作业 9

9.1 一台机器虚存采用分段机制，物理内存当前的空闲空间如下(按物理地址由小到大的顺序):12MB, 5MB, 18MB, 20MB, 8MB, 9MB, 10MB 和 15MB。此时要为三个段分配空间(按时间先后顺序): 段 A 申请 12MB，段 B 申请 10MB，段 C 申请 9MB。请分别给出采用 Best Fit，Worst Fit，First Fit 和 Next Fit 算法下，每次分配成的空闲空间状态(按物理地址由小到大顺序)，以及每次分配所需的比较次数。

Best Fit（最佳适应）算法:

每次分配时，选择能够容纳该段大小且剩余空间最小的空闲区域。

段 A（12MB）: 分配到 12MB 的空闲区域，剩余空间为 0MB。比较次数: 1 次。

段 B（10MB）: 分配到 10MB 的空闲区域，剩余空间为 0MB。比较次数: 6 次。（除去已分配给段 A 的）

段 C（9MB）: 分配到 9MB 的空闲区域，剩余空间为 0MB。比较次数: 5 次。

Worst Fit（最差适应）算法:

每次分配时，选择能够容纳该段大小且剩余空间最大的空闲区域。

分配过程:

段 A（12MB）: 分配到 20MB 的空闲区域，剩余空间为 8MB。比较次数: 8 次。

段 B（10MB）: 分配到 18MB 的空闲区域，剩余空间为 8MB。比较次数: 8 次。

段 C（9MB）: 分配到 15MB 的空闲区域，剩余空间为 6MB。比较次数: 8 次。（分配给 A、B 剩下的空间也要比较，要比较所有的空闲内存区域，以确定剩余空间最大的空闲区域）

First Fit（首次适应）算法:

每次分配时，从头开始查找第一个能够容纳该段大小的空闲区域。

分配过程:

段 A（12MB）: 分配到 12MB 的空闲区域，剩余空间为 0MB。比较次数 1 次。

段 B（10MB）: 分配到 18MB 的空闲区域，剩余空间为 8MB。比较次数: 2 次。（除去已分配给段 A 的）

段 C（9MB）: 分配到 20MB 的空闲区域，剩余空间为 11MB。比较次数: 3 次。

Next Fit（下一次适应）算法:

每次分配时，从上一次分配的位置开始查找第一个能够容纳该段大小的空闲区域。

分配过程:

段 A（12MB）：分配到 12MB 的空闲区域，剩余空间为 0MB。比较次数：1 次。

段 B（10MB）：分配到 18MB 的空闲区域，剩余空间为 8MB。比较次数：2 次。

段 C（9MB）：分配到 20MB 的空闲区域，剩余空间为 11MB。比较次数：2 次。（包括比较上次匹配剩下的空闲区域）

9.2 假设一台计算机使用 32-bit 的虚拟地址空间和三级页表，虚地址的划分为 8-bit | 6-bit | 6-bit | 12-bit（注：8 bit 对应为第一级页表的地址，以此类推），请计算：

（1）该计算机系统的页大小是多少？

（2）该三级页表一共能索引多少个页？

（3）现有一个程序的代码段大小为 8KB，数据段为 32KB，栈大小为 8KB，则在使用上述三级页表时，最少需要占用多少个物理页框？最多会占用多少个物理页框？（注：假设程序各段在地址空间中的布局可以自行决定）

（4）在上述（3）中，假设该计算机使用一级页表进行地址空间管理，则（3）中的程序需要占用多少个物理页框？

（1）、页大小：

虚拟地址空间为 32 位，对应 4 GB。页大小为 4 KB（2^{12}字节）。

（2）三级页表索引的总页数：

每个级别的页表对应虚拟地址的一部分。

根据地址划分：8 位 | 6 位 | 6 位 | 12 位，我们有：

第一级页表：2^8个条目（256 个条目）

第二级页表：2^6个条目（64 个条目）

第三级页表：2^6个条目（64 个条目）

三级页表索引的总页数是这些条目的乘积：256 * 64 * 64 = 1,048,576 页。

或者直接利用$2^{32} \div 2^{12} = 2^{20} = 1048576$ 页。

（3）、

代码段大小：8 KB

数据段大小：32 KB

栈大小：8 KB

程序所需的总内存：8 KB + 32 KB + 8 KB = 48 KB。

如果使用三级页表，由于页大小为 4 KB，如果按页对齐最少需要占用 12 个物理页，索引连续的物理页表，最少即，每级索引页表只使用一个物理页表（因为连续），共需要 1+1+1+12=15 个物理页框。

如果每个段都位于不同的物理页位置，且不一定按页对齐，代码段需要$\lfloor 8KB/4KB \rfloor + 1 = 3$个物理页，数据段$\lfloor 32KB/4KB \rfloor + 1 = 9$个物理页，栈$\lfloor 8KB/4KB \rfloor + 1 = 3$个物理页。共需要 15 个物理页，它们三个段的页是分别连续的（考虑这样的情况，是因为保证访存的正确性，例如代码段需要 pc+4，数据段对应不同大小的数据类型，栈更不必说）。

所以这样最多需要 1+3+3+12=19 个物理页框。

如果考虑同一个段可以不连续存放在页内，那么则需要共 1+12+12+12=37 个物理页框。

（4）、使用一级页表：

在这种情况下，假设一个一级页表管理。因此，程序仍然需要占用 12 个物理页框。

9.3 假设一台计算机上运行一个进程 A，该进程的地址空间大小为 4 MB（页大小为 4KB）。该计算机使用线性页表记录进程 A 的虚实映射关系，并且将 A 的页表都保存在内存中。该计算机 CPU 的 TLB 大小为 32 项，每项 4B，一次 TLB 查询或 TLB 填充的延迟均为 5 ns，请计算：

（1）假设该计算机使用软件处理 TLB miss，且操作系统进行一次页表查询的平均延迟为 100 ns，如果想让虚实地址映射的平均延迟为 40 ns，那么 TLB 的命中率应为多少？如果想让虚实地址映射的平均延迟不超过 20 ns，那么 TLB 的命中率应为多少？（上述各项操作的延迟不变）

答：

假设 TLB 的命中率为 h。

已知，TLB 查询的平均延迟为 5 ns。页表查询的平均延迟为 100 ns。

由题，$T_{平均时延} = h * T_{TLB查询时延} + (1-h) * (T_{TLB查询时延} + T_{页表查询延迟} + T_{TLB重填延迟} + T_{TLB查询时延})$

命中则只需要 TLB 查询时延，若未命中，则需要包括 TLB 查询，页表查询时延，TLB 重填时延。代入得，若想让虚实映射平均延迟至少为 40ns，则命中率至少为 68.18%

若为 20ns，则命中率至少为：86.36%

9.4 现有如下 C 程序

```
uint32 X[N];
int step = M, i = 0;
for(i=0;i<N;i+=step) X[i] = X[i] + 1;
```

请计算：

（1） 假设该程序运行在一台计算机上，该计算机的虚址空间为 32-bit，物理地址空间为 2 GB，页大小为 4 KB，如果采用一级页表，则该页表的页表项一共有多少？

（2） 假设该计算机的 CPU 的 TLB 大小为 32 项，每项 4B，那么题述程序中的 M 和 N 取值为多少时，会使得程序中循环的每一次执行都会触发 TLB miss？（假设 TLB 初始为空）

（3） 在（2）中，M 和 N 取值多少时，会使得程序中的循环执行时 TLB hit 最多？（假设 TLB 初始为空）

答：

（1）虚拟地址空间大小 = 2^32 bit = 4 GB ，物理地址空间大小 = 2 GB ，页大小 = 4 KB = 2^12 B。

因此，页表项的数量 = 虚拟地址空间大小 / 页大小 = 4 GB / 4 KB = 1M（即 2^20）。

（2）TLB 大小为 32 项，每项 4B，一个 TLB 可以存储 32 个虚拟页到物理页的映射关系，每页可以存储 4KB 即 1024 个 uint32 类型的数据结构。假设数组的起始地址是以页对齐的，为了使得程序中的循环的每一次执行都会触发 TLB miss，注意此时 TLB 表项为空，需要保证每次循环访问的地址都至少大于 TLB 存储的项数，即步长 M 需要大于等于2^{10}，且数组 X 的大小 N 需要大于 TLB 每项可以存储的 uint32 数量。因此，M 需要大于等于2^{10}，N 需要大于2^{10}。

（3）为了使得程序中的循环执行时 TLB hit 最多，需要尽可能让每次循环访问的地址都在 TLB 已经存储的页上。因此，当 M=1，只要 N 不超过一个页中的元素数量（即 4KB/4B=1024），每次循环都会在同一页中，从而最大化 TLB hit，即每次访存都命中。（注意到，这需要保证数组的起始地址按页对齐。否则需要额外的 TLB 项）。所以 M 此时的值为 1，N 的值可以为 1024 的倍数，TLBhit 命中最多，除了每次循环开始第一次需要重填，之后均会 hit，次数为 1023.

注意，以上均假设只存在一级 TLB 结构。