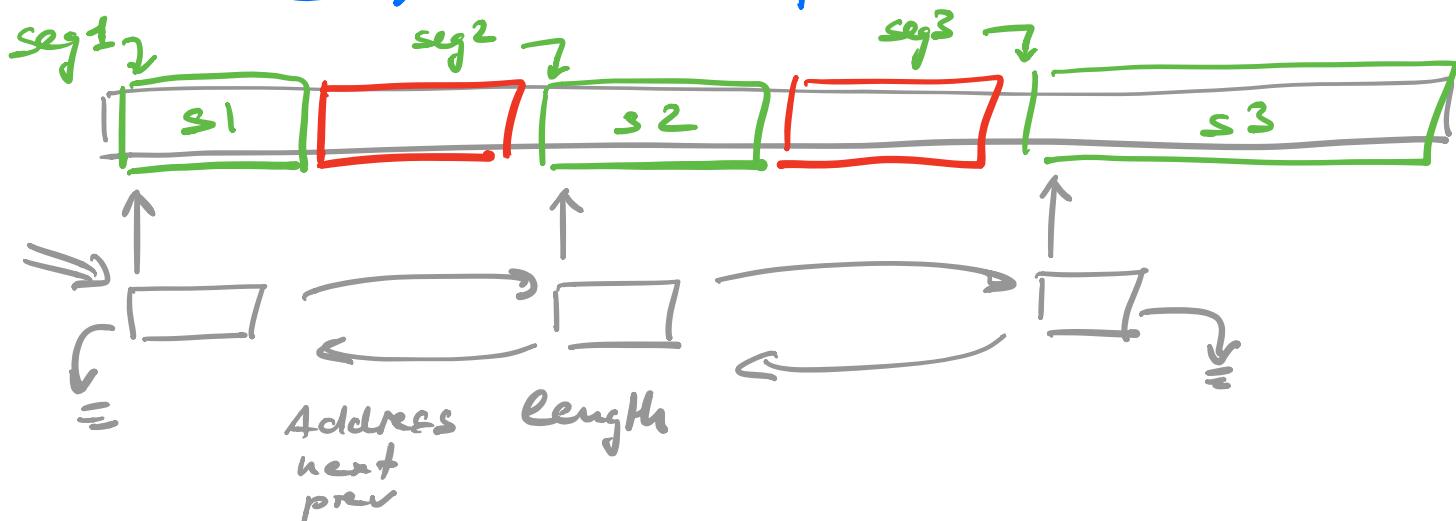


Welcome to a brief discussion of MS 2 of MP1

MS1: Salami Allocator.

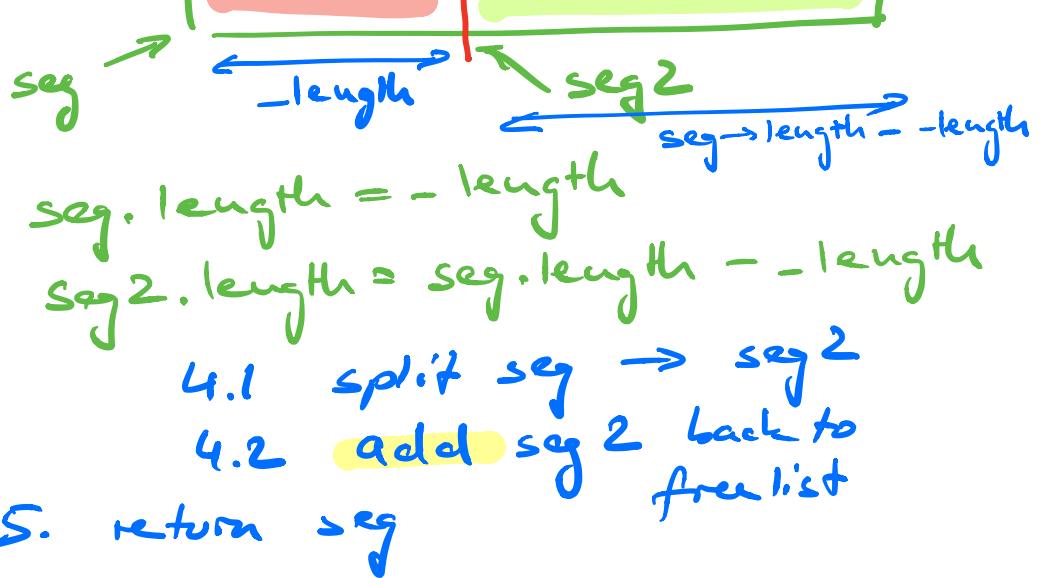
Disadvantage: cannot return memory back to allocator.

Free() had to fake it.



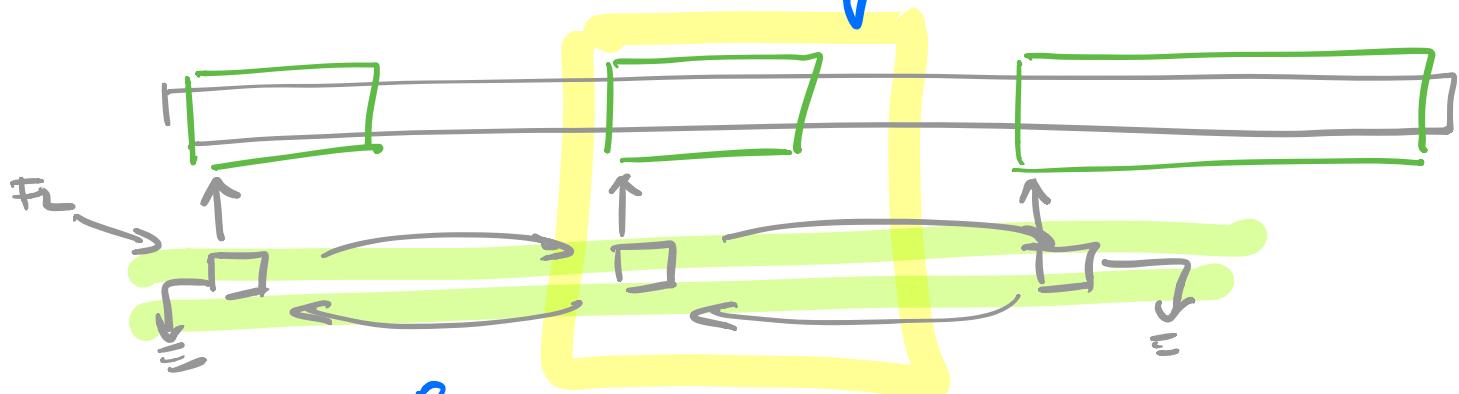
free-list.

- Malloc ($-\text{length}$)
1. traverse free-list until we find segment that is long enough.
 - 1.1 $\text{seg} = \text{freelist} \rightarrow \text{head}$.
 - 1.2 while ($\text{seg} \neq \text{null}$ & $\text{seg} \rightarrow \text{length} < -\text{length}$)
 $\text{seg} = \text{seg} \rightarrow \text{next};$
 2. $\text{seg} == \text{null} \rightarrow \text{fail!}$
 3. **remove** seg from free list.
 4. if $\text{seg} \rightarrow \text{length} > -\text{length}$
 $\text{seg} \rightarrow \text{length}$



Free(-a)

1. create seg for -a
2. add seg to free-list

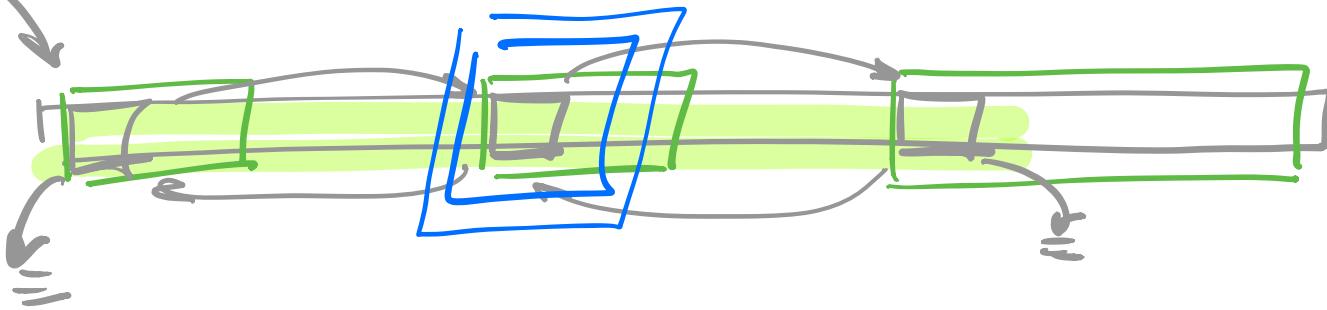
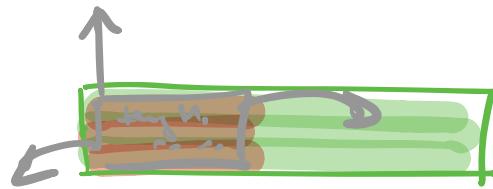


Problems!

Prob 1: Free(-a) $-a \xrightarrow{?} \text{seg}$

Prob 2:

free list



FreeList: - just a doubly-linked list.

free-list.h/c (FL)

Segment Header (SH)

- length
- cookie
- is-free

FL : Add (SH*)
Remove (SH*)

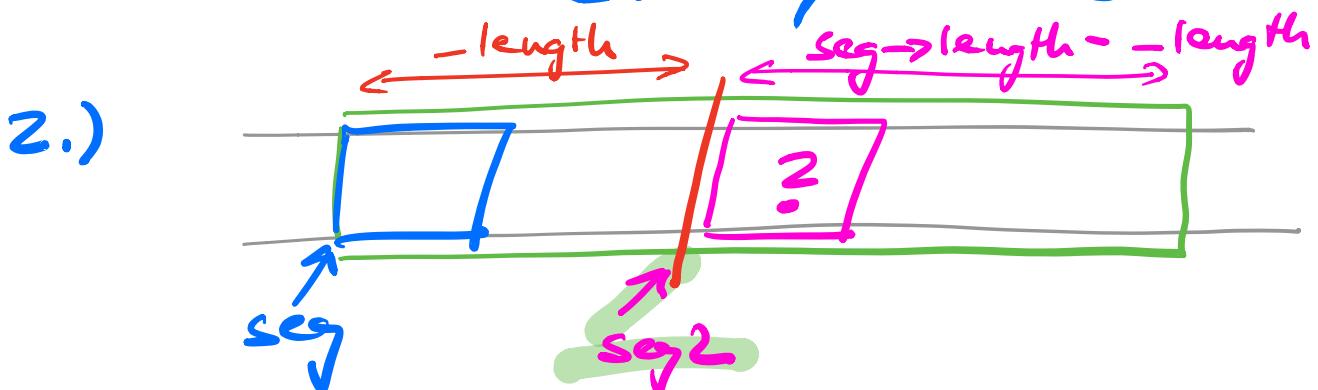
Problem: How are SH's created and placed

Creation: Where?

1.) When we create the Allocator:



Q: How do we construct a SH to be placed exactly here?!



Q: How do we force "new" to place new object at a predetermined place in memory?

Recall: "new"

1.) allocates mem
→ ptr

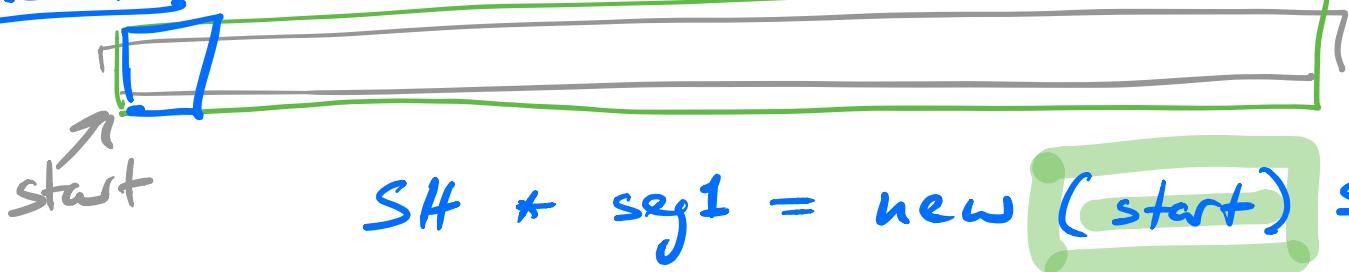
2.) call constructor.

placement new: skips step 1

$SH * seg = \text{new } SH(...);$

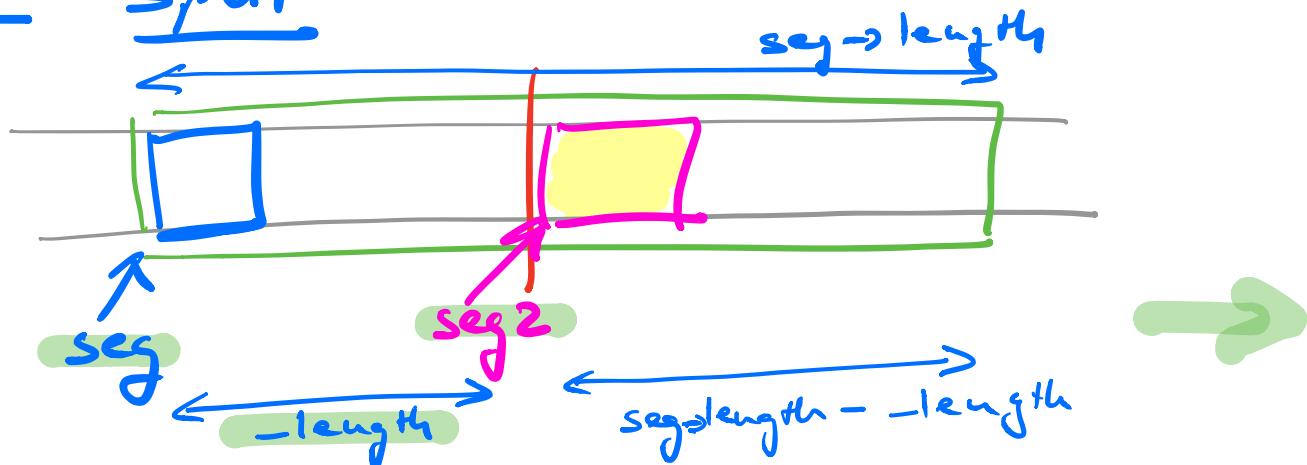
$SH * seg = \text{new } (\text{ptr}) \ SH();$

Case 1:



$SH * seg1 = \text{new } (\text{start}) \ SH();$

Case 2: split



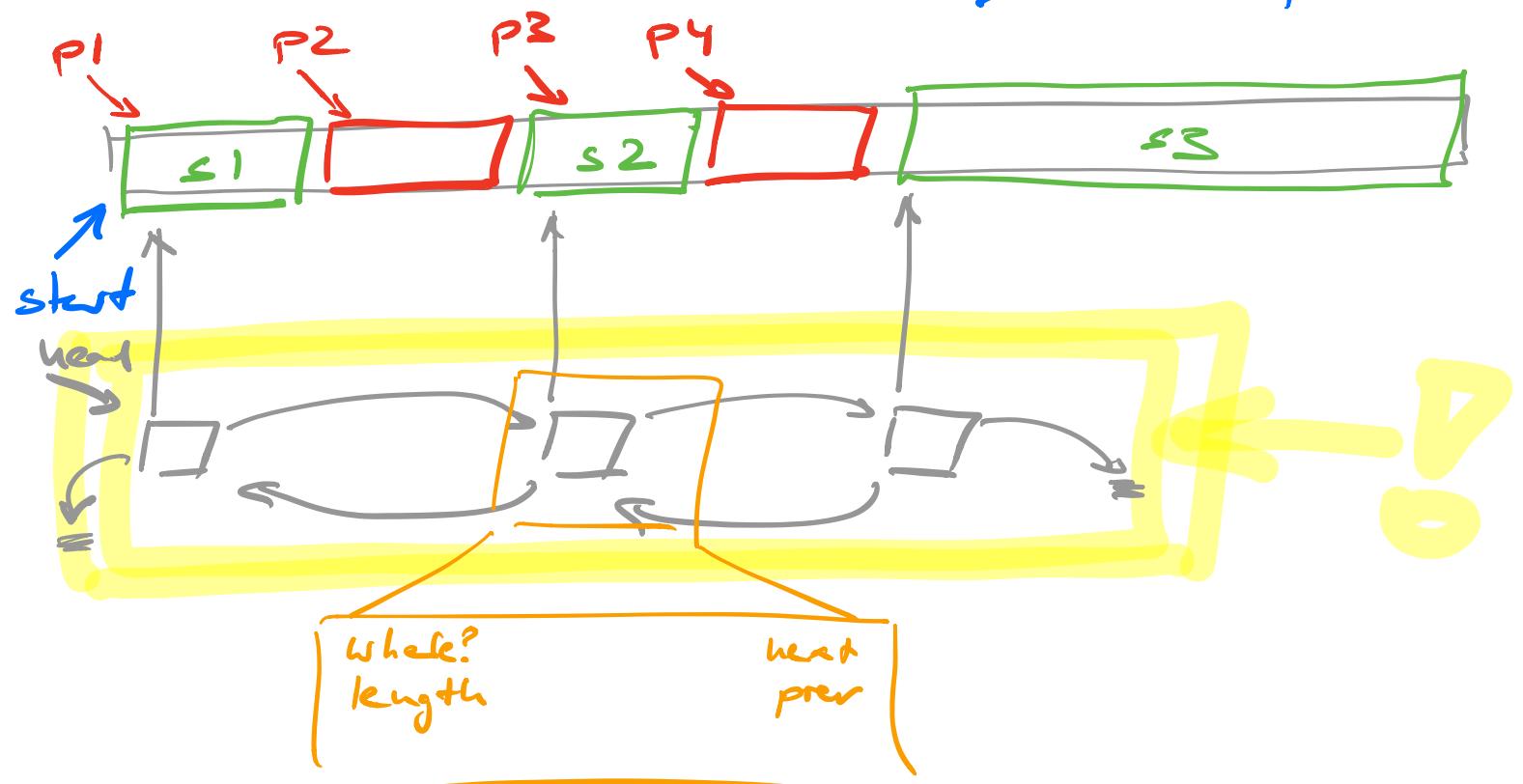
Split:

$SH * seg2 = \text{new } ((char)seg + \text{length}) \ SH(...);$

Welcome to a brief discussion of
NS2 of NP1!

NS1: Salami Allocator.

Disadvantage: Cannot return mem. to allocator \Rightarrow free() fakes it.



malloc(-length)

1. traverse list until we find long-enough segment

1.1 $seg = head;$

1.2 while ($seg \neq null \text{ and } seg \rightarrow length < -length$)
 $seg = seg \rightarrow head;$

$seg = seg \rightarrow head;$

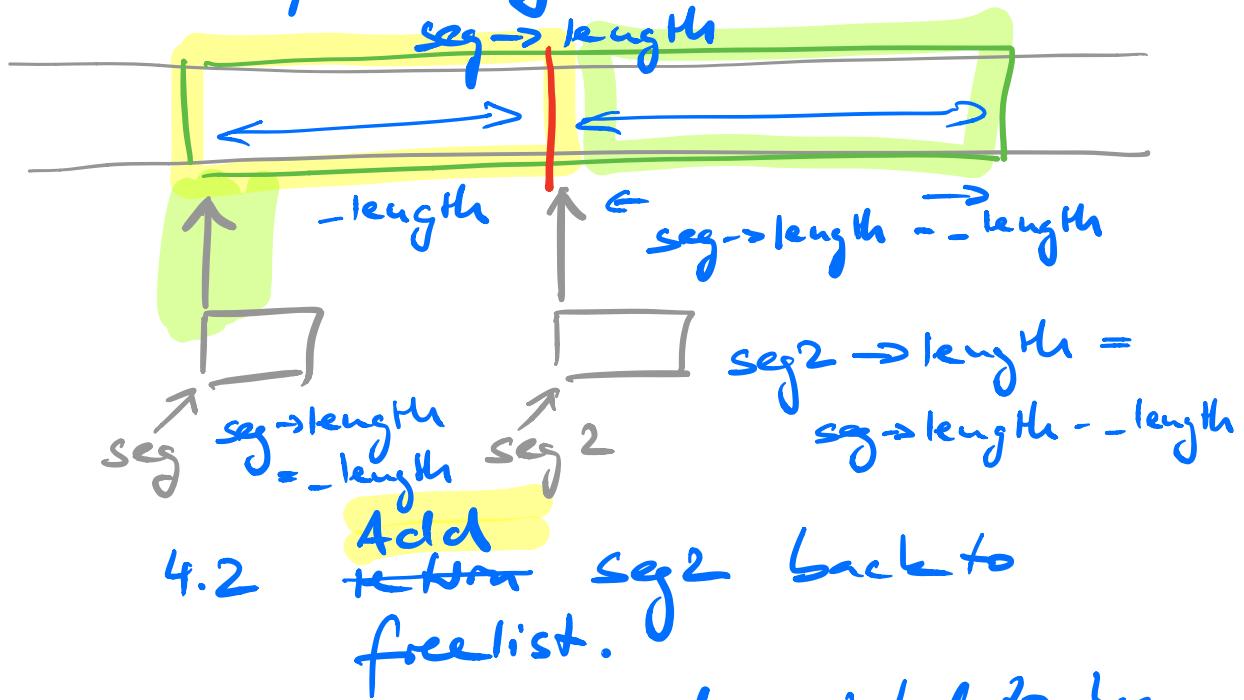
2. found?

if ($seg == null$) \rightarrow fail!

3. remove seg from freelist.

4. if ($\text{seg} \rightarrow \text{length} > -\text{length}$)

4.1 split segment



5. return segment pointed to by seg .

Free(-a):

1. map $-a$ to seg node
→ seg

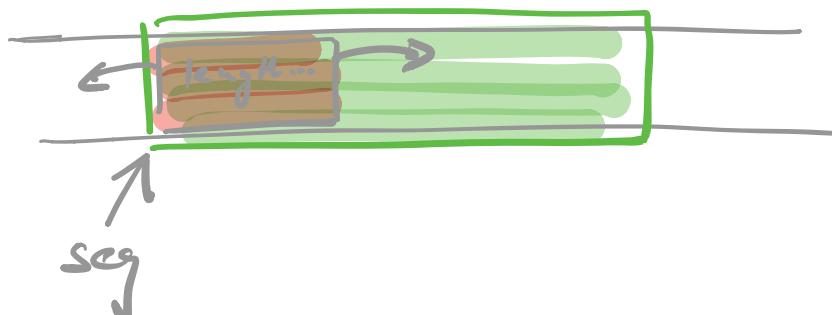
2. Add seg to freelist.

Problems:

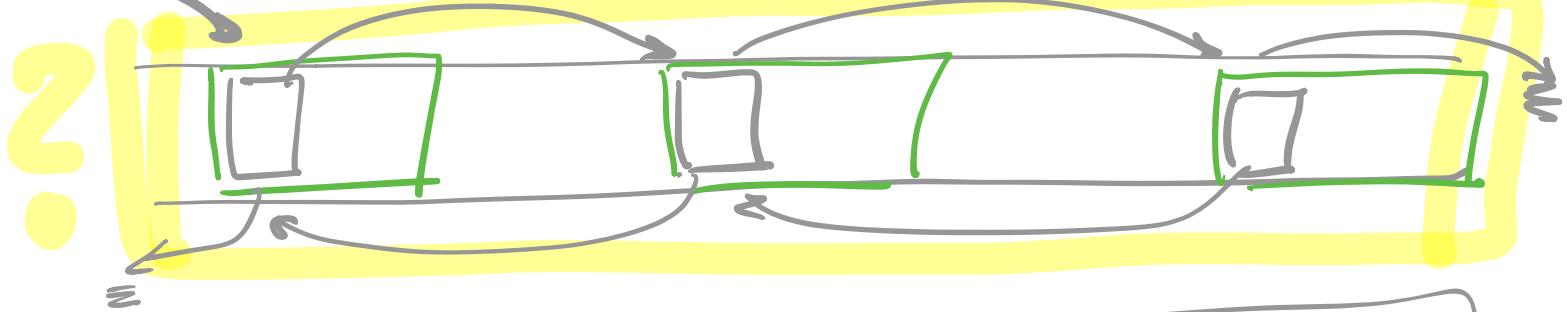
P1:

$$-a \xrightarrow{?} \text{seg}$$

P2:



head



Flist: just a doubly-linked list

$SH^* \text{ head}; (SH^* \text{ tail})$

Add (SH^*)

Remove (SH^*)

Problem: Where are the segment
headers (nodes) created
and placed?

Where?

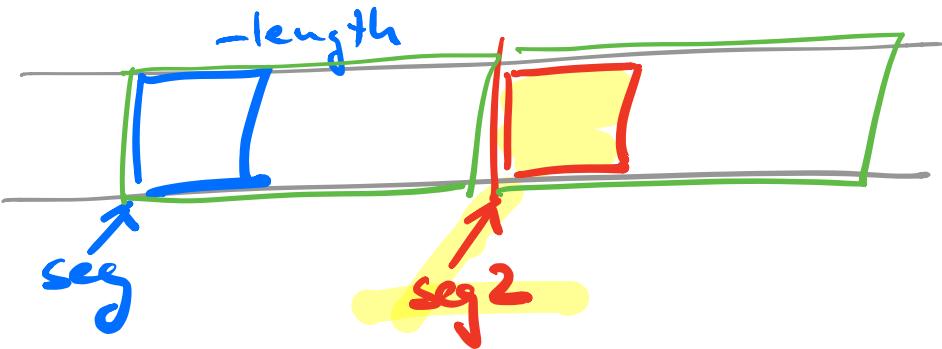
Case 1:



↑ how do I construct by seg
↑ headers so that it is placed
here?!

Case 2:

Split:



Q: How do I force "new" to place the new object at a predefined location??

Recall: What does "new" do?

1.) allocates memory for new object
→ ptr.

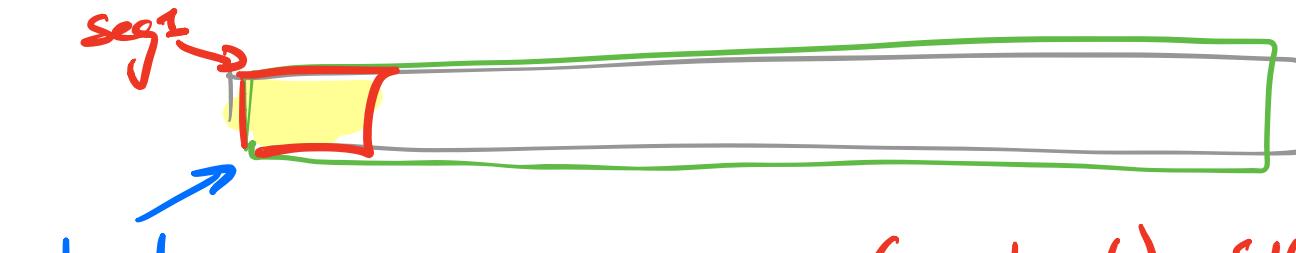
2.) call constructor for object.

"placement new"

$SH * \text{seg} = \text{new } SH(...);$

$SH * \text{seg} = \text{new}(\underline{\text{ptr}}) \ SH(...);$

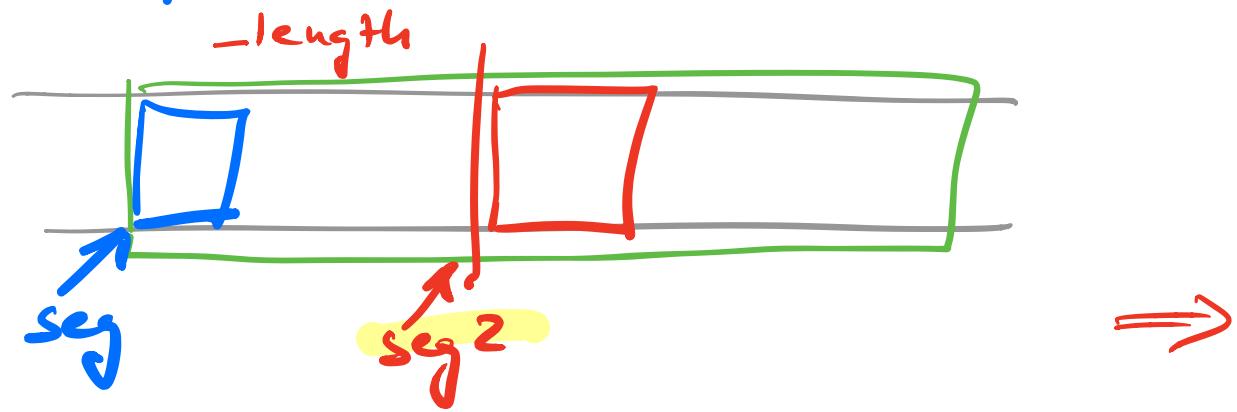
Case 1:



$\text{seg1} = \text{new}(\underline{\text{start}}) \ SH(...);$

Case 2:

"Split"



`SH * seg2 = new((char*) seg - length) SH(....)`