Xi XIA
2016213482
2016215117

8.11 Given six memory partitions of 300 KB, 600 KB, 350 KB, 200 KB, 750 KB, and 125 KB (in order), how would the first-fit, best-fit, and worst-fit algorithms place processes of size 115 KB, 500 KB, 358 KB, 200 KB, and 375 KB (in order)? Rank the algorithms in terms of how efficiently they use memory.

Answer:

1. first-fit:

115 -> 300
500-> 600
358 -> 750
200 -> 350
375 -> not able to allocate

2. Best-fit:

115 -> 125
500 -> 600
358 -> 750
200 -> 200
375 -> not able to allocate

3. Worst-fit:

115 -> 750
500 -> 600
358 -> not able to allocate
200 -> 350
375 -> not able to allocate

efficiency: best-fit > first-fit > worst-fit

8.25 Consider a paging system with the page table stored in memory.

a. If a memory reference takes 50 nanoseconds, how long does a paged memory reference take?

Answer:

100 ns: 50 ns to access page table, 50 ns to access word in memory

b. If we add TLBs, and 75 percent of all page-table references are found in the TLBs, what is the effective memory reference time? (Assume that finding a page-table entry in the TLBs takes 2 nanoseconds, if the entry is present.)

Answer:

EAT = 0.75*50 + 0.25*100 + 2 = 64.5 ns

8.26 Why are segmentation and paging sometimes combined into one scheme?

Answer:

Xi XIA
2016213482
2016215117

Segmentation and paging are often combined in order to improve upon each other. Segmented paging is helpful when the page table becomes very large. A large contiguous section of the page table that is unused can be collapsed into a single-segment table entry with a page-table address of zero. Paged segmentation handles the case of having very long segments that require a lot of time for allocation. By paging the segments, we reduce wasted memory due to external fragmentation as well as simplify the allocation.

## 8.28 Consider the following segment table:

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

What are the physical addresses for the following logical addresses?

a. 0,430
b. 1,10
c. 2,500
d. 3,400
e. 4,112

Answer:

a. 219+430=649
b. 2300+10=2310
c. illegal reference, trap to operating system
d. 1327 + 400 = 1727
e. illegal reference, trap to operating system