EECE.4810/EECE.5730: Operating Systems

Spring 2019

Key Questions

Memory Management (Lec. 24-30)

QUESTIONS

1. Describe the three key address space-related abstractions an operating system supports.
2. What is address binding? What are the points at which addresses are bound and the implications of each?
3. Explain the basics of dynamic address translation.
4. Describe base and bounds address translation.
5. What is fragmentation?
6. Describe the different storage allocation schemes used (first fit, best fit, worst fit).
7. Explain how segmentation can be used to manage address spaces.
8. What information is included in a typical segment table?
9. Explain the basics of memory management through paging.
10. Describe the organization of a multilevel page table.
11. Describe the organization of a hashed page table.
12. Describe the organization of an inverted page table.
13. Describe the contents of a page table entry.
14. How is a page chosen to evict when necessary?
15. What is the purpose of the dirty bit in a page table entry?

EXAMPLES

1. Given the following lists of holes and address space requests:
   - Holes: 300 KB, 600 KB, 350 KB, 200 KB, 750 KB, 125 KB
   - Address spaces: 115 KB, 500 KB, 358 KB, 200 KB, 375 KB

Assuming you're using base and bounds translation, how would these processes be placed using (a) first-fit, (b) best-fit, and (c) worst-fit allocation?

2. Given the segment table below:

Segment #	V	Base	Bounds	Access
0	1	219	600	read/write
1	1	2300	14	read/write
2	1	90	100	read/exec
3	1	1327	580	read/write
4	0	1952	96	read

What is the physical address corresponding to each virtual address below? Virtual addresses are specified as a pair of values: the segment number and offset.

a. 0, 430

b. 1, 10

c. 2, 500

d. 3, 400

e. 4, 112

3. Consider a logical address space of 256 pages with 4 KB page size, mapped onto a physical memory of 64 frames

a. How many bits are in the virtual address?

b. How many bits are in the physical address?

c. What's the total size of each address space (virtual and physical)?

4. Given a system using 32-bit virtual addresses, a 4 KB page size, and 4 bytes in each page table entry, what's the size of the page table?

5. Assume the current process uses the page table below:

Virtual page #	Valid bit	Reference bit	Dirty bit	Frame #
0	1	1	0	4
1	1	1	1	7
2	0	0	0	--
3	1	0	0	2
4	0	0	0	--
5	1	0	1	0

a. Which virtual pages are resident in physical memory?

b. Which resident pages are candidates for eviction?

c. Assuming 1 KB pages and 16-bit addresses, what physical addresses would the virtual addresses below map to?

i. 0x041C

ii. 0x08AD

iii. 0x157B