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）

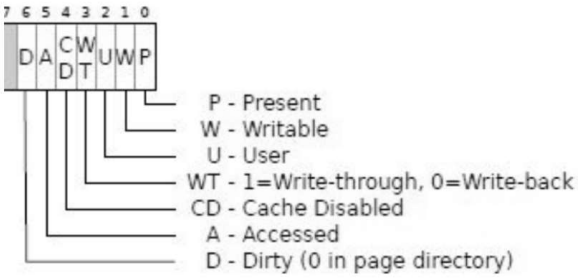
在进行地址翻译的过程中，操作系统需要借助页表(Page Table)的帮助。考虑一个 32 位的系统，页大小是 4KB，页表项(Page Table Entry)大小是 4 字节(Byte)，如果不使用多级页表，常驻内存的页表一共需要_____页。

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。