

Bài Tập Memory Management

1. Quản lý bộ nhớ

- Tại sao có sự phân biệt giữa paging và demand paging ?
- Điều gì tốt cho demand paging
- Tên 1 bộ cấp phát trang
- Khi nào lỗi trang xảy ra
- Điều gì xảy ra khi lỗi trang xảy ra
- Thuật giải thay thế trang làm gì

2. Answer following questions:

- Define **external** and **internal** fragments. Consider the following memory management schemes: fixed-size partitions, variable-size partitions, and paging. Which schemes have external fragments, and which schemes have internal fragments? Why?
- Given memory holes (**i.e.**, unused memory blocks) of 100K, 500K, 200K, 300K and 600K (in address order) as shown below, how would each of the **first-fit**, **next-fit**, **best-fit** and **worst-fit** algorithms allocate memory requests for 290K, 420K, 110K and 350K (in this order)? The shaded areas are used/allocated regions and are not available. Write your answer into the following diagrams. You should clearly write down the size of each memory block and indicate its status

3. Giả sử trong quá trình quản lý bộ nhớ ảo dạng phân đoạn, HĐH duy trì Segment Table:

Segment	Base	Limit
0	300	700
1	1200	500
2	2000	600

Hãy tính địa chỉ vật lý cho mỗi địa chỉ logic sau: (1,200), (1,0), (0,700), (2,0), (2,600)

4. Giả sử có bảng đoạn sau:

Segment	Base	Limit
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

Tính địa chỉ vật lý cho mỗi địa chỉ logic sau:

- a. (0, 430) b. (1,010) c. (2,500) d. (3,400) e. (4,112)

5. Vùng địa chỉ logic (vùng nhớ ảo) của 1 tiến trình được chia thành các trang A, B, C, D, E được ánh xạ sang vùng địa chỉ vật lý theo bảng sau:

STT	Trang logic	Khung trang số
1	A	3
2	B	
3	C	5
4	D	1
5	E	

Hãy minh họa bằng hình vẽ các thành phần sau:

- Vùng địa chỉ logic

- Bảng trang (sử dụng bit “true – false”)
 - Vùng địa chỉ vật lý
 - Nội dung đĩa cứng lưu các trang
6. Dung lượng vùng nhớ là 3000K. Địa chỉ bắt đầu cấp phát là 0K. Cấp phát bộ nhớ bằng kỹ thuật phân vùng động với giải thuật Worst-Fit. Xây dựng danh sách quản lý cấp phát tại từng thời điểm tương ứng với chuỗi cấp phát sau :
- A(300K)→ B(400K)→ C(550K)→ thu hồi B→ D(400K)→ thu hồi A→ E(300K)→ F(350K)→ thu hồi D→ G(50K)
7. A paging system uses 16-bit address and 4K pages. The following shows the page tables of two running processes, **Process 1** and **Process 2**. Translate the logical addresses in the table below to their corresponding physical addresses, and fill the table entries with your answers.

Process 1		Process 2		Process	Address	Page #	Offset	Physical Address
0	3	0	2	Process 1	11,034			
1	7	1	0	Process 2	12,345			
2	1	2	6					
3	5	3	4					

8. Một máy tính cung cấp cho mỗi tiến trình 65536 byte của không gian địa chỉ được chia thành các trang 4096 byte. Một chương trình có kích thước text 32768 byte, dữ liệu kích thước 16386 byte và stack kích thước 15870 byte. Hỏi chương trình này có nằm vừa vặn trong không gian địa chỉ không ? nếu kích thước trang là 512 byte, thì sao ? lưu ý rằng 1 trang không cùng chứa phần của 2 segment khác nhau.
9. Với mỗi địa chỉ ảo dạng thập phân sau, tính số trang ảo và offset cho trang 4K và trang 8K : 20000, 32768, 60000
10. Cho 5 partition bộ nhớ có kích thước lần lượt là 100KB, 500KB, 200KB, 300KB, 600KB. Làm thế nào mỗi thuật giải first fit, best fit, worst fit đặt các tiến trình 212KB, 417KB, 112KB, 426KB. Thuật giải nào sử dụng bộ nhớ hiệu quả nhất.
11. A CPU generates 32-bit virtual addresses. The page size is 4 KB. The processor has a translation look-aside buffer (TLB) which can hold a total of 128 page table entries and is 4-way set associative. The minimum size of the TLB tag is ?
12. Virtual memory uses a page table to track the mapping of virtual addresses to physical addresses. To speed up this translation, modern processors implement a cache of the most recently used translations, called the translation-lookaside buffer (TLB). This exercise shows how the page table and TLB must be updated as addresses are accessed. The following list is a stream of virtual addresses as seen on a system. Assume 4-KB pages and a four entry fully associative TLB with LRU replacement policy. If pages must be brought in from disk, give them the next largest unused page number (i.e., starting at 13).

0x0FFF	Valid	Tag	PP #	LRU
0x7A28	1	11	12	2
0x3DAD	1	7	4	3
0x3A98	1	3	6	4
0x1C19	0	4	9	1

Initial TLB: (Note: Values are in base-10)

The LRU column works as follows: The older an entry is, the lower its LRU number will be. Each time an entry is used, its LRU number is set to 4 (since it is now the most-recently used), and the other numbers are adjusted downward accordingly.

Initial Page Table: (Note: Values are in base-10)

Index	Valid	Physical Page or On Disk
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
10	1	3
11	1	12