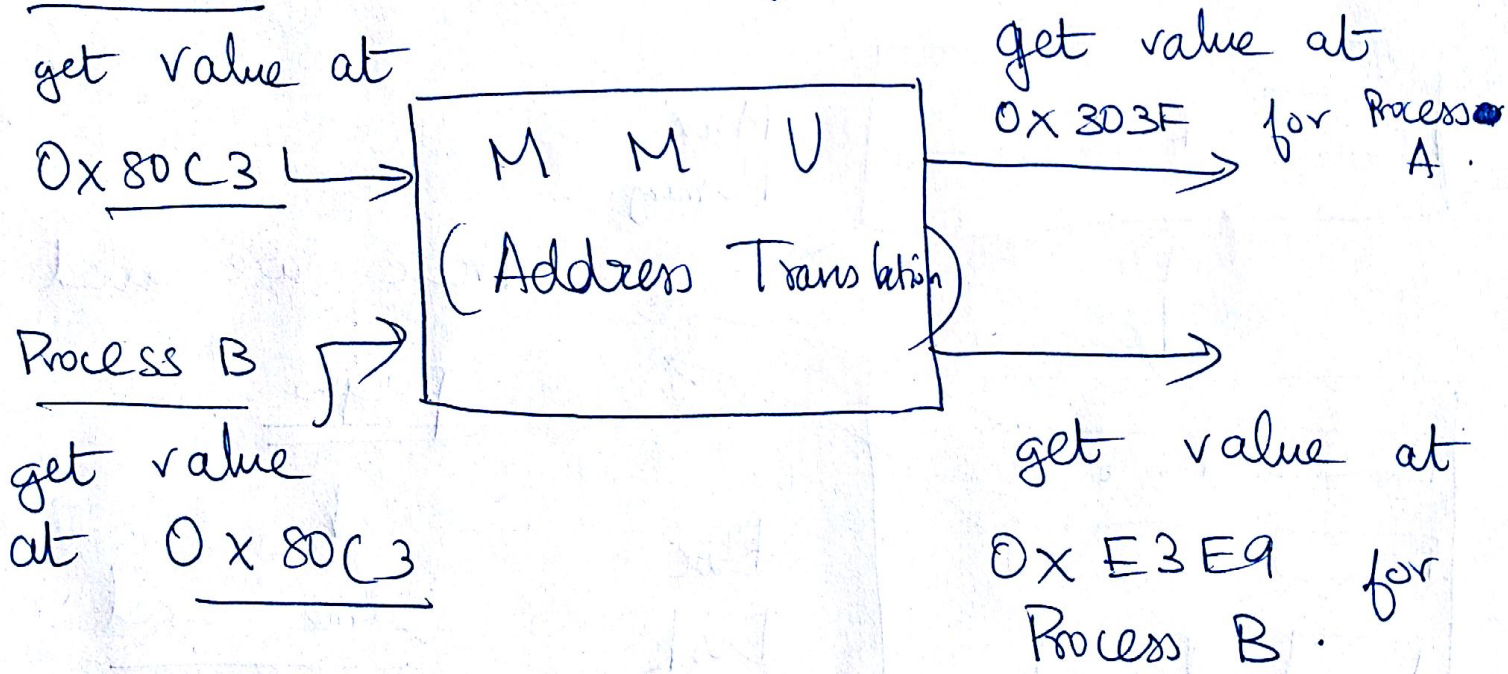
4/6.

Chapter 6 -> Ignore 6.1 & 6.6

Last Class

Address Protection. → Seperating the address spaces of different processes.

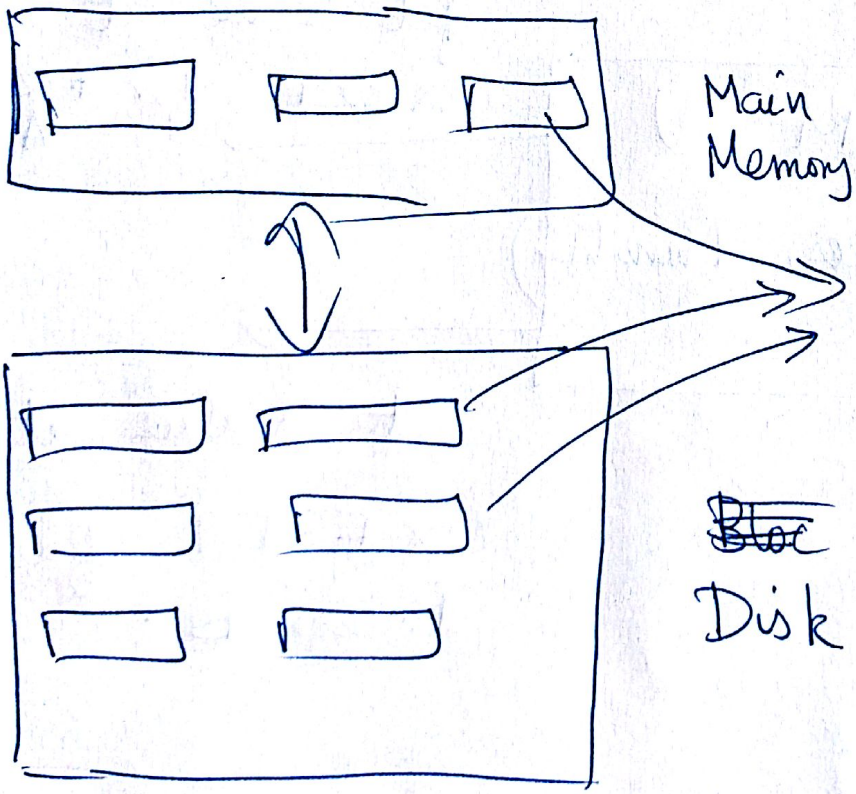
Process A → a program in execution.



Virtual Addressing → Virtual Memory.

it is implemented by the OS + Hardware.

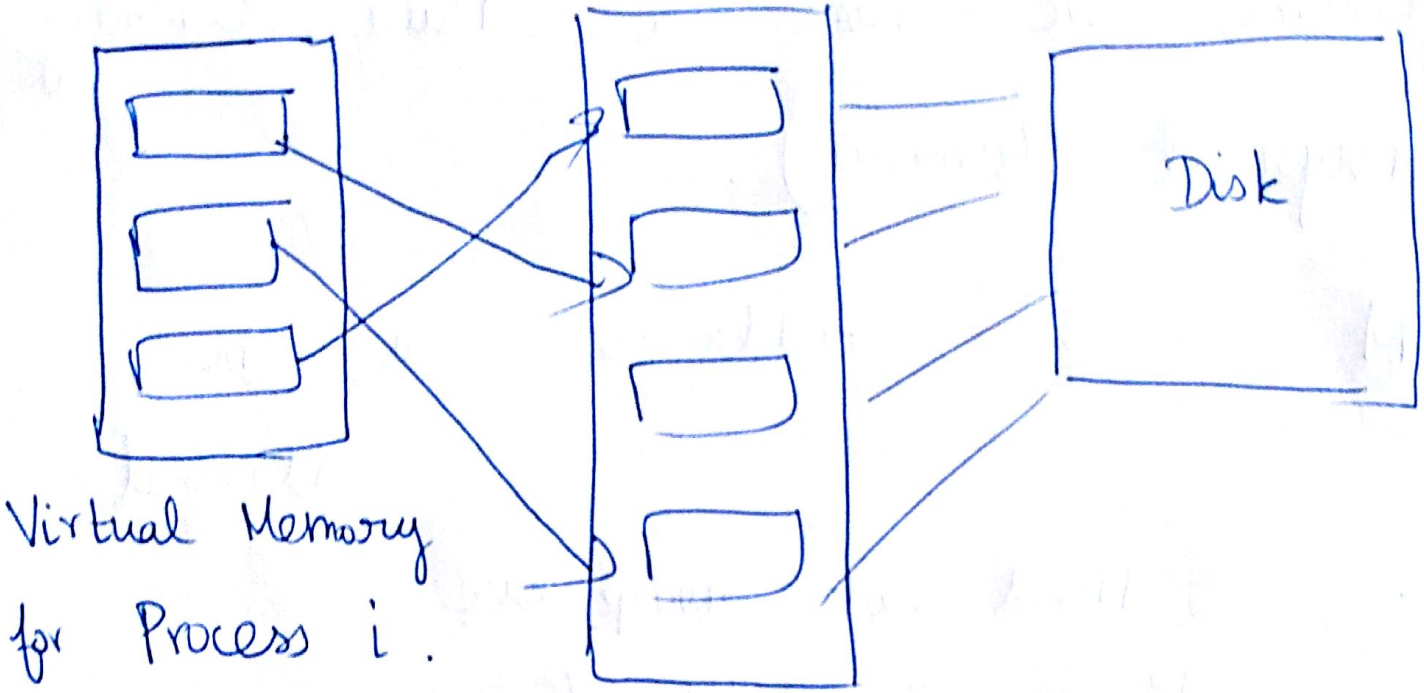
One more level of abstraction.



These are called pages !

↓

same-sized.



Suppose we have a $n = 32$ (-bit) processor..

↘

can address

$N = 2^n = 4.3$ billion addresses

00...00 32 0's

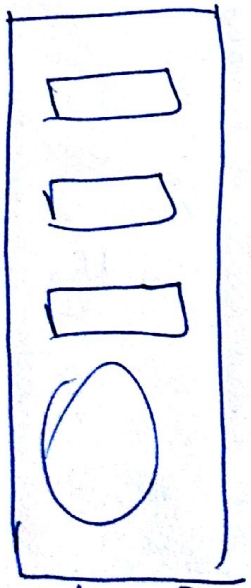
11...11 32 1's

Assume we have a Main Memory (Physical Memory).

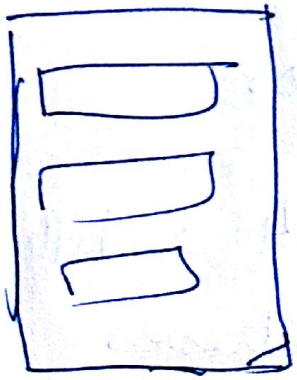
$M = 2^{(m)}$ addresses can be addressed.

(Need not always have to be a power of two)

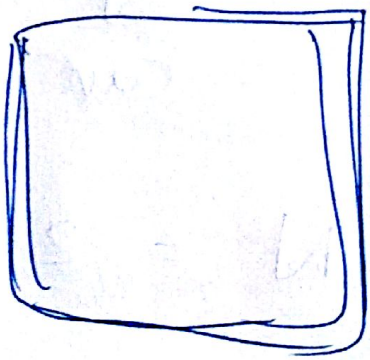
What if the 'virtual' memory is larger than the physical memory available?



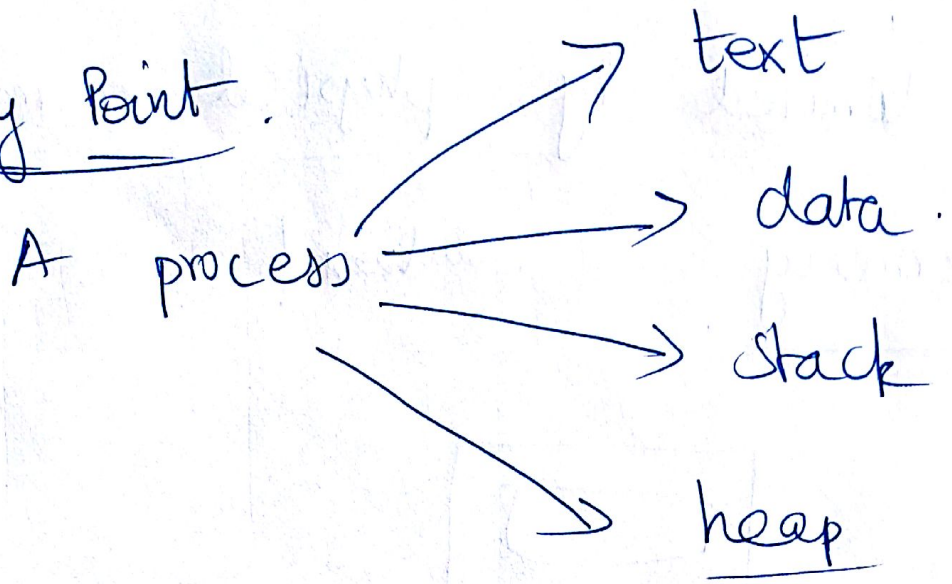
VM for Process i



Physical memory



Key Point.



need to be in the main memory completely for the process to own.

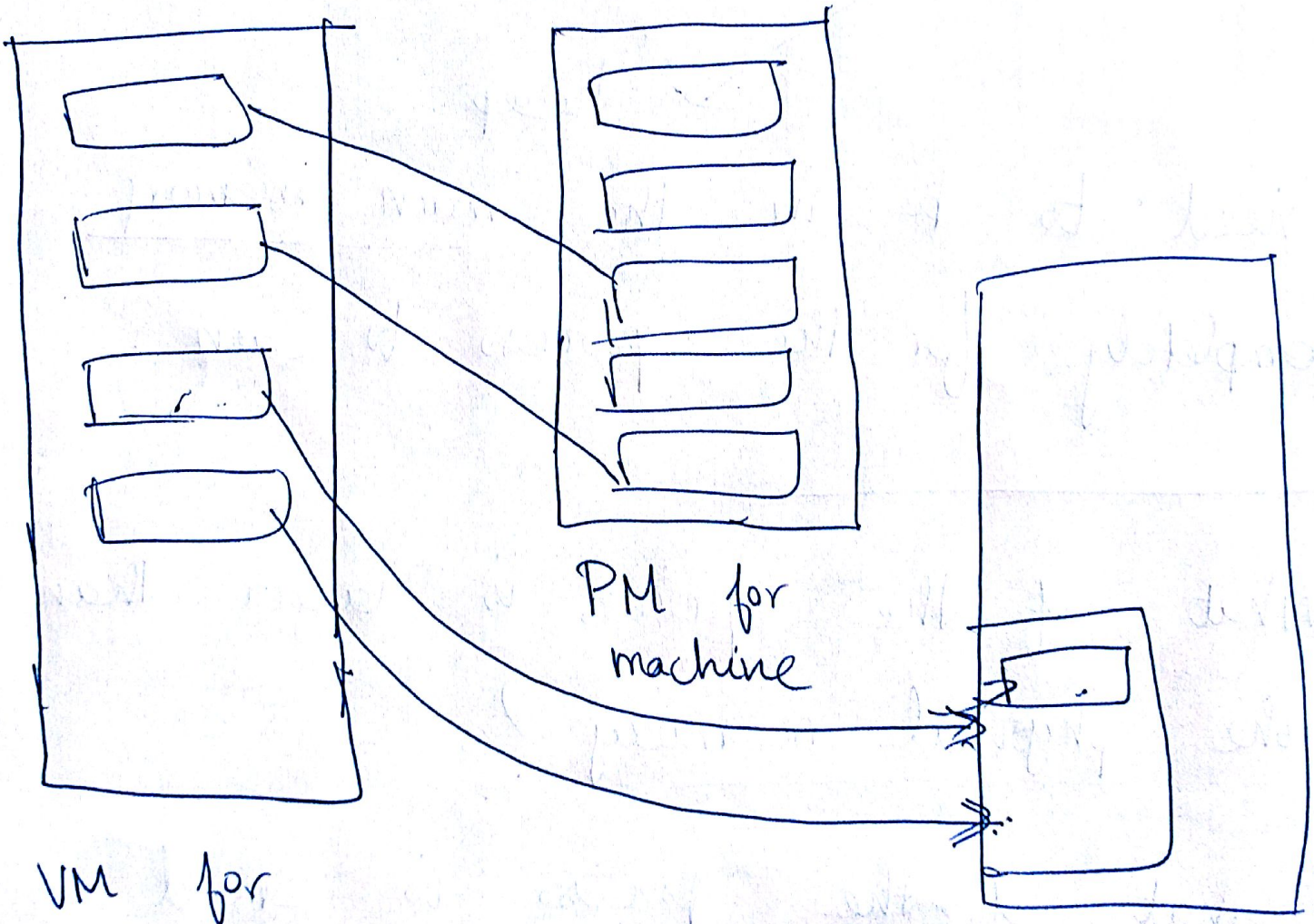
---

① What if the process is larger than the physical memory?

② What if the process is small enough to fit in the physical memory, BUT the physical memory doesn't have enough space?

We are limited by physical memory

Virtual Memory. ← Solution!



Lets you access more memory than what is physically available.

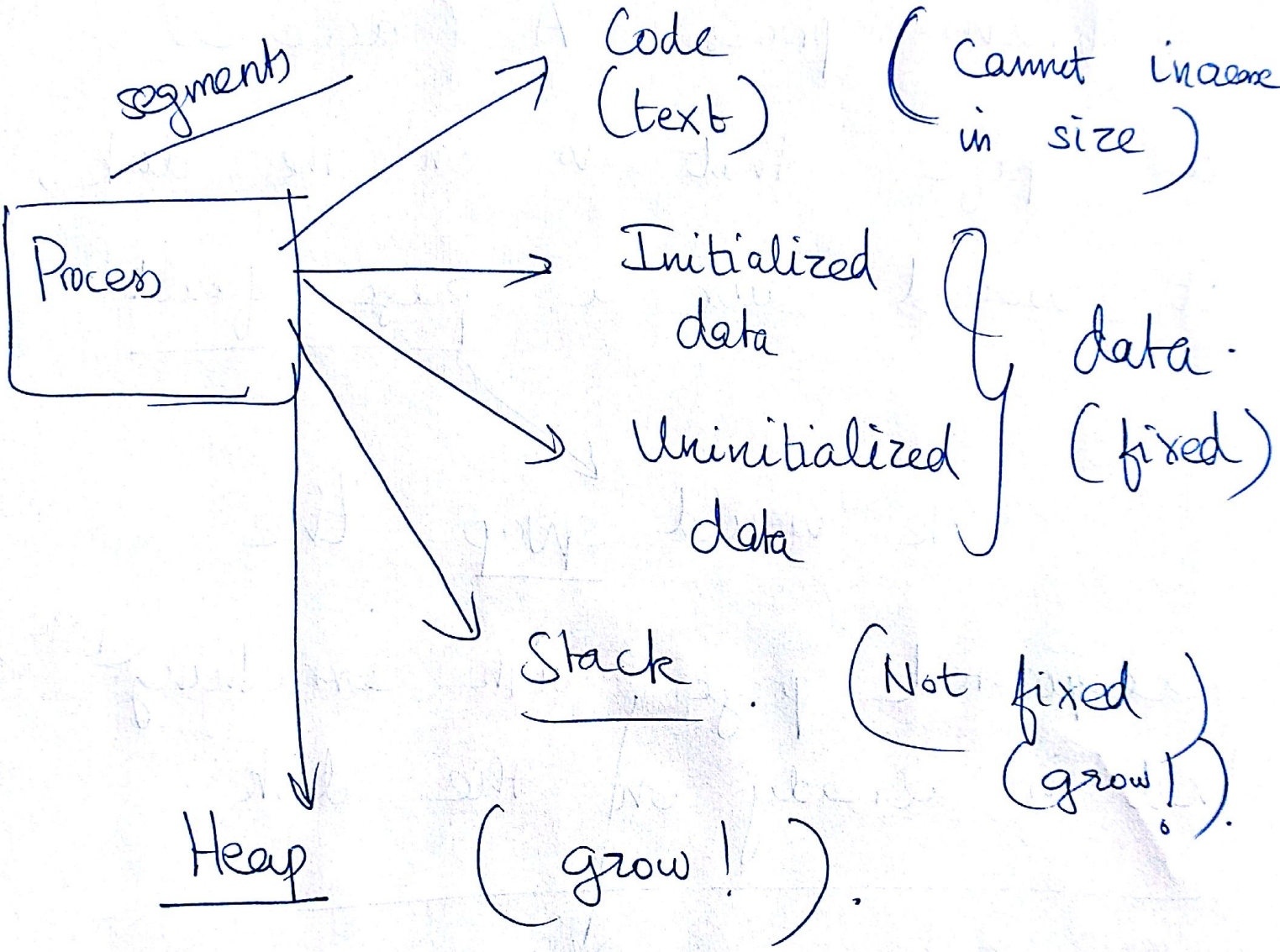
Page fault

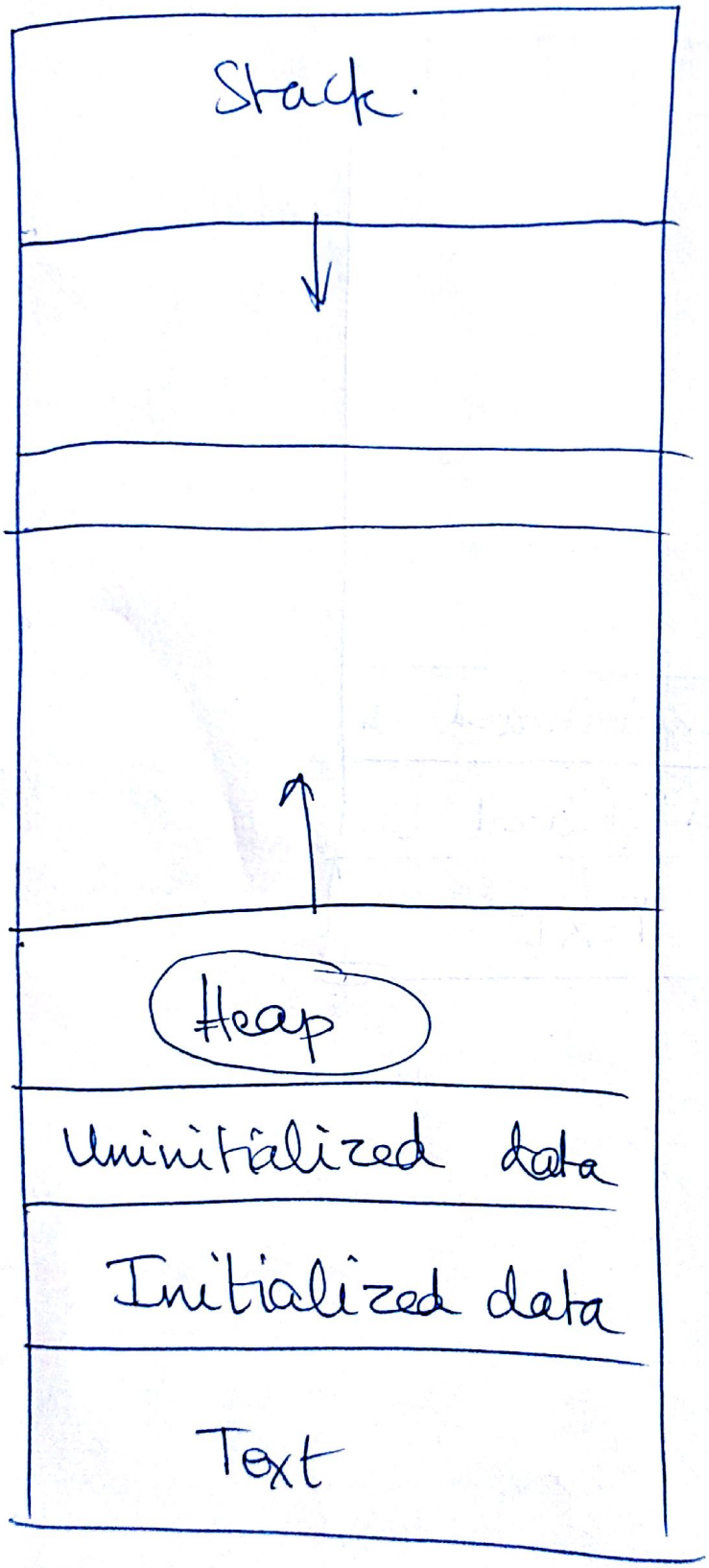
When process A accesses a page that is on the disk, it would cause a page fault.

We would swap the requested page, with something (page) that is already on the disk.

On to the physical memory

9.9 Dynamic Memory Allocation

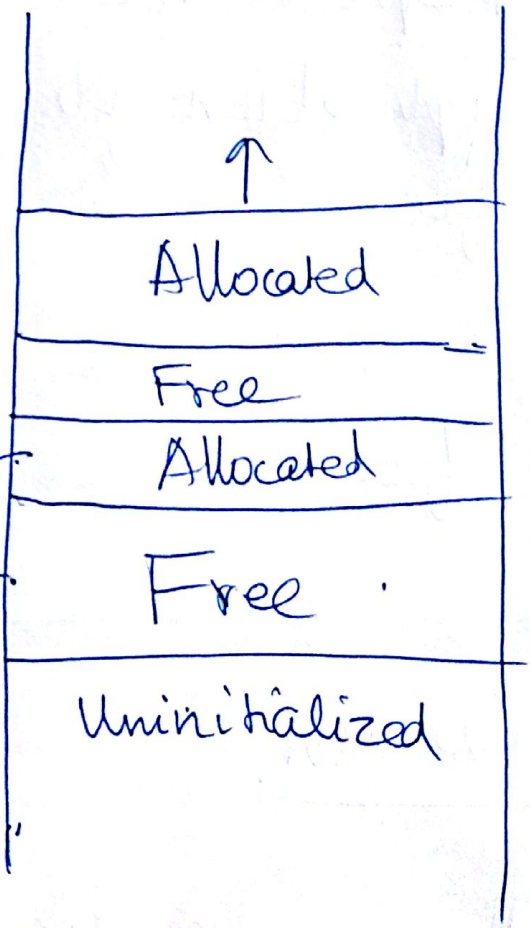




Heap → collection of various-sized blocks

each block is a contiguous chunk.

Can either be allocated or free



Remain allocated until they are freed.

Available to be allocated.

Explicit

Implicitly

Explicit Allocators

↓

Require the application to free

(C, C++)

free

Implicit Allocator

Require the allocator to detect when a block is empty an free it.

(Java)

↓

garbage collector.

Allocator

When malloc is called,

looks for a chunk of data in the heap that is big enough

+ returns a pointer to it.

+ it records that the chunk is not free anymore
  - list.

When free is called (with a pointer)

+ it identifies how big that chunk is
  - looking up the list
+ adds this to the list of free blocks.

- What if when malloc is called and there is not enough space on the heap?
  - sbrk system call (Check next lecture)