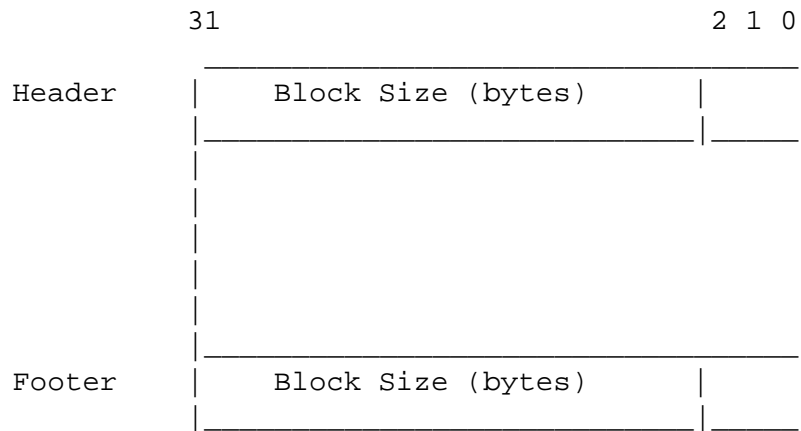


Dynamic storage allocation

The following problem concerns dynamic storage allocation.

Consider an allocator that uses an implicit free list. The layout of each allocated and free memory block is as follows:



Each memory block, either allocated or free, has a size that is a multiple of eight bytes. Thus, only the 29 higher order bits in the header and footer are needed to record block size, which includes the header and footer. The usage of the remaining 3 lower order bits is as follows:

- bit 0 indicates the use of the current block: 1 for allocated, 0 for free.
- bit 1 indicates the use of the previous adjacent block: 1 for allocated, 0 for free.
- bit 2 is unused and is always set to be 0.

Problem 64. (8 points):

Given the contents of the heap shown on the left, show the new contents of the heap (in the right table) after a call to `free(0x400b010)` is executed. Your answers should be given as hex values. Note that the address grows from bottom up. Assume that the allocator uses immediate coalescing, that is, adjacent free blocks are merged immediately each time a block is freed.

Address

0x400b028

0x00000012

0x400b024

0x400b611c

0x400b020

0x400b512c

0x400b01c

0x00000012

0x400b018

0x00000013

0x400b014

0x400b511c

0x400b010

0x400b601c

0x400b00c

0x00000013

0x400b008

0x00000013

0x400b004

0x400b601c

0x400b000

0x400b511c

0x400affc

0x00000013

Address

0x400b028

0x400b024

0x400b020

0x400b01c

0x400b018

0x400b014

0x400b010

0x400b00c

0x400b008

0x400b004

0x400b000

0x400affc