第二次作业（part1+part2=59+27=86 分）

Part1（59 分）

一．填空题（每空 1 分，共 12 分）

1. 对于一个磁盘，其平均旋转速率是 15000 RPM，平均寻道时间是 4ms，单个磁道上平均扇区数量是 800，则这个磁盘的平均访问时间是 ___6.005ms____

2. 对于一个磁盘，其有两个扇片，10000 个柱面，每个磁道平均有 400 个扇区，而每个扇区平均有 512 个字节。那么这个磁盘的的容量为 __ 8192MB____

注意：7.63GB 或者 7812.5MB 也算对，按 1024 进位计算的也都算正确。

3. Cache 为 8 路的 2M 容量，B=64，则其 Cache 组的位数 s= ___12___

4. 在现代计算机存储层次体系中，访问速度最快的是 ___寄存器____

5. 若高速缓存的块大小为 B(B>8)字节，向量 v 的元素为 int，则对 v 的步长为 1 的应用的不命中率为__4/B___

6. 某CPU使用32位虚拟地址和4KB大小的页时，需要PTE的数量是 ___2^{20}_____

7. 缓存不命中的种类有 ___强制性（冷）不命中___ 、 __容量不命中_____、 __冲突不命中_____ 。

8. 虚拟页面的状态有 ____未分配____、已缓存、未缓存共 3 种。

9. Linux 虚拟内存区域可以映射到普通文件和 ___匿名文件_____，这两种类型的对象中的一种。

10. 虚拟内存发生缺页时，缺页中断是由 ___MMU（内存管理单元）___ 触发的。

二．分析题（共 47 分）

1. 对于一个机器而言，有如下的假设，内存是字节寻址，并且内存访问是 1 字节的字。地址宽度是 13 位，其高速缓存是 2 路组相联的，块大小是 4 字节，一共有 8 个组。高速缓存的具体内容如图所示。

2路组相联高速缓存

	行0						行1					
组索引	标记位	有效位	字节0	字节1	字节2	字节3	标记位	有效位	字节0	字节1	字节2	字节3
0	09	1	86	30	3F	10	00	0	—	—	—	—
1	45	1	60	4F	E0	23	38	1	00	BC	0B	37
2	EB	0	—	—	—	—	0B	0	—	—	—	—
3	06	0	—	—	—	—	32	1	12	08	7B	AD
4	C7	1	06	78	07	C5	05	1	40	67	C2	3B
5	71	1	0B	DE	18	4B	6E	0	—	—	—	—
6	91	1	A0	B7	26	2D	F0	0	—	—	—	—
7	46	0	—	—	—	—	DE	1	12	C0	88	37

a) 对于该地址格式进行划分，划分出 tag 位，组索引位和块内偏移的区间。

b) 对于地址 0x0E34 ，指出其对应的 tag 位，组索引位和块内偏移的值，并说明高速缓存是否命中，如果命中，写出对应的字节（用 16 进制表示）。

c) 对于地址 0x0DD5，指出其对应的 tag 位，组索引位，并说明高速缓存是否命中，如果命中，写出对应的字节（用 16 进制表示）。

解答：(共 11 分)

（1）块大小是 4 字节，所以块内偏移大小为 2 位。因为有 8 个组，所以组索引位数位 3。所以剩下的 Tag 位为 8 位。（3 分）

Tag								CI			CO	
12	11	10	9	8	7	6	5	4	3	2	1	0

（2）0X0E34 = 0b 011100011010 00 （4 分）

所以，此时组索引为 5，tag 位为 0x71，块内偏移为 0。所以高速缓存命中，对应的字节位 0x0B。

（3）0x0DD5 = 0b011011101010 1 （4 分）

所以，此时组索引为 5，tag 位为 0x6e，块内偏移为 0x1，所以高速缓存未命中。

2. 对于一个直接映射的高速缓存系统，假设其大小是 256 字节，块大小是 16 字节，现在定义三个操作，L 为装载操作，S 为数据存储操作，M 为数据更改操作。L 和 S 最多引发一次缓存 miss，而 M 操作可以看作是对于同一个地址先进行了 L 操作，之后进行了 S 操作。分析下面的操作序列，对于高速缓存的的命中和淘汰情况。（假设高速缓存最开始是空的）

L 10，1
M 20，1
L 22，1
S 18，1
L 110，1
L 210，1
M 12，1

说明：L 10，1 表示对于地址 0x10 位置，进行了一个字节的装载操作。
解答：（共 7 分，每个操作分析 1 分）
大小是 256 字节，块大小是 16 字节，所以 一共有 256/16 = 16 块。因为是直接映射，一共有 16 个组。所以组索引位为 4 位。块内偏移为 4 位。

在最开始的情况下，高速缓存内容为空。
L 10，1 ：0x10 的地址为 0000 0001 0000。此时未命中任何元素。根据组索引，占用了第一块内容。
M 20，1: 0x20 的地址为 0000 0010 0000。此时未命中任何元素。但是会重新装入缓存，之后的 S 操作会命中。
L 22，1: 0x22 的地址为 0000 0010 0010。此时命中。
S 18，1: 0x18 的地址为 0000 0010 1000。此时命中。
L 110，1: 0x110 的地址为 0001 0001 0000。此时不命中。同时进行淘汰替换。
L 210，1: 0x210 的地址为 0010 0001 0000。此时不命中。同时进行淘汰替换。
M 12，1: 0x12 的地址为 0000 0001 0010。此时不命中，同时进行淘汰替换。但是会重新装入缓存，之后的 S 操作会命中。

3. 在一台具有块大小 16 字节（B=16）、整个大小为 1024 字节的直接映射数据缓存的机器上测量如下代码的高速缓存性能：

假设：

- sizeof(int) = 4。
- grid 从内存地址 0 开始。
- 这个高速缓存开始时是空的。

- 唯一的内存访问是对数组 grid 的元素的访问，变量 i、j、total_x 和 total_y 存放在寄存器中。
- 数据结构定义

```
struct position {
    int x;
    int y;
};

struct position grid[16][16];

int total_x = 0, total_y = 0;
int i, j;
```

A. Test 1

```
for (i = 0; i < 16; i++){
    for(j = 0; j < 16; j++){
        total_x += grid[i][j].x;
    }
}

for (i = 0; i < 16; i++){
    for(j = 0; j < 16; j++){
        total_y += grid[i][j].y;
    }
}
```

1. 读总数是多少?
2. 缓存不命中的读总数是多少?
3. 不命中率是多少?

B. Test 2

```
for (i = 0; i < 16; i++){
    for(j = 0; j < 16; j++){
        total_x += grid[i][j].x;
```

```
            total_y += grid[i][j].y;
        }
    }
```

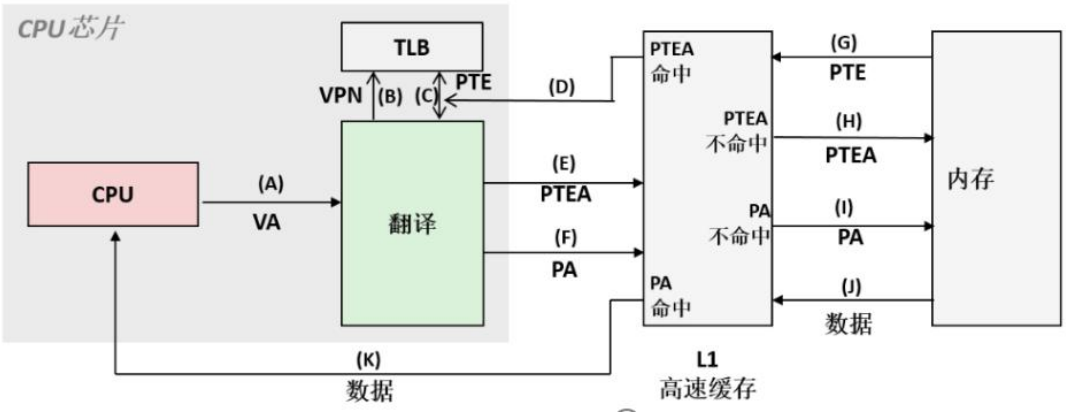
1. 读总数是多少?
2. 缓存不命中的读总数是多少?
3. 不命中率是多少?
4. 如果高速缓存有两倍大，那么不命中率是多少?

解析：（共 7 分，每个问 1 分）
A：读总数：16*16*2 = 512 2.
读不命中次数：256 3.
读不命中率：50%
B：读总数：16*16*2 = 512 2.
读不命中次数：128 3.
读不命中率：25%
缓存容量变大，可以将 grid 全部缓存，并不会发生置换，但是最开始产生的不命中仍然存在，仍然有 128 次，所以应该是 25%。

4. 对于一个地址，高速缓存通常使用地址的中间部分作为组索引，为什么不用高位地址作为组索引?
解答：（共 5 分，按正确程度酌情给分）
如果使用高位作为组索引，那么连续一块内存地址会映射到相同的高速缓存块中。这样并没有完全利用高速缓存中所有有限的地址空间。如果以中间位作为索引，相邻的块映射到不同的高速缓存组中，那么就可以让其均匀映射，充分利用高速缓存。

5. 下图展示了一个虚拟地址的访存过程，每个步骤都用不同的字母表示。请针对下面不同的情况，用字母序列表示不同情况下的执行流程。

（1）TLB 命中，缓存物理地址命中。
（2）TLB 不命中，缓存页表命中，缓存物理地址命中。
（3）TLB 不命中，缓存页表不命中，缓存物理地址不命中。



解答：（共 6 分，每问 2 分）

首先虚拟地址会先访问 MMU，进行地址翻译。然后访问 TLB ，如果 TLB 命中，则可以直接拿到对应的实际物理地址，如果 TLB 未命中，则根据页表地址在缓存页表中寻找相应的位置，如果缓存页表命中，则拿到实际的物理地址，如果缓存页表也未命中，此时则到内存中进行查找。

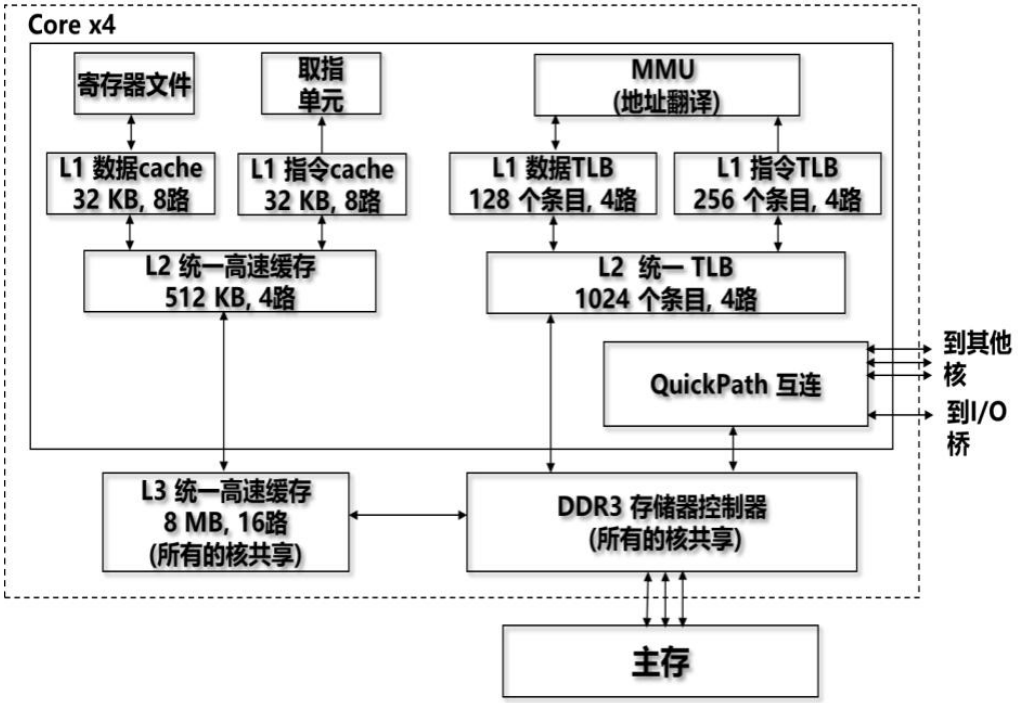
拿到物理地址之后，则在缓存物理地址中寻找，如果命中，则直接返回对应的数据，反之，未命中则在内存中进行寻找。

（1）A B C F K

（2）A B E D F K

（3）A B E H G D F I J K

7.Intel I7 处理器的虚拟地址为 48 位。虚拟内存的页大小是 4KB，物理地址为 52 位，Cache 块大小为 64B。物理内存按照字节寻址。其内部结构如下图所示，依据这个结构，回答问题。



（1）虚拟地址的 VPN 占多少位？一级页表占多少项？L1 数据 TLB 的组索引位数 TLBI 占多少位？
（2）L1 数据 Cache 有多少组？相应的 Tag 位，组索引位和块内偏移位分别是多少？
（3）对于某指令，其访问的虚拟地址为 0x804849B，则该地址对应的 VPO 为多少？对应的 L1 TLBI 位为多少？（用 16 进制表示）
（4）对于某指令，其访问的物理地址为 0x804849B，则该地址访问 L1 Cache 时，CT 位为多少？CO 位为多少？（用 16 进制表示）

解答：（共 11 分）

（1） （3 分）

页大小为 4KB，那么页内偏移为 12 位。所以对应的 VPN 为 36 位。
因为其为四级页表结构，所以其一级页表占 9 位，一共 $2^9 = 512$ 项。L1 数据 TLB 一共有 128 个条目，4 路，所以一共有 32 组，所以 TLBI 占 5 位。

（2） （4 分）

L1 数据 Cache 大小 32KB，8 路，Cache 块大小为 64B，所以一共有 32KB/(8*64B) = 64 组。所以，组索引位为 6 位，块内偏移位 6 位。因为其物理地址为 52 位，所以其余的 40 位为 Tag 位。

（3） （2 分）

因为访问的是虚拟地址。L1 指令 TLB 一共有 256 个条目，4 路，所以，一共有 256/4=64 组，所以 TLBI 为 6 位。虚拟地址为 0x804849B = 1000 0000 0100 1000 0100 1001 1011 所以，VPO 为 0x49B，TLBI 位为 0x01000=0x008。

（4） （2 分）

因为访问的实际物理地址。同（2）分析类似，L1 指令 Cache 的 Tag 位为 40 位，组索引位为 6 位，偏移位为 6 位。物理地址为 0x804849B = 1000 0000 0100 1000 0100 1001 1011 所以 CT 为 0x8048，CO 为 0x1B。

Part2（27 分）

选择题（6 分，每题 1 分）

1. 链接时两个文件同名的弱符号，以（C）为基准
   A. 连接时先出现的
   B. 连接时后出现的
   C. 任一个
   D. 链接报错

2. 链接时两个同名的强符号，以哪种方式处理？（ D ）
   A. 链接时先出现的符号为准
   B. 链接时后出现的符号为准
   C. 任一个符号为准

D．链接报错

3. 以下关于程序中链接“符号”的陈述，错误的是（ B ）

A．赋初值的非静态全局变量是全局强符号

B．赋初值的静态全局变量是全局强符号

C．未赋初值的非静态全局变量是全局弱符号

D．未赋初值的静态全局变量是本地符号

解析：强符号：函数和初始化全局变量；

弱符号：未初始化的全局变量，以及函数声明。

带 static 修饰符的肯定叫做局部或本地变量

4. C 源文件 m1.c 和 m2.c 的代码分别如下所示，编译链接生成可执行文件后执行，结果最可能为（ D ）

$ gcc –o a.out m2.c m1.c ; ./a.out

0x1083020 ； _____； _____

A. 0x1083018, 0x108301c B. 0x1083028, 0x1083024

C. 0x1083024, 0x1083028 D. 0x108301c, 0x1083018

```
// m1.c
#include <stdio.h>
int a1 ;
int a2 = 2 ;
extern int a4 ;
void hello()
{
    printf("%p;", &a1);
    printf("%p;", &a2);
    printf("%p\n", &a4);
}
```

```
//m2.c
int a4 = 10 ;
int main()
{
    extern void hello() ;
    hello() ;
    return 0 ;
}
```

解析：D（a2 和 a4 在.data 中, a1 在.bss 中，按照布局, .data 地址比.bss 要小，又由于 m2.c 的编译先于 m1.c，故选 D 不选 A ）

5. 对于以下一段代码，可能的输出为：（ A ）

```
int count = 0;
int pid = fork();
if (pid == 0){
    printf("count = %d\n",--count);
}
else{
    printf("count = %d\n",++count);
}
printf("count = %d\n",++count);
```

A.1 2 -1 0　　B.0 0 -1 1

C.1 -1 0 0　　D.0 -1 1 2

6. Linux 进程终止的原因可能是(D)

A.收到一个信号　　B.从主程序返回　　C.执行 exit 函数　　D.以上都是

二．填空题（7 分，每空 1 分）

1. C 语句中的全局变量，在__链接/重定位_____阶段被定位到一个确定的内存地址。
2. 子程序运行结束会向父进程发送__SIGCHLD_____信号。
3. 向指定进程发送信号的 linux 命令是__kill_____。
4. Unix 内核通过三个表项描述符表、打开文件表、V-nod 表 表示打开文件。
5. Unix 内核通过调用函数___dup2_____实现 I/O 重定向。

三．分析题（本题 14 分，写对一个给 1 分）

请阅读以下程序，然后回答问题（假设程序中的函数调用都可以正确执行）：

```
int main() {
        printf("A\n");
        if (fork() == 0) {
            printf("B\n");
        }else {
            printf("C\n");
           A
        }
        printf("D\n"); exit(0);
    }
```

（1）如果程序中的 A 位置的代码为空，列出所有可能的输出结果：

分别是 ABDCD ABCDD ACBDD ACDBD

（2）如果程序中的 A 位置的代码为：

waitpid(-1, NULL, 0);

列出所有可能的输出结果：

ABDCD ABCDD ACBDD

解析：waitpid(-1, NULL, 0);父进程收回所有子进程，挂起当前进程，直到所有子进程结束。所以 B 后面一定有两个 DD

（3）如果程序中的 A 位置的代码为：

printf("E\n");

列出所有可能的输出结果：

ABDCED ABCEDD ACEBDD ACEDBD ACBEDD ACBDED ABCDED