

1、unmapped 空间已经 512MB 了，是如何对应到 32MB 的物理空间的？

A: unmapped 使用直接偏移量的方式映射物理地址，不经过 TLB。所以只有 80000000 和 a0000000 开头的 32MB 可以映射到物理空间，后面的地址不映射到物理地址。不过实际上 a0000000~bfffffff 这段空间后面的某些地址可能会映射到硬件的其他器件上。

2、任务 2 中超出物理空间大小的虚拟页框无法映射到物理页框，如何处理？

A: 可以不映射物理页框，并标为 invalid，如果访问到 invalid 的页表项就打印报错信息。任务 2 中只要求对全部虚拟地址有一个响应，任务 3 的时候所有页表项都初始化为 invalid，访问 invalid 页表项需要分配地址映射，只有没有空闲物理空间时才报错。

3、哪些物理页框要被用来做虚实映射？

A: 由同学们自己来设计，只要不和内核代码冲突就可，不要求把物理内存全部都利用上。

4、体系结构书上写的 TLB refill 例外不判断页表项是否 valid，如果 invalid 的话再次触发 TLB invalid 例外，OS 研讨课中如何处理？

A: TLB refill 例外并不强制要求这么实现，完全可以由同学们灵活发挥。龙芯处理器中这样实现 TLB refill 的原因是考虑到 TLB refill 例外的触发次数很多，因此在 TLB refill 例外中减少一次判断可以提高例外处理程序的处理速度。同时，TLB refill 单独一个例外向量入口的原因也是为了减少判断，提高这个例外的处理速度。请注意：TLB refill 例外的向量入口是 0x80000000，而非其他例外的 0x80000180，包括 TLB invalid 例外的入口也是 0x80000180。

5、什么时候需要做 TLB flush？

A: 在页表项的内容发生改变，而该页表项的内容在 TLB 中存在时，需要 flush 这一项 TLB 项。在本实验的任务 1、2、3 中，改变页表项内容且该项内容在 TLB 中的情况不会发生，所以不需要考虑 TLB flush。

6、测试用例 process2 中的 printk 会导致打印错误？

A: 如果 printk 在向串口输出控制符时被时钟中断，会导致输出不完整。可以改为 printf。