MP1 Report

June 6, 2017

1. Describing your findings.

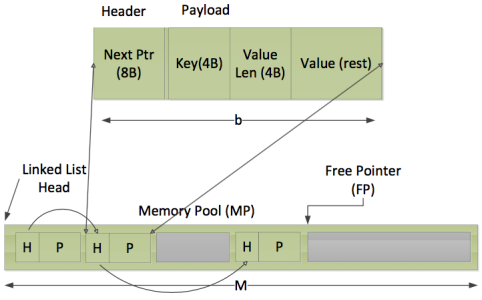
```
This value length is too big to fit into this block!!
key is 100, value length is 50
key is 5, value length is 50
key is 200, value length is 50
key is 7, value length is 50
key is 39, value length is 50
key is 25, value length is 50
key is 400, value length is 50
key is 50, value length is 50
key is 200, value length is 50
key is 300, value length is 50
The node with the key 7 has been deleted!!
The node with the key 55 doesn't exist!!
There is not enough memory to insert a node!!
The node with the key 3 doesn't exist!!
key is 100, value length is 50
key is 5, value length is 50
key is 200, value length is 50
key is 39, value length is 50
key is 400, value length is 50
key is 200, value length is 50
key is 300, value length is 50
key is 13, value length is 17
The node with the key 3 doesn't exist!!
The node with the key 13 has been founded!!
Key = 13, Value Len = 17, Value = 18268192
```

2. Do you notice any wastage of memory when items are deleted? If so, can your program avoid such wastage? How would you do so?

Yes, there is some wastage of memory when items are deleted. When an item is deleted, we move all the items to its previous location. But the last memoy of this linked list still be used. It is a wastage of memory.

3. Can you think of a scenario where there is space in the memory but no insertion is possible?

Figure #1: Structural view of a linked list in memory



From this structure, we can see when the value length is short and the block size is relatively bigger, we can got some remmaing memory in the value(rest). For this part of memory, we can't insert any node, because it still belongs to its original node.

4. What is the maximum size of the value when the pointers are 8 bytes?

The maximum size of value = block size - (size of(pointer)=8) - (size of(key)=4) - (size of(value length)=4)

5. compile and run way

compile:make

run: ./testlist