

# 作业5

本次作业共包含3道大题，总计25分。

## Q1 页表（16分）

已知内存是字节可寻址的、每次内存访问针对的是32-bit的word、虚拟地址24位、物理地址20位、页面大小为4096字节、TLB是二路组相联，共有16个TLB项（即2-way set associative with 16 total entries）。在下面的表格中，所有的数字都是十六进制的。TLB和页表前32项的内容如下：

TLB				Page Table					
Index	Tag	PPN	Valid	VPN	PPN	Valid	VPN	PPN	Valid
0	011	1C	1	00	04	1	10	03	1
	02C	B3	1	01	13	0	11	24	0
1	13C	A4	0	02	28	1	12	1E	1
	0B2	7E	1	03	1F	1	13	08	1
2	001	05	1	04	3E	1	14	02	1
	1A3	B6	0	05	6C	0	15	2A	1
3	002	08	1	06	09	1	16	3C	0
	003	17	1	07	17	0	17	3B	0
4	1C2	21	1	08	24	1	18	1C	1
	013	09	0	09	07	1	19	16	0
5	1CF	38	1	0A	05	1	1A	21	0
	08B	51	1	0B	2D	1	1B	17	1
6	003	7A	1	0C	06	0	1C	22	1
	13C	7F	1	0D	3B	1	1D	2E	0
7	031	B2	0	0E	21	1	1E	7A	1
	0A4	3C	1	0F	08	0	1F	4B	1

### Q1.1 地址格式（2分）

在图上标注出虚拟、物理地址中所包含的字段（下面所列举的字段若存在则标记，若不存在跳过即可）：

- 虚拟地址：VP0、VPN、TLBI、TLBT

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

- 物理地址：PP0、PPN

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

### Q1.2 虚拟地址转换（14分）

对于给定的两个虚拟地址 0x01DBE3、0x9E6CF2，请分别写出 相应的TLB 表项和物理地址，并指出：

- TLB 是否命中（回答 Y 或 N）

- 是否发生 page fault (回答 Y 或 N )
  - 以页表前 32 项为准；即如果不在页表前 32 项内，则发生page fault；系统保证page fault异常处理后可以解决问题
  - 如果发生 page fault，请在 PPN 一项填 - 。
- 获得该地址所存储数据的访问时间（以下简称 "访问时间"）：
  - 假设一次内存访问时间100ns，一次快表（TLB）访问时间为10ns，处理一次缺页需要108 ns （含更新TLB和页表的时间）

	0x01DBE3	0x9E6CF2
VPN		
TLB Index		
TLB Tag		
TLB Hit? (Y/N)		
Page Fault? (Y/N)		
PPN		
访问时间（单位为ns）		

## Q2 重定位（3分）

foo.c 的内容如下所示，其中 test\_call() 共执行5次调用，依次为 test1 ~ test5：

```
// foo.c
extern void (*test1)();

static void test2() {}

void test3() {}

extern void test4();

void test_call(void (*test5)()) {
    test1();
    test2();
    test3();
    test4();
    test5();
}
```

现生成 foo.o 文件，在链接期间需要全局重定位的调用有哪些（即生成的 foo.o 文件中，需要重定位的符号有哪些）？

### Q3 fork() (6分)

阅读程序，并回答问题（每题2分）：

1. 该程序共输出多少行 hello ？

```
void doit() {
    fork();
    fork();
    printf("hello\n");
    return;
}

int main() {
    doit();
    printf("hello\n");
    exit(0);
}
```

2. 该程序共输出多少行 hello ？

```
void doit() {
    if (fork() == 0) {
        fork();
        printf("hello\n");
        exit(0);
    }
    return;
}

int main() {
    doit();
    printf("hello\n");
    exit(0);
}
```

3. 该程序输出的 counter 值应为多少？

```
int counter = 1;
int main() {
    if (fork() == 0) {
        counter--;
        exit(0);
    } else {
        wait(NULL);
        counter++;
        printf("counter = %d\n", counter);
    }
    exit(0);
}
```