会有 page fault。

第 2 小题（各 2 分）
（1）不确定；因为是大页，一定不是当前页目录项对应的页表页，但不一定该页面不会用作其他页目录项对应的页表页。
（2）是；当前页目录项（977）对应的页表页。
（3）不确定；任何页面都可能用作页表页。

第 3 小题
B；（2 分）
虚拟地址对应的页表页的页面号（0x09C33）已知，通过其地址直接加 3G（即 0xC0000000），即可得到当前页表页的基地址（0xC9C33000），在加上对应的第 0x27 乘以 4 到页内偏移。

不会；（2 分）
因为地址 0xC9C33000 在大页映射范围内，已经被大页 TLB 项覆盖到了，会直接命中。

不能直接修改。（1 分）
因为 TLB 项中的内容和页表中的内容不一致，需要将对应的 TLB 项设置为失效，然后通过 TLB miss 重新加载页表结构中新的地址映射关系，之后才能访问对应的虚拟地址。（1 分）

得分

第五题（12 分）

1. （2 分，每空一分，考察多级页表和单级页表的设计思想的理解）

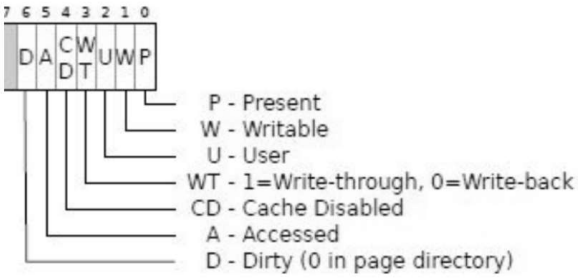
在进行地址翻译的过程中，操作系统需要借助页表(Page Table)的帮助。考虑一个 32 位的系统，页大小是 4KB，页表项(Page Table Entry)大小是 4 字节(Byte)，如果不使用多级页表，常驻内存的页表一共需要_____页。

考虑下图已经显示的物理内存分配情况，在二级页表的情况下，已经显示的区域的页表需要占据_____页。

<table>
<tr><td>VP0</td><td rowspan="6">已分配页</td></tr>
<tr><td>…</td></tr>
<tr><td>VP1023</td></tr>
<tr><td>VP1024</td></tr>
<tr><td>…</td></tr>
<tr><td>VP2047</td></tr>
<tr><td>Gap</td><td rowspan="2">未分配页</td></tr>
<tr><td>1023
unallocated
pages</td></tr>
<tr><td>VP10239</td><td rowspan="4">已分配页</td></tr>
<tr><td>VP10240</td></tr>
<tr><td>…</td></tr>
<tr><td>VP11263</td></tr>
</table>

2. （6 分，每错一空扣一分，扣完为止。考察地址翻译过程）

IA32 体系采用小端法和二级页表。其中两级页表大小相同，页大小均为 4KB，结构也相同。TLB 采用直接映射。TLB 和页表每一项的后 7 位含义如下图所示。为简便起见，假设 TLB 和页表每一项的后 8~12 位都是 0 且不会被改变。注意后 7 位值为“27”则表示可读写。



当系统运行到某一时刻时，TLB 内容如下：

索引	TLB 标记	内容	有效位
0	0x04013	0x3312D027	1
1	0x01000	0x24833020	0
2	0x005AE	0x00055004	1
3	0x00402	0x24AEE020	0
4	0x0AA00	0x0005505C	0
5	0x0000A	0x29DEE000	1
6	0x1AE82	0x00A23027	1
7	0x28DFC	0x00023000	0

一级页表的基地址为 0x0C23B00，物理内存中的部分内容如下：

地址	内容	地址	内容	地址	内容	地址	内容
00023000	E0	00023001	BE	00023002	EF	00023003	BE
00023120	83	00023121	C8	00023122	FD	00023123	12
00023200	23	00023201	FD	00023202	BC	00023203	DE
00023320	33	00023321	29	00023322	E5	00023323	D2
0005545C	97	0005545D	C2	0005545E	7B	0005545F	45
00055464	97	00055465	D2	00055466	7B	00055467	45
0C23B020	27	0C23B021	EB	0C23B022	AE	0C23B023	24
0C23B040	27	0C23B041	40	0C23B042	DE	0C23B043	29
0C23B080	05	0C23B081	5D	0C23B082	05	0C23B083	00
2314D200	23	2314D201	12	2314D202	DC	2314D203	0F
2314D220	A9	2314D221	45	2314D222	13	2314D223	D2

29DE404C	27	29DE404D	42	29DE404E	BA	29DE404F	00
29DE4400	D0	29DE4401	5C	29DE4402	B4	29DE4403	2A

此刻，系统先后试图对两个已经缓存在 cache 中的内存地址进行写操作，请分析**完成写之后**系统的状态（写的地址和上面的内存地址无交集），完成下面的填空。若不需要某次访问或者缺少所需信息，请填“\”。

第一次向地址 0xD7416560 写入内容，TLB 索引为：_____，完成写之后该项 TLB 内容为：_____，

二级页表页表项地址为：_____，物理地址为：_____。

第二次向地址 0x0401369B 写入内容，

TLB 索引为：_____，完成写之后该项 TLB 内容为：_____

二级页表页表项地址为：_____，物理地址为：_____。

3. （2 分，考察对于虚拟内存独立地址空间的理解）

本学期的 fork bomb 作业中，大家曾用 fork 逼近系统的进程数量上限。下面有一个类似的程序，请仔细阅读程序并填空。

```
#include <stdio.h>
#include <sys/wait.h>
#include <unistd.h>
#define N 4
int main() {
        volatile int pid, cnt = 1;
        for (int i = 0; i < N; i++) {
                if ((pid = fork()) > 0) {
                        cnt++;
        }
        while (wait(NULL) > 0);
        return 0;
}
```

整个过程中，变量 cnt 最大值是_____。假设所有的数据都已经存在于内存中，pid 和 cnt 在同一个物理页中。从第一个进程开始执行 for 语句开始，此过程对于 cnt 的操作至少会导致页表中_____次虚拟页对应的物理页被修改。

答案：

1：1024， 5

2：第一次试图向地址 0xD7416560 写入内容，TLB 索引为 6 ，完成写之后该项 TLB 内容是 0x00A23067 ，二级页表页表项地址为 \ ，物理地址是 0x00A23560 。第二次试图向地址 0x0401369B 写入内容，TLB 索引为 3 ，完成写之后该项 TLB 内容是 0x00BA4067 二级页表页表项所在地址为 0x29DE404C ，物理地址是 0x00BA469B 。

3：5，15

得分

第五题（10 分）虚拟存储

Intel 的 IA32 体系结构采用小端法和二级页表。其中两级页表的大小相同，页大小为 4KB。一级页表和二级页表的表项结构相同，其中页表项后六位的含义如下。

已知一级页表的地址为 0x0c23b000，物理内存中的部分内容如下图所示。

地址	内容	地址	内容	地址	内容	地址	内容
00023000	E0	00023001	BE	00023002	EF	00023003	BE
00023120	83	00023121	C8	00023122	FD	00023123	12
00023200	23	00023201	FD	00023202	BC	00023203	DE
00023320	33	00023321	29	00023322	E5	00023323	D2
00023FF8	29	00023FF9	FF	00023FFA	DE	00023FFB	BC
00055004	03	00055005	D0	00055006	74	00055007	89
0005545C	97	0005545D	C2	0005545E	7B	0005545F	45
00055460	97	00055461	D2	00055462	7B	00055463	45
00055464	97	00055465	E2	00055466	7B	00055467	45
0C23B020	55	0C23B021	EB	0C23B022	AE	0C23B023	24
0C23B040	55	0C23B041	AB	0C23B042	2A	0C23B043	01
0C23B080	05	0C23B081	5D	0C23B082	05	0C23B083	00
0C23B09D	05	0C23B09E	D3	0C23B09F	F2	0C23B0A0	0F
0C23B274	05	0C23B275	3D	0C23B276	02	0C23B277	00
0C23B9FC	25	0C23B9FD	D2	0C23B9FE	14	0C23B9FF	23
2314D200	23	2314D201	12	2314D202	DC	2314D203	0F
2314D220	A9	2314D221	45	2314D222	13	2314D223	D2

2314D4A0	BD	2314D4A1	BC	2314D4A2	88	2314D4A3	D3
2314D890	00	2314D891	2D	2314D892	B3	2314D893	00
24AEE001	07	24AEE002	A0	24AEE003	37	24AEE004	C2
24AEE520	D1	24AEE521	DA	24AEE522	8C	24AEE523	B5
29DE2504	02	29DE2505	AD	29DE2506	FF	29DE2507	56
29DE4400	D0	29DE4401	5C	29DE4402	B4	29DE4403	2A
29DE9402	00	29DE9403	20	29DE9404	73	29DE9405	D4
29DEE500	B0	29DEE501	CD	29DEE502	23	29DEE503	1A

TLB 采用直接映射，TLB 的内容如下所示。

索引	TLB 标记	内容	有效位
0	0x08001	2314d220	1
1	0x01000	24aee520	0
2	0x005AE	00055004	0
3	0x016BA	0c23b09d	1
4	0x0AA00	0005545c	1
5	0x0000A	29dee500	0
6	0x5AE82	00023320	1
7	0x28DFC	00023000	1

1. （2 分）某用户态进程试图写入虚拟地址：0x080016ba。该访问的最后结果是_____。

(a) 该进程成功写入，未触发异常

(b) 该进程触发了一个缺页异常

(c) 该进程触发了一个非法访问异常

2. （2 分）下面描述了具体的访问过程，请填空。如果某个空在访问过程中已不可用，请填入“--”

TLB 的索引为_____，访问为 (a) 命中 (b) 不命中（请勾选）。

1.1 一级页表表项地址为_____。（2 分）

1.2 二级页表表项地址为_____。（2 分）

1.3 最后物理地址为_____。（2 分）

答案：1. (c), 2.1 1 (b), 2.2 0x0C23B080, 2.3 0x00055004, 2.4 8974D6BA

说明：联合考察虚存、高速缓存（通过 TLB 考察）、大端法和小端法的知识。