

In `/proc/self/maps` (file associate with process)

1. Virtual start address -> text mapped into memory
2. Permissions (RWXP)
3. Virtual end address

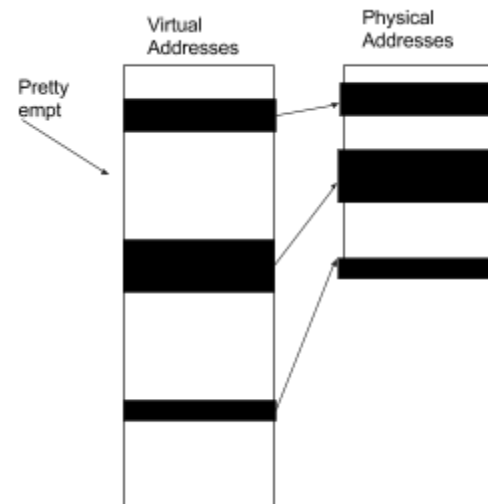
With this information you can recreate the exact moment in the code

Can you not ship the read only areas as they aren't needed for execution as long as you have original input? → cache - linker does symbol iteration - need everything because addresses dispatched before main is run - data can be changed before main executes and then permissions are changed.

Does it work if you shift the size of RAM → Addresses are virtual, they aren't the actual addresses, they map to the physical addresses

If you have all of the physical addresses used, checkpoint, and restore on a system with less physical memory, the OS will not crash, it will just place any overflow into SWAP on the hard drive (which causes problems because the access times are incredibly slow, but it works)

ONLY care about pointers as long as you create old pointers and old addresses you can restore, can't shift addresses as non-relative things (ex. stuff in lib C) will break, have to use same addresses.



OS looks at processes as:                      start →                      end  
we are looking at processes as:              start  
→ some middle state → end

## How to recreate checkpoint image?

`“/proc/self/maps”`

Start address inclusive

End address exclusive

(startAdd, endAdd - 1)

Went over generating checkpoint image and recreating it for the first homework, as outlined [here](#).

## System Calls

