Moving Data in assembly:

movq Source, Destination; dereferences if an address (moves an actual value to the destination)

leaq: moves an actual address to the destination.

Immediate: constant integer data

Register: one of the 16 registers

Memory: 8 consecutive bytes of memory at address given by register

One, but NOT BOTH source and destination can come from memory.

Macro: preprocessor directive replaces all instances of macro with the definition in code.

Example: #define X 17 replaces all instances of X in the code with 17

Nested Array Element Access in assembly:

For reading data, try and read sets of X byte lengths, where you have an X-byte machine  
(i.e. read 4 bytes at a time on a 4 byte machine, 8 bytes on 8 byte machines, etc.)

Placing the struct in memory: look at the largest data type inside the struct, make sure that is placed on the boundary it needs, which will then make more structs be placed on the boundary they need, and all the smaller ones are fine.

The Stack has an 8 MB limit, if you try to allocate memory somewhere and your program crashes, that might be why. Dynamically allocate to fix that problem, to take memory from the heap instead of the stack.

Direct-mapped cache: (E = 1), one line per set

Fully associative cache: One set; every line in the cache is within the same set.

Coalescing: combining adjacent free blocks when free is called.

Deferred coalescing: don’t coalesce *every* time free is called, but only sometimes. Those times might be when traversing the free list for malloc, or when the external fragmentation reaches a certain point.