	Heap	Stack	Global.bss	Global.data	Porque?
Hypercounter				X	Fora de qualquer método e inicializado
Counter			X		Fora de qualquer método e não inicializado
*idk2					
Cook1		X			Memorias alocadas diretamente e dentro de um método
Clk1		X			Memorias alocadas diretamente e dentro de um método
Cook2->c		X			Memorias alocadas diretamente e dentro de um método
*arr em clk1	X				Memoria alocada somente na chamada de uma execução
*arr em clk2	X				Memória alocada somente na chamada de uma execução

---

Link vídeo: https://youtu.be/Lr_ZdcfHK0c

Heap memory: É uma porção da memória alocada dinamicamente, ela continua a existir até que ela seja liberada ou o programa terminado.

Stack memory: É aonde variáveis locais vivem, memorias alocadas nessa área vivem apenas até o fim da função que elas compõem.

Global memory: São variáveis que existem durante todo o programa, inicializadas ou não. Elas existem enquanto o programa existir e são destruídas assim que finalizado.

Fragmentação externa e eficiência: Fragmentação externa acontece quando memoria livre é separada em pequenos blocos e é interceptada por memoria alocada. Isso é uma falha de alguns algoritmos de alocação de memória, quando eles falham em ordenar a memoria usada no programa eficientemente.

Fixed-size blocks: Alocação de blocos de tamanho fixo usa uma lista de blocos de memoria livre de tamanho fixo, geralmente todos de mesmo tamanho. Isso funciona bem para sistemas embarcados simples onde nenhum objeto grande precisa ser alocado, mas sofre de fragmentação, especialmente com endereços longos de memória.

Buddy blocks: Nesse sistema, memoria é alocada em vários “poços” de memoria em vez de apenas um, aonde cada “poço” representa um bloco de memoria com um tamanho 2^n, ou alguma outra progressão de tamanho conveniente. Todos os blocos de um tamanho especifico são mantidos/armazenados em uma lista ou arvore ordenada e todo novo bloco que é formado durante à alocação são adicionados para seu “poço” especifico para uso posterior.

Slab allocation: Esse mecanismo de alocação de memória “pre-aloca” dita memoria em pedaços cabíveis para objetos de certo tipo ou tamanho. Esses pedaços são chamados de caches e o alocador apenas tem que manter um registro da lista de lugares vários na cache.

Stack allocation: Stack é uma área especial do computador aonde são guardadas temporariamente variáveis criadas por uma função. No Stack, as variáveis são declaradas, armazenadas e inicializadas durantes o tempo de funcionamento do programa.

Alinhamento de memória Globalmente: Não achei no moodle o livro e não achei acesso virtual

Em pilha: Não achei no moodle o livro e não achei acesso virtual

Dinamicamente: Não achei no moodle o livro e não achei acesso virtual

---

Sites de consulta:

https://www.guru99.com/stack-vs-heap.html

https://en.wikipedia.org/wiki/Memory_management

https://www.geeksforgeeks.org/memory-layout-of-c-program/

https://www.geeksforgeeks.org/stack-vs-heap-memory-allocation/