本周主要学习了黑书的虚拟存储器

首先，TLB要从虚拟存储器说起。何为虚拟存储器？它是“一种将贮存用作辅助存储器高速缓存的技术”。Cache可以对程序中最近访问的代码和数据提供快速访问。主存也可以为通常由磁盘实现的辅助存储器充当cache）。简单地说，虚拟存储器就是实现程序地址到物理地址的转换。

其主要目的是允许云计算在多个虚拟机之间有效而安全地共享存储器。

其目的之二是允许单用户程序使用超过主要存储器的容量。

虚拟存储器与cache的工作原理一样，但某些名词不同。

以下是一些重要名词：

页：虚拟存储器中的块

**缺页**：访问的页不在主存储器中

**虚拟地址**：虚拟空间的地址

**地址转换**：虚拟地址->物理地址

**物理地址**：主存中的地址

图示

描述已自动生成

**虚拟存储器的重定位**：将程序变成一组固定大小的页，不必再在主存中寻找连续的块来放置程序，而只需要找到足够数量的页。

虚拟存储器中的地址分为虚页号和页偏移，在地址转换中，虚拟地址通过转换构成物理地址的高位部分，而页偏移不需要转换，直接作为物理地址的低位部分。页偏移的位数决定了页的大小。

图示

描述已自动生成

上图为虚拟地址到物理地址的映射，页偏移为12位因此页大小为4kb，物理页数为218个，虚拟页有220个，因此导致了虚拟页数是物理页数的4倍，造成了潜在的**页缺失**。

一次缺页将花费数百万个时钟周期。为了避免缺页，操作系统可以选择任意一个页进行替换，被被选择的显然是长时间不用到的页，具体方法是在页中设置标志位，每隔一段时间后进行复位，使用之后再置1，检测一段时间内是否常为0（即LRU替换策略）。

在虚拟存储系统中，使用页表作为索引存储器的表对页进行定位。页表保存在主存中，使用虚页号索引。页表中有多位页表项，每个页表项具有一位的有效位，若该位无效，则说明该页不在主存中，就发生一次缺页。如果该位为有效位，则说明该页在主存中，并且该项包含物理页号。

图片包含 图示

描述已自动生成

虚页号->索引页表获得物理地址

**页的存放与查找**

**缺页故障**

引入交换区。LRU策略放入

**TLB（Translation-Lookaside Buffer）【快表】**

用于记录最近使用地址的映射信息的高速缓存，从而可以避免每次都要访问快表