0.7x

Implement malloc and free – also PA4

Operations on pointers

`int32_t * a` assume a holds address 0x...100

We have done:

lookup at index `a[i]`

assign at index `a[i] = v`

print `printf("%p\n", a)`

`a[3]` what address is looked up? 0x...160 / 0x...10C / 0x...110 (0x..100 + 32*3) — 32 bits, so 4 bytes, so +100+12

What would we write to set address 0x...114 to 37?

`a[5] = 37`

`uint8_t * b` assume b holds address 0x..200

`b[3]` what address? 0x..203

What would we write to set 0x...214 to 37? `b[20] = 37`

`a[3]` vs. `b[3]` the size of the type int32_t vs uint8_t determines the offset/address to use

dereferencing a pointer

`*a` it means look up the value at a

equiv to `a[0]`

`a + n` (pointer arithmetic) means calculate the address at start byte

`a + sizeof(T) * n` where a's type is T*

`a + 4*n` when a's type is int32_t*

69

*(a+3) means a[3]

*(a+5) = 37 means a[5] = 37