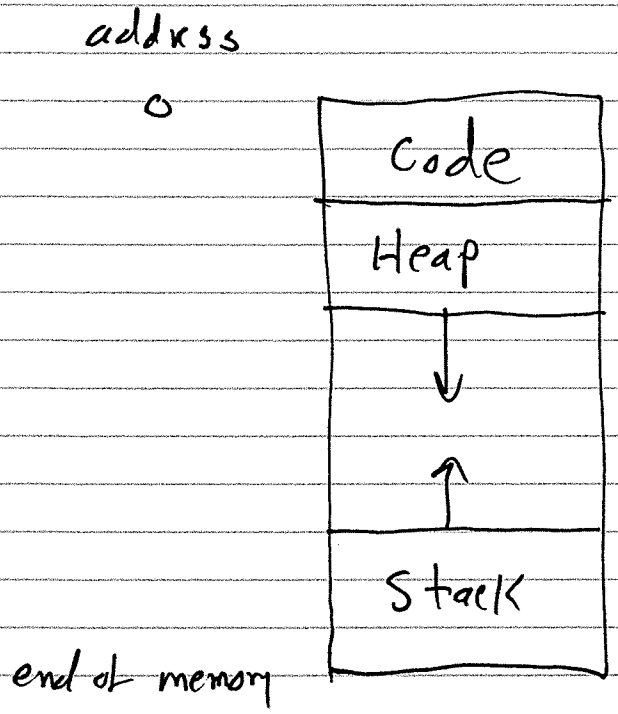
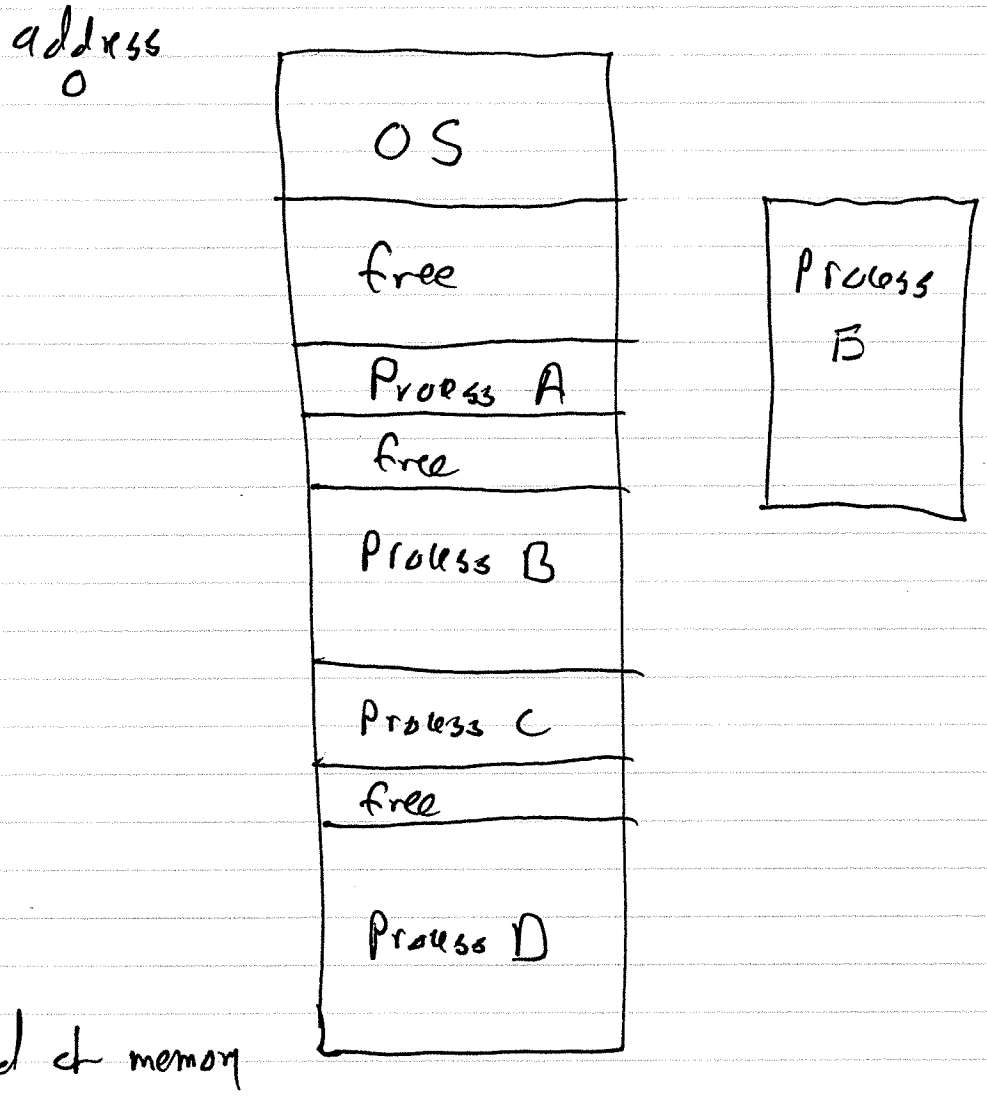
CS 411 Memory Management

A process' view of memory:



It "thinks" it "owns" all of memory and it "thinks" its address space begins at location 0

The reality:



Assume: - A process' memory is contiguous

- Different process' use different amounts of memory

- The OS places a process where it "fits"

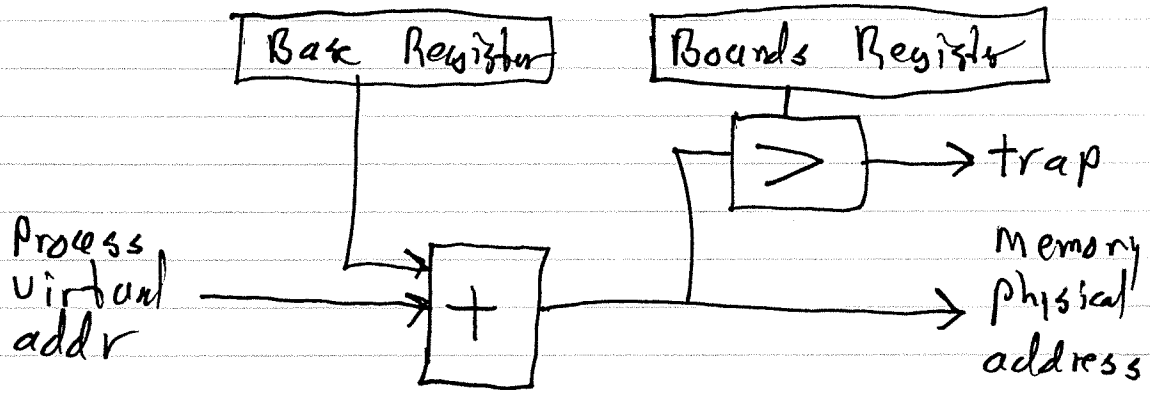
There's enough free space to run process B, but this space isn't contiguous.

To virtualize memory (make a process think its view is reality) we must:

- Translate a process' address (virtual) to memory's physical address
- prevent process' from "scribbling" on each others's memory
- Be able to coalesce free space
  → Relocate process'
- Be efficient

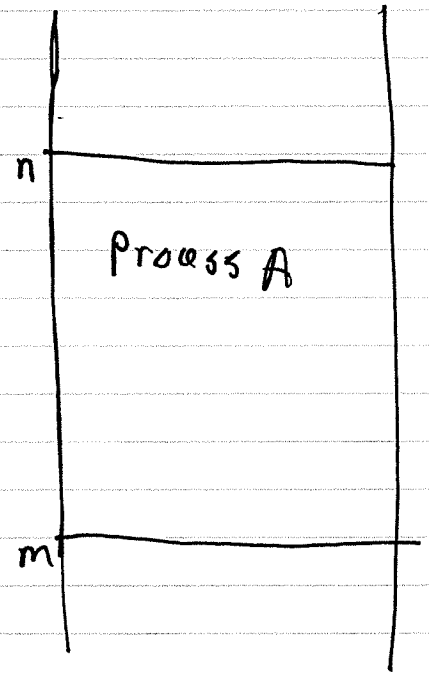
The simple solution:

Base + Bounds registers in the CPU and an adder (mmu)



Example:

Base Register is set to n and Bounds register is set to m when A runs



A is allocated n to m-1

Let's Say

$n = 1024$

$m = 2048$

Virtual address	Physical address	Trap
0	~~[illegible]~~ 1024	n
42	?	?
2100	?	?