**第五章 虚拟存储器**

**1、什么是程序局部性原理？程序局部性主要体现在哪里方面？**

答：程序局部性原理是指程序在执行时将呈现出局部性规律，即在一较短的时间内，程序的执行仅局限于某个部分，相应地，它所访问的存储空间也局限在某个区域。局部性主要表现在以下2个方面：

（1）时间局部性。如果程序中的某条指令一旦执行，则不久以后该指令可能再次执行；如果某个数据被访问，则不久以后该数据可能被再次访问。产生局部性的典型原因是程序中存在中大量的循环操作。

（2）空间局部性。一旦程序访问了某个存储单元，则不久以后，其附近的存储单元也将被访问，即程序在一段时间内所访问的地址，可能集中在一定的范围内，其典型情况是程序的顺序执行。

**2、什么是虚拟存储器？虚拟存储区有什么特征？**

答：虚拟存储器是指具有请求调入功能和置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。其逻辑容量由内存容量和外存容量之和所决定，其运行速度接近于内存速度，而每位的成本又接近于外存。虚拟存储器有以下特征：

（1）多次性。一个作业中的程序和数据，无需在作业运行时一次性地全部装入内存，允许被分成多次调入内存运行，即只需将当前要运行的那部分程序和数据装入内存即可开始运行。

（2）对换性。一个作业中的程序和数据，无需在作业运行时一直常驻内存，允许在作业的运行过程中进行换进换出。即在进程运行期间，运行将那些暂时不使用的代码和数据从内存调至外存的对换去（换出），待需要时再将它们从外存调至内存（换进）。

（3）虚拟性。指能够从逻辑上扩充内存容量，使用户所看到的内存容量远大于实际内存容量。虚拟性以多次性和对换性位基础。

**3、在请求分页系统中，页表应包括哪些数据项？每项的作用是什么？**

答：页表应包括：页号、物理块号、状态位P、访问字段A、修改位M和外存地址。

其中状态位P 指示该页是否调入内存，供程序访问时参考；访问字段A 用于记录本页在一

段时间内被访问的次数，或最近已有多长时间未被访问，提供给置换算法选择换出页面时参

考；修改位M 表示该页在调入内存后是否被修改过；外存地址用于指出该页在外存上的地

址，通常是物理块号，供调入该页时使用。

**4、在请求分页系统中，应从何处将所需页面调入内存？**

答：请求分页系统中的缺页从何处调入内存分三种情况：

（1）系统拥有足够对换区空间时，可以全部从对换区调入所需页面，提高调页速度。在进程运行前将与该进程有关的文件从文件区拷贝到对换区。

（2）系统缺少足够对换区空间时，不被修改的文件直接从文件区调入；当换出这些页面时，未被修改的不必换出，再调入时，仍从文件区直接调入。对于可能修改的，在换出时便调到对换区，以后需要时再从对换区调入。

（3）UNIX 方式。未运行页面从文件区调入。曾经运行过但被换出页面，下次从对换区调入。UNIX 系统允许页面共享，某进程请求的页面有可能已调入内存，直接使用不再调入。

**5、简述产生“抖动”的原因以及“抖动”的预防方法。**

答：产生“抖动”的原因：同时在系统中运行的进程太多，分配给每一个进程的物理快太少，不能满足进程正常运行的基本要求，致使每个进程在运行时，频繁地出现缺页，必须请求系统将所缺少的页面调入内存。这使得系统中排队等待页面调进/调出的进程数量增加，对磁盘的有效访问也随之增加，造成每个进程的大部分时间都用于页面的换进/换出。

“抖动”预防方法：（1）采取局部置换策略；（2）把工作集算法融入到处理机调度中；（3）利用“L=S”准则调节缺页率；（4）选择暂停的进程。

**6.在一个请求分页虚拟存储系统中，用户编程空间32个页，页长1KB，内存空间16KB。如果应用程序有10页长，若已知逻辑地址页号0，1，2，3已分得物理块4，7，8，10，试把虚拟地址0AC5H、1AC5H转换为物理地址。**

答：0A5CH对应2号页，在8号物理块中，转换为物理地址22C5H;

1AC5H对应6号页，不在内存，不能转换。

**7、在一个请求分页系统中，假如一个作业的页面走向是2,3,2,1,5,2,4,5,3,2,5,2。目前没有任何页面装入内存，当分配给该作业的物理块数为3时，请分别计算采用OPT算法、FIFO算法、LRU算法、CLOCK算法时，访问过程中所发生的缺页次数和缺页率。**

答（1）OPT算法

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2 | 3 | 2 | 1 | 5 | 2 | 4 | 5 | 3 | 2 | 5 | 2 |
| 页框1 | 2 | 2 | 2 | 2 | 2 | 2 | 4 | 4 | 4 | 2 | 2 | 2 |
| 页框2 |  | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 页框3 |  |  |  | 1 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 是否缺页 | 是 | 是 |  | 是 | 是 |  | 是 |  |  | 是 |  |  |

缺页6次，缺页率=6/12=50%

（2）FIFO算法

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2 | 3 | 2 | 1 | 5 | 2 | 4 | 5 | 3 | 2 | 5 | 2 |
| 页框1 | 2 | 2 | 2 | 2 | 3 | 1 | 5 | 5 | 2 | 2 | 4 | 3 |
| 页框2 |  | 3 | 3 | 3 | 1 | 5 | 2 | 2 | 4 | 4 | 3 | 5 |
| 页框3 |  |  |  | 1 | 5 | 2 | 4 | 4 | 3 | 3 | 5 | 2 |
| 是否缺页 | 是 | 是 |  | 是 | 是 | 是 | 是 |  | 是 |  | 是 | 是 |

缺页9次，缺页率=9/12=75%

（3）LRU算法

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2 | 3 | 2 | 1 | 5 | 2 | 4 | 5 | 3 | 2 | 5 | 2 |
| 页框1 | 2 | 3 | 2 | 1 | 5 | 2 | 4 | 5 | 3 | 2 | 5 | 2 |
| 页框2 |  | 2 | 3 | 2 | 1 | 5 | 2 | 4 | 5 | 3 | 2 | 5 |
| 页框3 |  |  |  | 3 | 2 | 1 | 5 | 2 | 4 | 5 | 3 | 3 |
| 是否缺页 | 是 | 是 |  | 是 | 是 |  | 是 |  | 是 | 是 |  |  |

缺页7次，缺页率=7/12=58.3%

（4）CLOCK算法

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 2 | 3 | 2 | 1 | 5 | 2 | 4 | 5 | 3 | 2 | 5 | 2 |
| 页框1 | 2\* | 2\* | 2\* | ->2\* | 5\* | 5\* | ->5\* | ->5\* | 3\* | 3\* | ->3\* | ->3\* |
| 页框2 | -> | 3\* | 3\* | 3\* | ->3 | 2\* | 2\* | 2\* | ->2 | ->2\* | 2 | 2\* |
| 页框3 |  | -> | -> | 1\* | 1 | ->1 | 4\* | 4\* | 4 | 4 | 5\* | 5\* |
| 是否缺页 | 是 | 是 |  | 是 | 是 | 是 | 是 |  | 是 |  | 是 |  |

缺页8次，缺页率=8/12=66.7%