

Shefali Emmanuel

CSCI 340 Operating Systems Memory Allocation Homework

Assigned Tuesday Oct. 29, Due Friday Nov. 1st by 11:59 PM EST via OAKS Dropbox

1. Suppose that an operating system's list of available (free) blocks of memory contains blocks having the following sizes.

freeList $\rightarrow 10 \rightarrow 4 \rightarrow 20 \rightarrow 18 \rightarrow 7 \rightarrow 9 \rightarrow 12 \rightarrow 15$

a) (5 Pts) Suppose that we are using *first fit* allocation policy and we receive malloc requests of size 12, 10, and 9, in that order. Beginning with the above freeList, show the contents of the freeList after each malloc request. Indicate if a malloc request cannot be satisfied.

freeList $\rightarrow 10 \rightarrow 4 \rightarrow 20 \rightarrow 18 \rightarrow 7 \rightarrow 9 \rightarrow 12 \rightarrow 15$

malloc(12)

freeList $\rightarrow 10 \rightarrow 4 \rightarrow 8 \rightarrow 18 \rightarrow 7 \rightarrow 9 \rightarrow 12 \rightarrow 15$

malloc(10)

freeList $\rightarrow 4 \rightarrow 8 \rightarrow 18 \rightarrow 7 \rightarrow 9 \rightarrow 12 \rightarrow 15$

malloc(9)

freeList $\rightarrow 4 \rightarrow 8 \rightarrow 9 \rightarrow 7 \rightarrow 9 \rightarrow 12 \rightarrow 15$

b) (5 Pts) Suppose that we are using *next fit* allocation policy and we receive malloc requests of size 12, 10, and 9, in that order. Beginning with the above freeList, show the contents of the freeList after each malloc request. Indicate if a malloc request cannot be satisfied.

freeList $\rightarrow 10 \rightarrow 4 \rightarrow 20 \rightarrow 18 \rightarrow 7 \rightarrow 9 \rightarrow 12 \rightarrow 15$

malloc(12)

freeList $\rightarrow 10 \rightarrow 4 \rightarrow 8 \rightarrow 18 \rightarrow 7 \rightarrow 9 \rightarrow 12 \rightarrow 15$

malloc(10)

freeList $\rightarrow 10 \rightarrow 4 \rightarrow 8 \rightarrow 8 \rightarrow 7 \rightarrow 9 \rightarrow 12 \rightarrow 15$

malloc(9)

freeList $\rightarrow 10 \rightarrow 4 \rightarrow 8 \rightarrow 8 \rightarrow 7 \rightarrow 12 \rightarrow 15$

c) (5 Pts) Suppose that we are using *best fit* allocation policy and we receive malloc requests of size 12, 10, and 9, in that order. Beginning with the above freeList, show the contents of the freeList after each malloc request. Indicate if a malloc request cannot be satisfied.

freeList $\rightarrow 10 \rightarrow 4 \rightarrow 20 \rightarrow 18 \rightarrow 7 \rightarrow 9 \rightarrow 12 \rightarrow 15$

malloc(12)

freeList $\rightarrow 10 \rightarrow 4 \rightarrow 20 \rightarrow 18 \rightarrow 7 \rightarrow 9 \rightarrow 15$

malloc(10)

freeList $\rightarrow 4 \rightarrow 20 \rightarrow 18 \rightarrow 7 \rightarrow 9 \rightarrow 15$

malloc(9)

freeList $\rightarrow 4 \rightarrow 20 \rightarrow 18 \rightarrow 7 \rightarrow 15$

d) (5 Pts) Suppose that we are using *worst fit* allocation policy and we receive malloc requests of size 12, 10, and 9, in that order. Beginning with the above freeList, show the contents of the freeList after each malloc request. Indicate if a malloc request cannot be satisfied.

freeList $\rightarrow 10 \rightarrow 4 \rightarrow 20 \rightarrow 18 \rightarrow 7 \rightarrow 9 \rightarrow 12 \rightarrow 15$

malloc(12)

freeList $\rightarrow 10 \rightarrow 4 \rightarrow 8 \rightarrow 18 \rightarrow 7 \rightarrow 9 \rightarrow 12 \rightarrow 15$

malloc(10)

freeList $\rightarrow 10 \rightarrow 4 \rightarrow 8 \rightarrow 8 \rightarrow 7 \rightarrow 9 \rightarrow 12 \rightarrow 15$

malloc(9)

freeList $\rightarrow 10 \rightarrow 4 \rightarrow 8 \rightarrow 8 \rightarrow 7 \rightarrow 9 \rightarrow 12 \rightarrow 6$

* 3 Pts for choosing the correct block to be allocated, and 2 Pts for showing the correct block left over.

2. (4 Pts) A naïve or simplistic implementation of *best fit* and *worst fit* would have the system evaluate every block in the freeList for $O(n)$ time complexity. What additional step might be done to reduce the search time in an implementation of *worst fit* or *best fit*? What would be

the Big-O time complexity for the process of searching for the best fit or worst fit block after this additional step were done?

If there were a way to retain the alterations made to the freeList as malloc()'s were called then the best and worst fit cases wouldn't need to have $O(n)$ complexity but would become linear.

Another potential option would be to have the elements stored in a searchable ordered format such as a binary search tree. The complexity would be around $O(\log n)$, which beats the current $O(n)$ time complexity. One could potentially also utilize a skip list to fulfill this purpose which has the complexity of $O(\log n)$. Another option could be utilizing the binary buddy system which would result in $O(\log \text{ base } 2 \text{ of } n)$.