

第4章 存储器管理 习题参考答案

1.为什么要进行内存管理？要管理与内存相关的哪几方面问题？

答：内存是计算机系统中的重要部件，是计算机内部四种资源之一。现代操作系统依靠内存容纳多任务以并发的方式工作，内存的性能制约着整个计算机系统的性能。从系统启动开始，操作系统就需要不断地进行着内存的分配和回收等操作，所以操作系统中需要对内存这种资源进行管理，充分发挥内存的性能和提高资源利用率。

要管理四个方面的内存问题：内存映射、内存分配/回收、内存扩充、内存共享和保护。内存映射完成虚拟内存和物理内存之间的映射关系；内存分配/回收实现物理内存的分配和回收；内存扩充是借助辅助存储器弥补和缓和内存不足的问题，在逻辑上扩充内存容量；内存的共享和保护使进程可以共享使用内存中的模块，同时保证进程之间的隔离安全。

2.程序装载和链接各有哪几种方式？

答：根据链接工作进行的时机，程序的链接方式有三种：

- 静态链接：指程序在装入内存执行前就将目标模块与库模块链接成为一个完整的可执行程序，所以无法单独进行模块的升级，也不能共享模块。

- 装入时动态链接：在程序装入内存执行时边装入边链接，可以进行模块单独升级，仍然不能实现模块的共享。

- 运行时动态链接：待执行到需要的模块或者库的时候才进行相应的链接，既可以方便地进行模块更新，也可以实现模块共享。并且可减小程序尺寸和启动装入量，加快程序启动速度并节省内存空间。

根据地址映射进行的时机，程序的装入方式也有三种：

- 绝对装入：将程序加载装入到编译时就指定的地址开始的内存空间中，装入及运行过程中都不能改变这个起始地址，程序中使用的是绝对地址，适用于固件程序。

- 静态重定位装入：程序在装入内存时，根据起始地址，把相关的所有目标模块装配在一起，并且修改模块中各逻辑地址为统一物理地址空间中的实际物理地址。一次性完成地址映射，之后不会再进行地址映射，程序装入内存后就不能移动位置。

- 动态重定位装入：装入时程序内使用逻辑地址，边执行指令边进行地址映射，转换成物理地址。可以加快指令执行速度，需要专门的地址映射机构。

3.某程序执行过程中提示有 **dll** 文件缺失，为什么有文件缺失的程序还能够执行？如何解决？

答：**dll** 文件是 Windows 操作系统中使用的动态链接库文件，说明程序采用的是动态链接的方式工作，当程序没有使用到该库文件的时候就无需对其进行链接，所以即使该库文件缺失也仍然可以执行程序的其他部分。

要解决该错误，需要先获取得到缺失的 **dll** 动态链接库文件，然后复制到 Windows 系统

相应目录中，如 C:\windows\system32，并进行注册即可。

4.可变分区分配时有哪几种基本分配算法？对于每一种算法，回答下列问题：

- (1) 应如何将各空闲分区链接成空闲分区链？
- (2) 在回收内存时，可能出现哪几种情况？应怎样处理这些情况？
- (3) 分析该算法的内存管理性能。

答：

(1) 首次适应算法。空闲分区按照分区起始地址从低到高排序后构成链表。

最佳适应算法。空闲分区按照分区大小从小到大排序后构成链表。

最坏适应算法。空闲分区按照分区大小从大到小排序后构成链表。

(2) 三种算法回收分区时需要检查待回收分区邻接分区的情况，共有四种：上邻接、下邻接、上下邻接和上下都不邻接。对于空闲分区链进行相应的节点修改。上邻接需要修改上邻接空闲节点的大小；下邻接需要修改下邻接空闲节点的起始地址和大小；上下邻接需要修改上邻接空闲节点的大小为上下邻接空闲分区大小与回收区的总和，然后删除原下邻接节点；无邻接需要增加一个新节点，起始地址就是回收区的起始地址，大小就是回收区的大小。

(3) 首次适应算法在低端内存频繁划分，会导致低端内存区域出现碎片，降低内存利用率，但是可能在高端内存保留较大的连续空间；最佳适应算法选择大小最接近的分区，所以留下分区也最小，可能导致碎片；最坏适应算法选择大小相差最大的分区，留下的分区尽量大，以期提高可利用率，但是反而可能导致缺少较大的连续空间。

5.什么是碎片？碎片是如何产生的？碎片对系统性能有什么影响？

答：碎片是指尺寸很小以致于难以利用的内存空间。内存碎片包括内存碎片和外部碎片，动态分区的时候在分区之间可能出现小空间难以分配利用，成为外部碎片；在固定分区或者分页系统中，分配的区域大小是固定的，如果装入的数据或者程序没有占据整个区域，就产生内部碎片。碎片会导致内存的实际使用量减少，影响内存的利用率。

6.可变分区回收时有哪几种情况？哪种情况会减少空闲分区数量？

答：可变分区回收时有四种情况：上邻接、下邻接、上下邻接和无邻接。其中上下邻接会减少空闲分区的数量。

7.什么时候需要使用紧凑？对系统有怎样的影响？

答：当系统中有较多碎片导致系统有较多分散的空闲内存空间但是却不能利用，不能容纳更多进程，此时通过紧凑可以把分散的小空闲分区合并成为连续的空闲内存分区。进行紧凑的时候系统要移动内存中的进程，因此系统不能继续执行其他进程，直到完成紧凑。

8.分页存储管理系统中页面大小的选择对进程有什么影响？

答：分页存储管理系统中的页面大小一般是 2^k 字节大小，如果页面大小太小，则进程

包含的页面数量会急剧增加，同时页表也变大，并且可能需要更多级别分页；如果页面大小太大，则页内碎片大小也增加，导致内存利用率下降。

9.为什么要使用多级页表？32 位系统使用几级页表？如何进行地址转换？

答：页表存储在内存中的页面上，当页表项数量超过一页的容量的时候，就需要两级或者多级页表。32 位系统通常使用 2 级页表。

进行地址转换的时候先根据页表寄存器中外部页表地址找到外部页表，根据外部页号检索外部页表，找到页表（其中一页）；然后根据外部页内地址检索页表，获得物理块号；最后用物理块号与页内地址合并构成实际物理地址。

10.在 64 位现代操作系统中，采用几级页表？其逻辑地址的结构是怎样的？

答：在 64 位操作系统中通常不会使用 64 位地址，例如使用 48 位，因此采用 4 级页表即可。其逻辑地址结构包括：|一级页号（9 位）|二级页号（9 位）|三级页号（9 位）|四级页号（9 位）|页内地址（12 位）。

11.在 32 位操作系统中，用户程序可以使用的虚拟地址空间是多大？

答：在 32 位操作系统中，用户程序可以使用的虚拟地址空间是 $2^{32}=4\text{GB}$ 。

12.在 32 位的 Windows 7 操作系统中怎样启用管理超过 4GB 的内存空间？

答：启用 PAE（Physical Address Extension），支持大于 4GB 内存，并且更改 Windows 默认设置，允许为应用程序分配大于 2GB 的内存。

13.什么是程序执行的局部性原理？为什么有这样的规律？

答：局部性包含两种：时间局部性和空间局部性。时间局部性是指刚访问的指令或者数据很快就会再次被访问；空间局部性是指刚访问的指令或者数据，其周围或者附近的指令或者数据马上就会被访问。

程序中的循环结构导致指令或者数据访问的时间局部性，程序的顺序执行导致指令访问的空间局部性，对于数组的访问也可以导致空间局部性。

14.对比分页与分段两种方式看待进程的差别。

答：页面只是容纳进程的物理容器，页面数量众多，并且相互之间并没有逻辑关系，要进行部分程序或者数据的共享和保护是很困难的，但是分页系统的内存利用率比较高，分页系统中的逻辑地址是一维线性地址；分段是从程序员的角度看待进程构成的一种方法，段中包含的是程序中自然划分的一组具有完整逻辑意义的信息集合，通过段的方式看待进程，可以通过共享段表更方便地进行段的共享和保护，段中的逻辑地址是二维地址。

15.为什么分页系统中增加快表 TLB 可以加快地址转换的速度？

答：快表 TLB 是访问速度接近 CPU 的高速存储器，在其中缓存了部分页表项，在进行地址映射的时候先根据页号检索 TLB，如果命中，就无需访存了。当 TLB 的命中率很高的

时候，地址转换中查询页表的速度接近访问 TLB 的速度，可以大大提高地址转换的速度。

16. 什么是 Belady 异常现象？在哪种置换算法中可能会出现这样的情况？

答：当分配的物理块数量增加的时候，缺页率没有不降反而升高的异常现象称为 Belady 异常现象。只有在 FIFO 页面置换算法中可能会出现这种异常现象。

17. 分段存储管理系统中进行地址转换的时候需要进行几次越界检查？如何检查？

答：分段系统在进行地址映射的时候需要进行两次越界检查，第一次是根据段号和段表长度判断是否越界，第二次是根据段内地址和段长判断是否越界。

18. 为什么要引入虚拟存储器技术？虚拟存储器系统对于系统的性能有怎样的影响？

答：现代计算机系统是多任务系统，同时容纳很多进行，内存的总量通常是不足以把这些进程全部装入的，为了容纳更多进行并发执行，并且提高内存利用率，需要引入虚拟存储器技术，使用部分辅存空间弥补内存空间的不足。虚拟存储器将暂时不能执行的进程换出到辅存上，必要的时候再换入，由于涉及到内存与辅存之间的数据传输，必须经过 I/O 系统，将会大大降低系统的访存性能。

19. 缺页中断与一般中断有什么区别？

答：缺页中断与一般中断一样都需要保护 CPU 现场、跳转到缺页中断处理程序、恢复 CPU 现场等步骤，但是缺页中断特殊之处在于：

缺页中断是在指令执行期间产生，CPU 一般是在一条指令执行结束后检查是否有中断需要处理，但是缺页中断的产生是在指令执行过程中发现所要访问的指令或者数据不在内存中，因此必须在指令执行中处理中断，执行调页过程。

一条指令在执行期间可能产生多次缺页中断。由于一条指令本身以及涉及的数据都可能因为所在页面不在内存中而产生缺页中断，所以可能多次发生缺页中断。

20. 在段页式存储管理系统的段表中没有段长一项，那么地址转换时如何判断逻辑地址是否会段长越界？

答：段页式存储管理系统的段表中没有段长，但是变为页表起始地址和页表长度。当进行地址映射的时候，先根据段号和段表长度的比较可以进行第一次越界判断；然后根据段号可以查到对应段的页表起始地址和页表长度，由段内页号和页表长度的比较，就可以进行第二次逻辑地址是否越界的判断。

21. 为了实现请求分页虚拟存储系统，需要对页表进行哪些扩充？说明扩充部分的作用。

答：支持分页虚拟存储系统的页表需要增加状态位、访问字段、修改位和外存地址。各部分的作用是：

状态位：表示页面是否在内存，若不在内存则产生缺页中断，也称为中断位。

访问字段：记录页面被访问的情况，依据所采用的页面置换算法，可能是被访问次数或

者未访问时间，供页面置换时参考。

修改位：表示页面装入内存后是否被修改过，供页面置换时参考。

外存地址：存储页面在外存上的地址，通常是物理块号，供调入页面时参考。

22. 某分页系统页面大小为 2KB，进程 A 的 0, 1,2,3 号页面分别装在 2,7,4,8 内存块中，请回答如下问题，并给出计算过程：

(1) 虚拟地址为 02A5 对应的物理地址是多少？

(2) 物理地址为 251D 对应的虚拟地址为多少？

(3) 若进程 A 长度为 8 页，试图写数据到虚拟地址 2A3D 对应的内存单元，然后再从该地址读取数据，这两次内存访问能否正常执行？为什么？

解：

(1) 页面大小为 2KB=800H （转换成 16 进制）

页号 $P=02A5/800=0$ ，页内地址 $W=02A5\%800=2A5$ ，查找页表得块号 2。

物理地址= $2\times800+2A5=12A5$

(2) 物理地址 251D 对应的虚拟地址为多少？

物理块号= $251D/800=4$ ，块内地址= $251D\%800=51D$ ，查页表得页号 2。

虚拟地址= $2\times800+51D=151D$

(3) 若进程 A 长度为 8 页，试图写数据到虚拟地址 2A3D 对应的内存单元，然后再从该地址读取数据，这两次内存访问能否正常执行？为什么？

不能。

页号= $5A3C/800=B$ ，页内地址： $5A3C\%800=23C$

$B>8$ ，超出进程 A 的页表长度。

23. 假设某进程有如下页面走向：3 2 1 0 3 2 4 3 2 1 0 4，请基于 FIFO 算法考察物理块分别为 3 和 4 时的缺页及置换情况。（Belady 现象）

答：

3 个物理块的情况：

页面走向	3	2	1	0	3	2	4	3	2	1	0	4
物理块	3	3	3	0	0	0	4			4	4	
		2	2	2	3	3	3			1	1	
			1	1	1	2	2			2	0	
缺页	+	+	+	+	+	+	+			+	+	

缺页 9 次。

4 个物理块的情况：

页面走向	3	2	1	0	3	2	4	3	2	1	0	4
理物	3	3	3	3			4	4	4	4	0	0

		2	2	2			2	3	3	3	3	4
			1	1			1	1	2	2	2	2
				0			0	0	0	1	1	1
缺页	+	+	+	+			+	+	+	+	+	+

缺页 10 次。

当进程分配的物理块数量增加的时候，期望发生缺页的次数应该减少，但是从上例中可以看到不但没有降低，反而有可能增大。这种异常现象称为 Belady 异常 (Belady's Anomaly)。

24. 在 5 个页框上使用 LRU 页面替换算法，当页框初始为空时，引用序列为 0 1 7 8 6 2 3 7 2 9 8 1 2 0，计算置换次数和缺页率？

解析：内存中驻留 4 个页框：

访问页面	0	1	7	8	6	1	3	8	2	9	8	1	2	0
页框 1	0	0	0	0	6		6		2	2		2		2
页框 2		1	1	1	1		1		1	9		9		0
页框 3			7	7	7		3		3	3		1		1
页框 4				8	8		8		8	8		8		8
是否缺页*	*	*	*	*	+		+		+	+		+		+

缺页 10 次，置换 6 次，缺页率 10/14。

25. 某系统采用二级页表的页式存储管理方式，页面大小为 2KB，页表项 4 字节，逻辑地址结构为：

| 页目录号 | 页号 | 页内偏移量 |

逻辑地址空间大小为 2^{31} 字节，则页目录表中包含表项的个数至少是多少项？

答：每页可以容纳 $2^{11}/4=2^9$ 个表项，所以表示逻辑空间需要 $2^{31}/2^{11}/2^9=2^{11}=2048$ 个页目录项。

26. 某分页管理系统主存为 128KB，划分为 64 块，若某程序占用了 8 页，0、1、2、3 页装入了 9、2、4、6 块中。回答如下问题：

(1) 该程序有多大？

(2) 逻辑地址 01A5CH 的物理地址是什么？

答：

① 该程序由 4 个页面，每个页面大小为 $128\text{KB}/64=2\text{KB}$ ，所以该程序为 $2\text{KB} \times 4=8\text{KB}$ 。

② 逻辑地址的页内地址为 11 位，所以 01A5CH 的页号为 3，查页表可知物理块号为 6，所以物理地址为 0325CH。

27. 某分页系统将页表存储在内存中，请回答如下问题：

(1) 如果一次访存的周期为 150ns，那么进行一次读取数据的内存访问时间是多少？

(2) 如果配置了 TLB 快表，命中率为 90%，访问快表的时间可以忽略不计，那么读取一次数据的内存访问时间是多少？

(3) 若希望访问内存的性能损失不超过 **10%**，则 **TLB** 的命中率不低于多少？

答：

(1) 分页系统访问数据需要进行地址映射，而在地址映射过程中需要访问存储在内存中的页表，所以访问一次数据实际要进行两次访存，因此访问时间是 **300ns**。

(2) 引入 **TLB** 后，有效访问时间= $90\% \times (\text{TLB 时间} + \text{一次访存时间}) + 10\% \times (\text{TLB 时间} + \text{两次访存时间})$ ，由于快表 **TLB** 的访问时间可以忽略不计，所以有效访问时间= $90\% \times \text{一次访存时间} + 10\% \times \text{两次访存时间} = 165\text{ns}$ 。

(3) 若性能损失不超过 **10%**，则有效访存时间 $\leq \text{一次访存} \times 1.1$ ，根据 (2) 中的计算公式，可得 **TLB** 命中率不低于 **90%**。

28. 某分段管理系统中，有段表如下表所示，请计算以下指令中的逻辑地址对应的物理地址，并判断指令的执行情况。

段号	段长	基址	控制
0	200	1100	R
1	20	510	RW
2	80	1510	X
3	310	100	W

(1) **LOAD r1, [0, 80];**

(2) **STORE r1, [1, 30];**

(3) **LOAD r1, [3, 200];**

(4) **JMP [2, 75]**

答：(1) 0 号段，段长 $200 > 80$ ，物理地址为 $1100 + 80 = 1180$ 。Load 指令需要读内存，因此 R 的权限合适，该指令可以执行成功。

(2) 1 号段，段长 $20 < 30$ ，段内地址越界，产生越界中断，指令中止执行。

(3) 3 号段，段长 $310 > 200$ ，物理地址为 $100 + 200 = 300$ 。Load 指令需要 R 权限，W 的权限不合适，所以该指令执行不成功。

(4) 2 号段，段长 $80 > 75$ ，物理地址为 $1510 + 75 = 1585$ 。JMP 指令需要跳转的位置也必须是一条指令，需要 X 权限，因此该指令可以执行成功。

29. 什么是系统抖动？发生系统抖动怎么办？

答：当系统频繁进行调页而导致系统时间大部分都消耗在调页而无法正常工作的时候，称为系统发生了抖动。如果系统已经发生了抖动，则应该减少并发进程的数量，缓解内存的紧张程度和降低调页请求数量。为了避免发生系统抖动，可以采用局部置换策略，或者在 CPU 调度算法中引入工作集算法。

30. 某虚拟分页存储系统，用于页面交换和调入的磁盘平均访问及传输时间为 **20ms**，页表保存在主存，访问时间为 **100ns**，为了提高访存性能设置了快表 **TLB**，假设快表的命中率为 **80%**，访问快表的时间忽略不计。当快表未命中时，有 **10%** 的几率缺页，请计算有效访存时间。

答：有效访问时间= $80\% \times (\text{TLB 时间} + \text{一次访存时间}) + 20\% \times (90\% \times (\text{TLB 时间} + 2$

次访存时间)+10%×(TLB 时间+调页时间+2 次访存时间))

依题意, TLB 时间忽略不计, 所以有效访问时间=80%×一次访存时间+20%×(90%×2 次访存时间+10%×(调页时间+2 次访存时间))=80%×100ns+20%×(90%×200ns+10%×(20ms+200ns))=0.4ms

说明: 由于调页花费的时间为 ms 级, 而其他时间都是 ns 级, 相差 10^6 数量级, 因此 ns 级的数据在上述计算中的加法中可以忽略不计。

31. 在某请求分页存储管理系统中, 一个作业的页面走向为 5、2、1、3、2、5、2、6、5、4、3、2, 当分配的物理块数分别是 3 和 4 时, 试分别计算采用 OPT、FIFO 和 LRU 算法的缺页次数和缺页率(假设初始没有页面驻留内存); 并从实现的难易程度及置换性能方面分析上述三种页面置换算法的优缺点。

答: (1) 物理块数=3, OPT 法

	5	2	1	3	2	5	2	6	5	4	3	2
物理块 1	5	5	5	5				5		5		5
物理块 2		2	2	2				6		4		4
物理块 3			1	3				3		3		2
	+	+	+	+				+		+	+	+

说明: 在本解答张, 当有两个页面以后都不再使用的时候, 按照 FIFO 进行淘汰的。也可以随意选择其中之一淘汰。

缺页次数=8, 缺页率=8/12。

(2) 物理块数=4, OPT 法

	5	2	1	3	2	5	2	6	5	4	3	2
物理块 1	5	5	5	5				5		4		
物理块 2		2	2	2				2		2		
物理块 3			1	1				6		6		
物理块 4				3				3		3		
	+	+	+	+				+		+		

缺页次数=6, 缺页率=6/12。

(3) 物理块数=3, FIFO 法

	5	2	1	3	2	5	2	6	5	4	3	2
物理块 1	5	5	5	3		3	3	6		6	6	2
物理块 2		2	2	2		5	5	5		4	4	4
物理块 3			1	1		1	2	2		2	3	3
	+	+	+	+		+	+	+		+	+	+

缺页次数=10，缺页率=10/12。

(4) 物理块数=4，FIFO 法

	5	2	1	3	2	5	2	6	5	4	3	2
物理块 1	5	5	5	5				6	6	6		6
物理块 2		2	2	2				2	5	5		5
物理块 3			1	1				1	1	4		4
物理块 4				3				3	3	3		2
	+	+	+	+				+	+	+		+

缺页次数=8，缺页率=8/12。

(5) 物理块数=3，LRU 法

	5	2	1	3	2	5	2	6	5	4	3	2
物理块 1	5	5	5	3	3	3	3	6	6	6	3	3
物理块 2		2	2	2	2	2	2	2	2	4	4	4
物理块 3			1	1	1	5	5	5	5	5	5	2
	+	+	+	+		+		+		+	+	+

缺页次数=9，缺页率=9/12。

(6) 物理块数=4，LRU 法

	5	2	1	3	2	5	2	6	5	4	3	2
物理块 1	5	5	5	5				5		5	5	5
物理块 2		2	2	2				2		2	3	3
物理块 3			1	1				6		6	6	2
物理块 4				3				3		4	4	4
	+	+	+	+				+		+	+	+

缺页次数=8，缺页率=8/12。

OPT 算法的缺页率最低，但是因为页面走向实际无法预知，因此是理论算法，无法实现。FIFO 算法实现最简单方便，每次置换的开销小，但是存在 Belady 异常。LRU 算法实现困难，需要较多的额外辅助开销。

32. 针对某虚拟存储系统，进行了 CPU 利用率和页面交换磁盘的利用率的检测，发现有四种情况：

- (1) CPU 利用率低，磁盘利用率高；
- (2) CPU 利用率高，磁盘利用率低；
- (3) CPU 利用率低，磁盘利用率低；
- (4) CPU 利用率高，磁盘利用率高。

请你分析一下，系统在这些不同情况时发生了什么事情？当 **CPU** 利用率低的时候，如果增加进程并发数能提高 **CPU** 利用率吗？虚拟存储器是否起到作用了？

答：（1）可能发生了抖动。（2）在系统中计算密集型进程为主。（3）系统中进程较少，或者进程大部分处于阻塞状态。（4）系统可能处于非正常状态，进程异常大量占用 **CPU** 和内存。

如果系统发生了抖动，此时 **CPU** 利用率低，但是如果继续增加进程并发数，则适得其反，会加剧系统的抖动。这可能是页面置换策略不当引起的，所以虚拟存储器此时无法解决。

33. 某请求分页系统采用局部置换策略，已知为某进程分配了四个页框，页面的使用情况如下表所示：

页号	装入时间	上次引用时间	A	M
0	130	248	0	0
1	210	230	1	0
2	100	244	1	1
3	180	260	1	1

如果某时刻进程要访问 **5** 号页面，请基于 **FIFO**、**LRU**、简单 **CLOCK** 和改进型 **CLOCK** 置换算法确定需要置换的页面。

答：5 号页面不在内存，发生缺页中断，调页过程中需要置换。由于采用局部置换策略，因此需要从目前进程的四个已装入内存的页面中淘汰一个页面。

FIFO：根据装入时间可知 2 号页面最先装入，因此选择 2 号页面置换；

LRU：根据上次引用时间可知 1 号页面是最近最早引用页面，因此选择 1 号页面置换；

简单 **CLOCK**：选择访问位为 0 的 0 号页面置换；

改进型 **CLOCK**：选择 A 和 M 位为 00 的 0 号页面置换。

34. 某计算机系统按字节编址，逻辑地址结构如下所示：

页目录号（10 位）	页表索引（10 位）	页内偏移量（12 位）
------------	------------	-------------

请回答下列问题：

（1）该系统采用的是几级页表？逻辑地址空间是多大？

（2）若假设页目录项和页表项都是 **4** 字节，则页目录需要占多少页？进程页表最多需占多少页面？

答：

（1）该系统采用的是二级页表。逻辑地址空间为 $2^{32}=4\text{GB}$ 。

（2）页目录项数量= $2^{10}=1024$ 项，页面大小= $2^{12}=4\text{KB}$ ，每页可以存放的也目录项数量= $4\text{KB}/4\text{B}=1024$ 项，因此页目录需要占 1 页。

进程页表最多页数= $2^{10} \times 2^{10}=2^{20}=1\text{M}$ 个。

35. **Linux** 系统中使用的四级分页系统，地址结构包括哪些部分？各部分的位数在 **32** 位和 **64** 位系统中如何变化？

答：**Linux** 系统中使用的四级分页地址结构包括：全局页目录 **PGD**、上级页目录 **PUD**、中间页目录 **PMD**、页表项索引 **PTE**、页内偏移量。

32 位系统中，地址总位数 **32** 位，实际使用两级页表，即 **PGD** 和 **PTE**，页内偏移量为 **12** 位，其他部分均为 **10** 位。

64 位系统中，地址总位数 48 位，使用四级页表，页内偏移量为 12 位，其他部分均为 9 位。

36. Linux 在管理物理内存页框时采用伙伴算法，**1)** 请先分析一下伙伴算法的优点和缺点。**2)** 描述以下页框的分配和回收过程。（其中系统中总共有 **2K** 个物理页框，一个进程第一次请求 **20** 个连续页框，第二次请求 **100** 个连续物理页框，接着另一个进程第一次请求 **50** 个连续物理页框，第二次请求 **200** 个连续物理页框。最后第一个进程执行结束。）

答：

(1) 伙伴算法优点：较好的解决外部碎片问题；可以满足分配连续内存的需要；适合大内存分配管理。缺点：合并的要求太过严格，只能是满足伙伴关系的块才能合并；因内部碎片存在浪费问题；若某连续内存块中仅仅一个页面被占用，整块内存区都不能合并；存在效率问题，频繁地合并又立即拆分的过程是无效率的。

(2) 2K 个物理页框开始是两虚的空闲空间，是伙伴系统中 **order=11** 的队列中的唯一节点，其他 **order (0-10)** 的队列都为空。

第一次请求：分配 20 个连续页框。伙伴系统会分配 (0-31) 32 个页框响应请求。第

二次请求：分配 100 个连续页框。伙伴系统会分配 (128-255) 128 个页框响应请求。第

三次请求：分配 50 个连续页框。伙伴系统会分配 (64-127) 64 个页框响应请求。第

四次请求：分配 200 个连续页框。伙伴系统会分配 (256-511) 256 个页框响应请求。第

五次请求：回收 20 个连续页框。回收 0-31 页框，与伙伴 32-63 合并。

第六次请求：回收 100 个连续页框。回收 128-255 页框，无伙伴可以合并。

37. (2009 年统考真题) 某请求分页管理系统中，假设进程的页表如右表所示。页面大小为 **4KB**，一次内存的访问时间为 **100 纳秒 (ns)**，一次快表 (**TLB**) 的访问时间是 **10ns**，处理一次缺页的平均时间为 **100 毫秒**（已含更新 **TLB** 和页表的时间），进程的驻留集大小固定为 **2** 个页框，采用 **FIFO** 法置换页面。假设 **1) TLB** 初始为空；**2)** 地址转换时，先访问 **TLB**，若 **TLB** 未命中时再访问页表（忽略 **TLB** 更新时间）；**3)** 有效位为 **0** 表示页面不在内存中。请问：

(1) 该系统中，一次访存的时间下限和上限各是多少？给出计算过程。

(2) 若已经先后访问过 **0、2** 号页面，则虚地址 **1565H** 的物理地址是多少？给出计算过程。

页号	页框号	有效位	装入时间
0	101H	1	2
1	—	0	—
2	254H	1	4

答：

(1) 访存的下限就是访存的最短时间 = 一次快表（命中）+ 一次内存访问（数据）
= $10\text{ns} + 100\text{ns} = 110\text{ns}$

访存的上限就是访存的最长时间 = 一次快表（不命中）+ 一次访存（页表）+ 一次缺页 + 一次快表（命中）+ 一次访存（数据）= $10\text{ns} + 100\text{ns} + 100\text{ms} + 10\text{ns} + 100\text{ns} = 100000.22\mu\text{s}$

(2) 页面大小 **4KB**，所以页内地址 12 位。1565H 的页号为 1，不在内存，缺页。采用

FIFO 法置换页面 0，所以 1 号页调入 101H 页框中，则 1565H 对应的物理地址为 101565H。

38. 某请求分页系统，用户空间为 **32KB**，每个页面 **1KB**，主存 **16KB**。某用户程序有 **7** 页长，某时刻该用户进程的页表如右表所示：

(1) 计算两个逻辑地址：**0AC5H**、**1AC5H** 对应的物理地址。

(2) 已知主存的一次存取为 **15ns**，对于 **TLB** 表（快表）的查询时间可以忽略，则访问上述两个逻辑地址共耗费多少时间？

页号	物理块号	是否在 TLB
0	8	是
1	7	是
2	4	否
3	10	否
4	5	否
5	3	是
6	2	是

答：

(1) 0AC5H 页号为 2，查页表得物理块号 4，所以物理地址为 12C5H

1AC5H 页号为 6，查页表得物理块号 2，所以物理地址为 0AC5H

(2) 访问 0AC5H 时，TLB 不命中，所以需要两次访存的时间 30ns

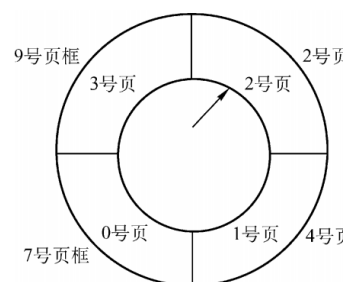
访问 1AC5H 时，TLB 命中，所以只需一次访存的时间 15ns

39. （2010 年统考题）设某计算机的逻辑地址空间和物理地址空间均为 **64KB**，按字节编址。若某进程最多需要 **6 页（Page）** 数据存储空间，页的大小为 **1KB**，操作系统采用固定分配局部置换策略为此进程分配 **4 个页框（Page Frame）**。在时刻 **260** 前的该进程访问情况见下表（问位即使用位）。

页号	页框号	装入时刻	访问位
0	7	130	1
1	4	230	1
2	2	200	1
3	9	160	1

当该进程执行到时刻 260 时，要访问逻辑地址为 17CAH 的数据。请回答下列问题：

- (1) 该逻辑地址对应的页号是多少？
- (2) 若采用先进先出 (FIFO) 置换算法，该逻辑地址对应的物理地址是多少？要求给出计算过程。
- (3) 若采用时钟 (CLOCK) 置换算法，该逻辑地址对应的物理地址是多少？要求给出计算过程（设搜索下一页的指针沿顺时针方向移动，且当前指向 2 号页框，示意图如题 39 图所示）。



题 39 图

答：

(1) 17ACH 转换成 2 进制：0001 0111 1010 1100 B（这里的 B 表示二进制）

页面大小 1KB，所以页内地址 10 位，所以地址结构为：0001 01 | 11 1010 1100

所以页号为：101B，即逻辑地址的页号为 5。

(2) 查页表可知 5 号页面不在内存中，产生缺页中断。根据 FIFO 置换，0 号页的装入时刻最早，所以换出 0 号页，换入 5 号页到 7 号页框。即用 111B 替换 101B 的页号部分，就可以得到物理地址：0001 1111 1010 1100B，即 1FACH。

(3) CLOCK 算法增加一个访问位，根据访问位的值确定换出（为 0）还是保留（为 1）。对于访问位为 1 的页面虽然保留，但是访问位置 0。然后检查下一个页面，直到检查到一个访问位为 0 的页面，才将它换出。

本题目中，页表中的所有页面访问位都是 1，所以，第一轮检查，每个页面都不会被换出，但是访问位都会变成 0。检查指针顺时针旋转一圈后，又指向 2 号页框，此时它的访问位已经是 0 了，所以将 2 号页框的 2 号页换出，5 号页面换入到 2 号页框中。替换页号 101B 即可得到物理地址：0000 1011 1010 1100B，即 0BACH。

40. （2013 年统考题）某计算机主存按字节编址，逻辑地址和物理地址都是 32 位，页表项大小为 4 字节。请回答下列问题。

- (1) 若使用一级页表的分页存储管理方式，逻辑地址结构为：

页号（20 位）	页内偏移量（12 位）
----------	-------------

则页的大小是多少字节？页表最大占用多少字节？

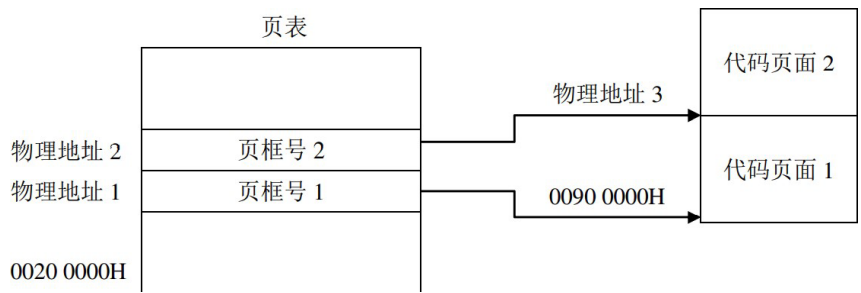
- (2) 若使用二级页表的分页存储管理方式，逻辑地址结构为：

页目录号（10 位）	页表索引（10 位）	页内偏移量（12 位）
------------	------------	-------------

设逻辑地址为 LA，请分别给出其对应的页目录号和页表索引的表达式。

(3) 采用 (1) 中的分页存储管理方式，一个代码段起始逻辑地址为 0000 8000H，其长度为 8 KB，被装载到从物理地址 0090 0000H 开始的连续主存空间中。页表从主存 0020

0000H 开始的物理地址处连续存放，如下图所示（地址大小自下向上递增）。请计算出该代码段对应的两个页表项的物理地址、这两个页表项中的页框号以及代码页面 2 的起始物理地址。



题 40 图

答：

(1) 页的大小 $= 2^{12}B = 4KB$ ，页表项个数 $= 2^{20}$ ，每个页表项占 4B，所以页表占用空间 $= (2^{20}) * 4B = 4MB$ 。

(2) 这个答案不唯一，根据原理实现即可。

页目录号可表示为： $((\text{unsigned int})(LA)) \gg 22 \& 0x3FF$ 。（先把 LA 右移 22 位，再跟 0x3FF 与操作，取低 10 位）

页表索引可表示为： $((\text{unsigned int})(LA)) \gg 12 \& 0x3FF$ 。（先把 LA 右移 12 位，再跟 0x3FF 与操作，取低 10 位）

(3) 代码段长度 $8KB = 2$ 个页，逻辑地址 0000 8000H，根据分页地址结构，后 12 位是页内地址，所以代码段对应的两个页号为 00008H, 00009H。

页表起始地址是 0020 0000H，每个表项 4B，则 8 号页和 9 号页的物理地址：

8 号页表项物理地址 $= 0020\ 0000H + 8 * 4 = 0020\ 0020H$

9 号页表项物理地址 $= 0020\ 0000H + 9 * 4 = 0020\ 0024H$

页框大小=页面大小，物理地址结构与逻辑地址结构相同，所以页框号是前 20 位，已知代码页 1 的物理地址为 0090 0000H，则页框号 1 = 00900H，因为代码页连续存放，

则页框号 2 = 页框号 1 + 1 = 00901H。

物理地址 3 指向的是代码页面 2 的起始位置，所以页内偏移量是 0，即物理地址 3 = 0090 1000H

41. （2015 年统考题）某计算机系统按字节编址，采用二级页表的分页存储管理方式，虚拟地址格式如下所示：

10 位	10 位	12 位
页目录号	页表索引	页内偏移量

请回答下列问题。

- (1) 页和页框的大小各为多少字节?进程的虚拟地址空间大小为多少页?
- (2) 假定页目录项和页表项均占 4 个字节, 则进程的页目录和页表共占多少页?要求写出计算过程。
- (3) 若某指令周期内访问的虚拟地址为 0100 0000H 和 0111 2048H, 则进行地址转换时共访问多少个二级页表? 要求说明理由。

答:

- (1) 页大小=页框大小= $2^{12}\text{B}=4\text{KB}$, 32 位逻辑地址空间大小为 $2^{32}\text{B}=4\text{GB}$ 。
- (2) 页目录占 1 页面, 页表占 2^{10} 个页面, 所以共占 1025 个页面。
- (3) 0100 0000H 的页目录号 0000 0001 00B, 页表索引 0H;
0111 2048H 的页目录号 0000 0001 00B, 页索引号 112H。
因为页目录号一样, 所以共访问 1 个二级也表。