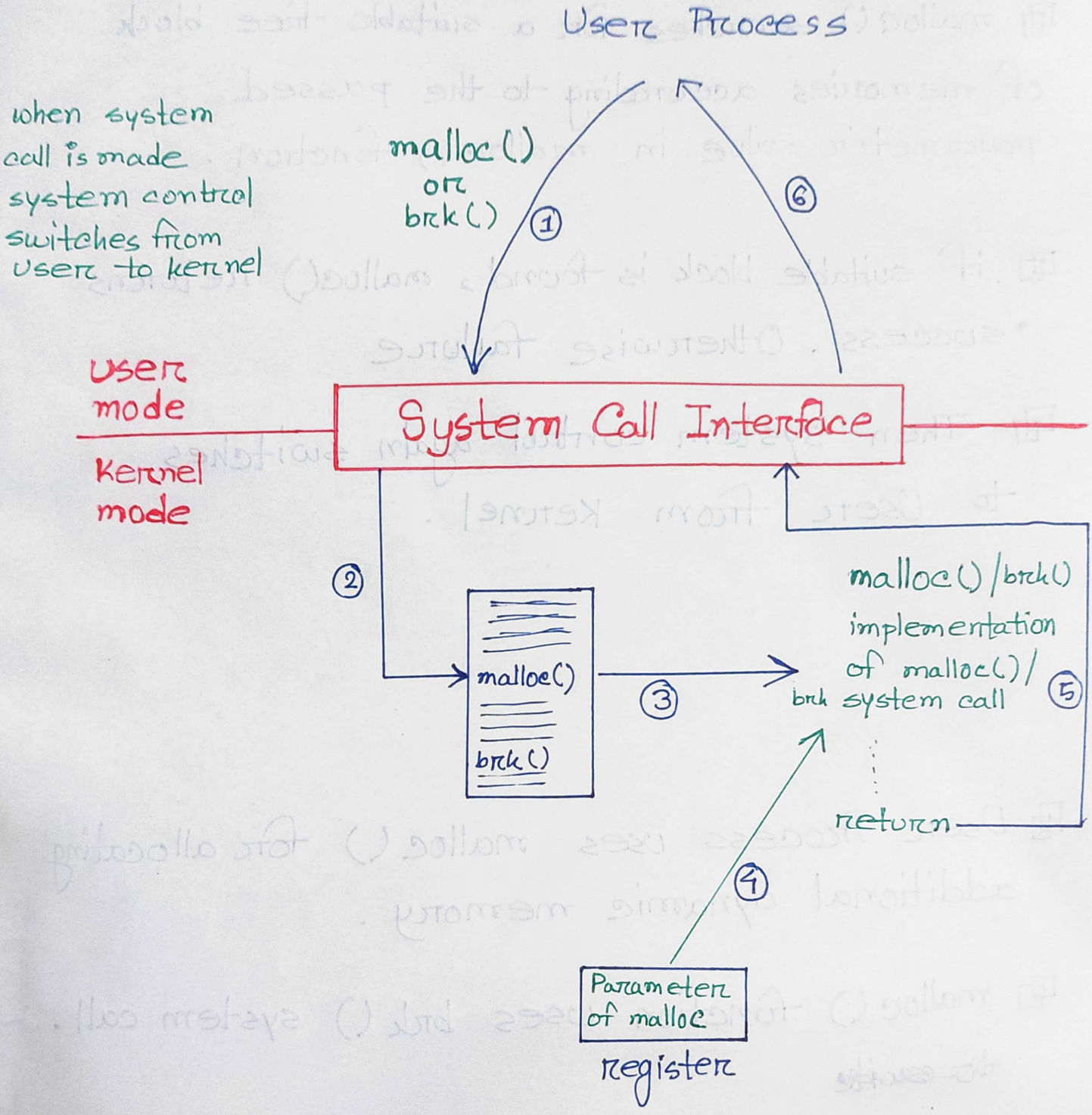

User Process
when system call is made system control switches from user to kernel
malloc() or brk()
1
6
User mode
Kernel mode
System Call Interface
2
malloc()
brk()
3
malloc()/brk() implementation of malloc()/brk system call
5
return
4
Parameter of malloc
register

- User Process uses malloc() for allocating additional dynamic memory.

- malloc() function uses brk() system call.

- When system call happens, system control will be switched from user mode to kernel mode. Because kernel mode is the privilaged one which only can access hardware like memory, I/O devices.

- Then in kernel mode, OS finds brk/malloc Application program.

- malloc() internally uses brk() system call

- internally, malloc() manages a pool of memory obtained from the OS using system calls like brk()

- malloc() searches for a suitable free block of memories according to the passed parameter value in malloc() function.

- if suitable block is found, malloc() returns success. Otherwise failure

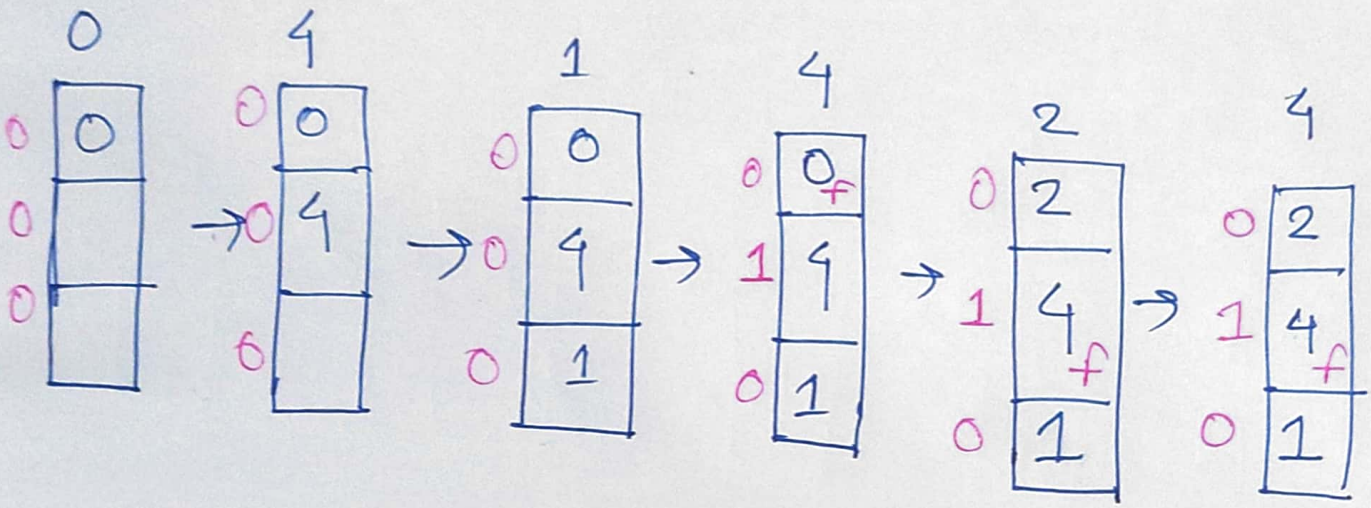
- Then system control again switches to User from Kernel.

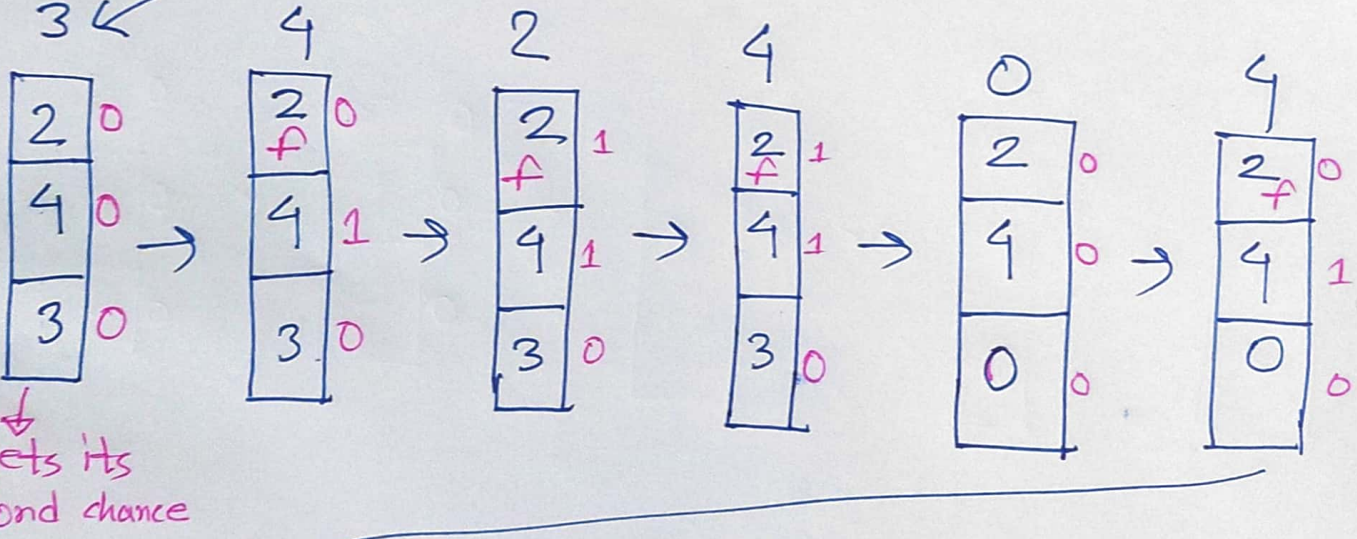
Does OS allow all requests for additional memory?

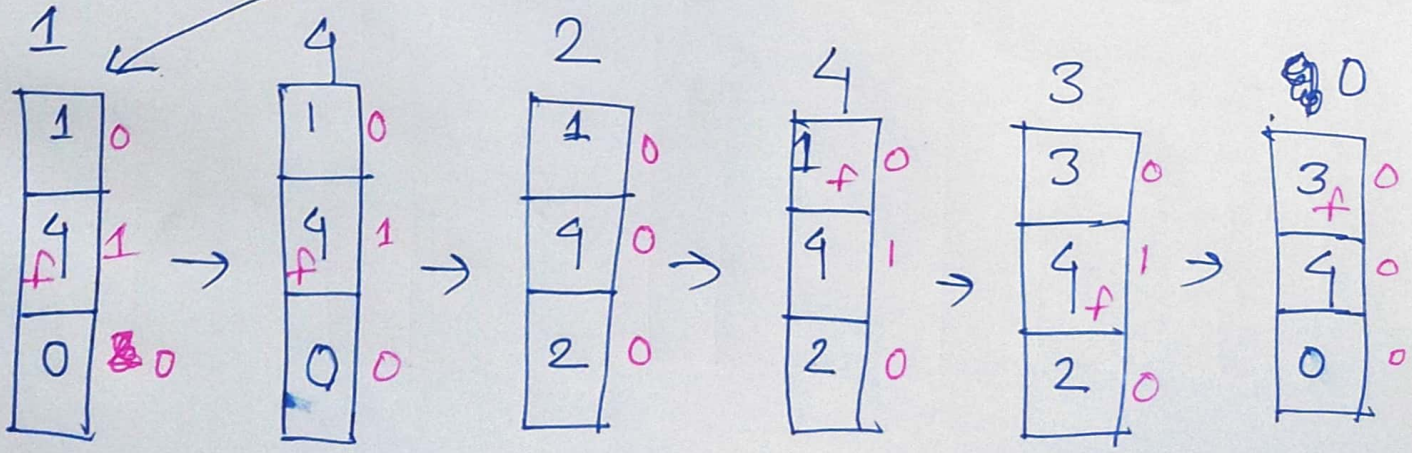
- OS does not allow all processes requests of additional memory allocation
- OS ensures memory protection
- Memory Protection uses protection bits.
- If the process which requests for additional memory is privileged one (such as {administrative or root), then it checks for whether the process has enough permissions (r,w ...), then if both matches, OS allows for additional memory request. Otherwise OS denies the request.

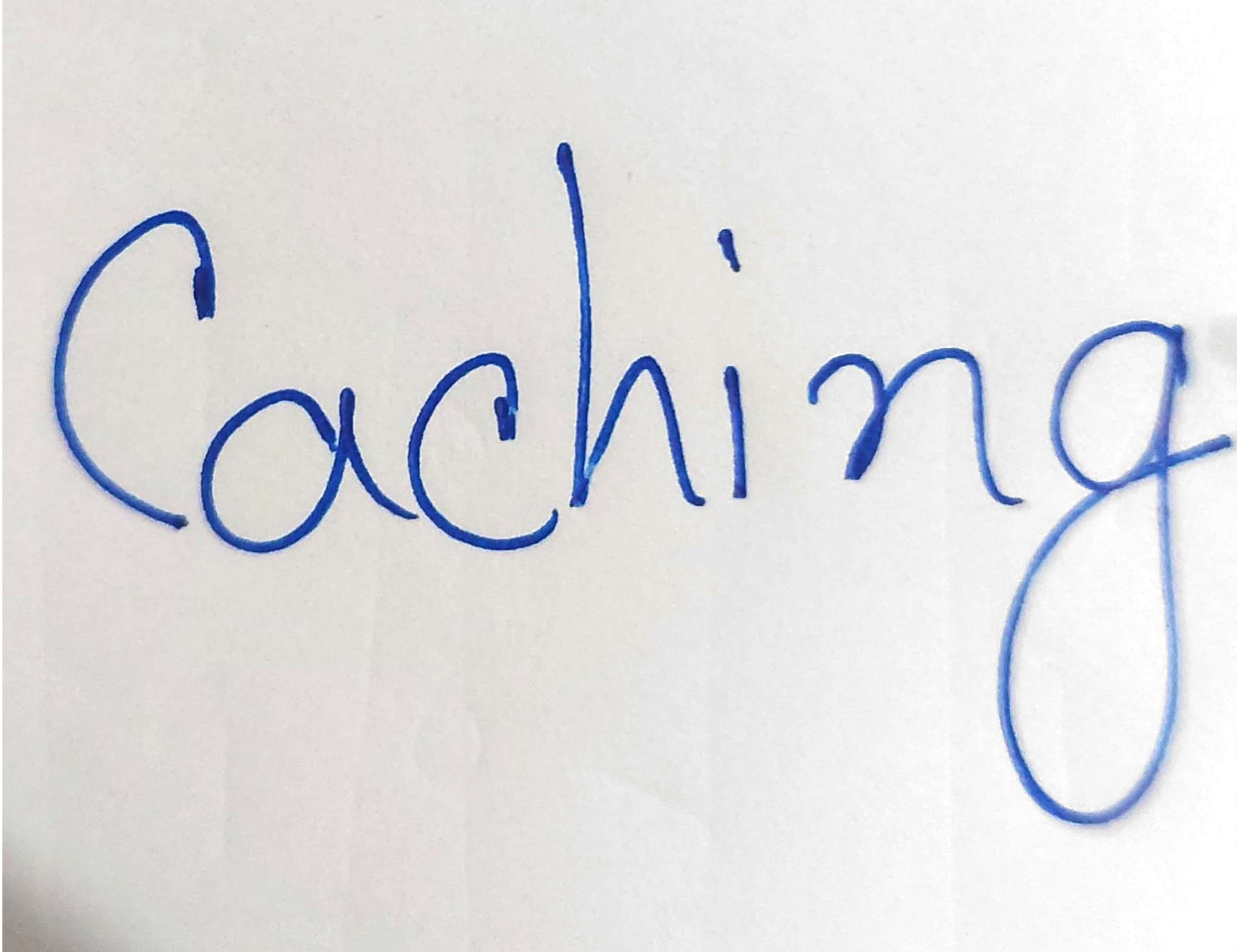
Second Chance / Clock Page Replacement Algo

● 1 → gets a second chance to be replaced.








Coaching

Caching is a technique used in computer science to temporarily store data in a quickly accessible location, typically in memory, to expedite future access to that data

Process of caching:

- **Request for Data**: When a system or application needs certain data, it first checks the cache to see if the data is already stored there.
- **Cache Check**: If the requested data is found in the cache, it's known as a cache hit. The data is retrieved directly from the cache, bypassing the slower data source.
- **Cache Miss**: If the requested data is not found in the cache, it's known as a cache miss. In this case, the system retrieves the data from the original data source, such as a database or a web server.
- **Data Retrieval**: After retrieving the data from the original source, it's stored in the cache for future accesses.
- **Expiration and Eviction**: Cached data may have a limited lifespan or may be evicted from the cache based on certain policies (like LRU - Least Recently Used, LFU - Least Frequently Used, etc.) to make room for newer or more frequently accessed data.
- **Repeat**: This process continues as long as the system continues to request data, with the cache serving as a middle layer between the system and the original data source.