

Operating System Quiz 2

Name:

Student ID:

1. Consider a system that uses single-level paging for virtual memory with 32 bit physical and virtual addresses. Suppose page size 64KB (216 bytes).

- (a) How many pages of virtual memory are there?
- (b) How many frames of physical memory are there?
- (c) How many bits are needed for the page offset?
- (d) How many bits are needed for the page number?
- (e) A process uses a contiguous 220 bytes of memory for its address space. How many valid entries will the page table have?

第一题就是 4GB / 64KB

第三题 2 的 16 次方= 64KB

第五题 2 的 20/2 的 16 = 2 的 4 次

2¹⁶, 2¹⁶, 16, 16, 2⁴

2. Suppose the FrameSize is 2048 and a process's page table is:

Page Number	Frame Number	Valid
[0]	4	T
[1]	2	T
[2]	-	F
[3]	3	T

Give the physical address for each of the following logical addresses or write "pagefault" if the reference would cause a page fault:

- (a) 56, (b) 5000, (c) 6500, (d) 1024

8248, page fault, 6500, 9216

4*2048+56=8248

4*2048+1024=9216

3. Suppose that there are only 3 frames of physical memory (initially empty), and a process accesses its page in the following order: 1,2,3,4,2,1,5,6,2,1,2,3,7,6,3,2,1,2,3,6. Show which pages are in each frames as the pages are reference and give the total number of page faults using each of the following page replacement algorithms:
- (a) FIFO

- (b) LRU
- (c) Optimal
- (d) Second Chance

16 算错了, 15 对, 11 算对了, 16

4. Assume we have a demand-paged memory. The page table is held in registers. It takes 8 milliseconds to service a page fault if an empty page is available or the replaced page is not modified, and 20 milliseconds if the replaced page is modified. Memory access time is 100 nanoseconds. Assume that the page to be replaced is modified 70 percent of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds?

$$0.2 \mu\text{sec} = (1 - P) \times 0.1 \mu\text{sec} + (0.3P) \times 8 \text{ millisec} + (0.7P) \times 20 \text{ millisec}$$

$$0.1 = -0.1P + 2400 P + 14000 P$$

$$0.1 \approx 16,400 P$$

$$P \approx 0.000006 \text{ 算对了}$$

5. Given five memory partitions of 100 KB, 500 KB, 200 KB, 300 KB, and 600 KB (in order), how would each of the first-fit, best-fit, and worst-fit algorithms place processes of 212 KB, 417 KB, 112 KB, and 426 KB (in order)? Which algorithm makes the most efficient use of memory?

First fit : 212->500, 417->600, 112->200, 426->fault

Best fit: 212->300, 417->500, 112->200, 426->600

Worst fit: 212->600, 417->500, 112->300, 426->fault

Best fit

6. 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。某进程最多需要 6 页数据存储空间，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。

页号	页框号	装入时间	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（16 进制）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？

17CAH=0001 0111 1100 1010， 页大小 1KB, 等于后 10 位表示页内，前 6 个表示页号码，所以页号=0001 01（这是一个关键技巧！），10 进制就是 5，所以页号=5

采用 FIFO，需要被置换出去的是 7，对应的二进制是 0001 1111 1100 1010，16 进制为 1FCAH，前 6 位变成 0001 11（7），后面 10 位不变

7. 请求分页管理系统中，假设某进程的页表内容如下表所示。页面大小为 4KB，一次内存的访问时间是 100ns，一次快表（TLB）的访问时间是 10ns，处理一次缺页的平均时间为 108ns（已含更新 TLB 和页表的时间），进程的驻留集大小固定为 2，采用最近最少使用置换算法（LRU）和局部淘汰策略。假设：（1）、TLB 初始为空；（2）、地址转换时先访问 TLB，若 TLB 未命中，再访问页表（忽略访问页表之后的 TLB 更新时间）；

（3）、有效位为 0 表示页面不在内存，产生缺页中断，缺页中断处理后，返回到产生缺页中断的指令处重新执行

设有虚地址访问序列：2362H、1565H、25A5H，请问：（1）、依次访问上述三个地址，各需多少时间？给出计算过程。（2）、基于上述访问序列，虚地址 1565H 的物理地址是多少？请说明理由。

页号	页框号	有效位（存在位）
0	101H	1
1	--	0
2	154H	1

1)由于页面大小为 4KB=2¹²KB，故逻辑地址末尾 12 位表示页内偏移 W，前 4 位表示页号 P。设访问内存时间为 t，查找快表时间为 a，处理缺页中断的时间为 T。

对于逻辑地址 2362H，其表示的页号为 2，访问次序依次为：访问 TLB 未命中(10ns)；访问页表命中 2 号页面，并将页表项副本放入 TLB（100ns），得到物理地址，访问内存（100ns）。

故逻辑地址 2362H 的访问时间为：EAT=a+t+t=10+100+100=210ns；

对于逻辑地址 1565H，其表示的页号为 1，访问次序依次为，访问 TLB 未命中(10ns)；访问页表未命中(100ns)；发生缺页中断(108ns)；访问 TLB 命中 1 号页面(10ns)；得到物理地址访问内存(100ns)。此时驻留集已满(0 号页面和 2 号页面)

故逻辑地址 1565H 的访问时间为：
 $EAT=a+t+T+a+t=T+2\times(a+t)=108+2\times(100+10)=100000220ns。$

对于逻辑地址 25A5H，其表示的页号为 2，访问次序依次为：访问 TLB 命中二号页面 (10ns)；得到物理地址访问内存(100ns)。

故逻辑地址 25A5H 的访问时间为: $EAT=a+t=10+100=110ns;$

2)访问逻辑地址 1565H 时由于驻留集已满(0 号页面和 2 号页面)。故应从页表中淘汰一个页面，根据 LRU 算法，2 号页面刚被使用过，故淘汰 0 号页面，将 1 号页面调入获得内存块号 101H。

则地址由内存块号和页内偏移量拼接得到物理地址为：101565H。

8. 某请求分页系统的局部页面置换策略如下：系统从 0 时刻开始扫描，每隔 5 个时间单位扫描一轮驻留集（扫描时间忽略不计），本轮没有被访问过的页框将被系统回收，并放入到空闲页框链尾，其中内容在下次被分配之前不被清空。当发生缺页时，如果该页曾被使用过且还在空闲页框链表中，则重新放回进程的驻留集中；否则，从空闲页框链表头部取出一个页框。假设不考虑其它进程的影响和系统开销，初始时进程驻留集为空。目前系统空闲页框链表中页框号依次为 32、15、21、41。进程 P 依次访问的<虚拟页号，访问时刻>是：<1, 1>、<3, 2>、<0, 4>、<0, 6>、<1, 11>、<0, 13>、<2, 14>。请回答下列问题。

- (1) 访问 <0, 4> 时，对应的页框号是什么？说明理由。
- (2) 访问 <1, 11> 时，对应的页框号是什么？说明理由。
- (3) 访问 <2, 14> 时，对应的页框号是什么？说明理由。
- (4) 该策略是否适合于时间局部性好的程序？说明理由。

虚拟页	<1,1>	<3,2>	<0,4>	5.扫描	<0,6>	10.扫描	<1,11>	<0,13>	<2,14>
进程驻留集	1->32	1->32	1->32	1->32	1->32	0->21	0->21	0->21	0->21
		3->15	3->15	3->15	3->15		1->32	1->32	1->32
			0->21	0->21	0->21				2->41
缺页	√	√	√						√
页框链表指针	32	15	21	41	41	41	41	41	41
空闲页框链表						1->32	3->15	3->15	3->15
						3->15			

参考上表

1) 虚拟页为<0,4>时, 对应的页框号为 21。写对了

<0,4>为第三个访问, 此时页匡量表指针指向 21 号内存块。

2) 虚拟页为<1,11>时, 对应的页框号为 32。因为曾被使用过, 写对了

<1,11>为第五个访问, 此时已经扫描两次(5 时刻、10 时刻); 5 时刻扫描时, 1、3、0 号虚拟页号在 1-5 时刻内均被访问过, 故仍在驻留集。10 时刻扫描时, 0 号虚拟页在 6-10 时刻内被访问, 1、3 号虚拟页未被访问, 故将 1、3 号虚拟页加入空闲页链表中。在访问<1,11>时, 发现 1 号虚拟页在空闲页框链表中, 故将其取出重新放回进程驻留集中。

故其对应的内存块号仍为 32。

3) 虚拟页为<2,14>时, 对应的页框号为 41。我写对了

<2,14>为第七个访问, 此时发生缺页, 故从空闲页框链表表头(即页框链表指针所指的空闲页框号)取出空闲页框号为 41 的空闲页框。

4) 该方法适用于时间局部性好的程序, 因为时间局部性越好, 在有限的进程驻留集下重复从空闲页框链表中取回页表项到驻留集的几率就越大。

9. 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?

大小= $2^{32}/4K = 2^{20}$, $2^{20} \times 4\text{Byte} = 2^{22} \text{ B} = 4\text{MB}$

(2)若使用二级页表的分存储管理方式, 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。

$0x3FF = 000000001111111111$ (只取最后 8 位)

Page Dir: $(LA \gg 22) \& 0x3FF$

Page No: (LA>>12) & 0x3FF

(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 8KB，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物理地址、这两个页表项中的框号以及代码页面 2 的起始物理地址。



物理地址 1 = 0020 0020H

物理地址 2 = 0020 0024H

物理地址 3 = 0090 1000H

10. A system with 32-bit addresses, 1 GB (2^{30}) main memory, and a 1 megabyte (2^{20} -bit) page size will have a page table that contains:
A. 4K (2^{12}) entries B. 4M (2^{20}) entries C. 1M (2^{20}) entries D. 1K (2^{10}) entries
11. Consider a system, which has a page size of 2KB. If we load a process of size 8000 bytes, what amount of memory would be unused?
A. 0 bytes B. 144 bytes C. 192bytes D. 1856 bytes