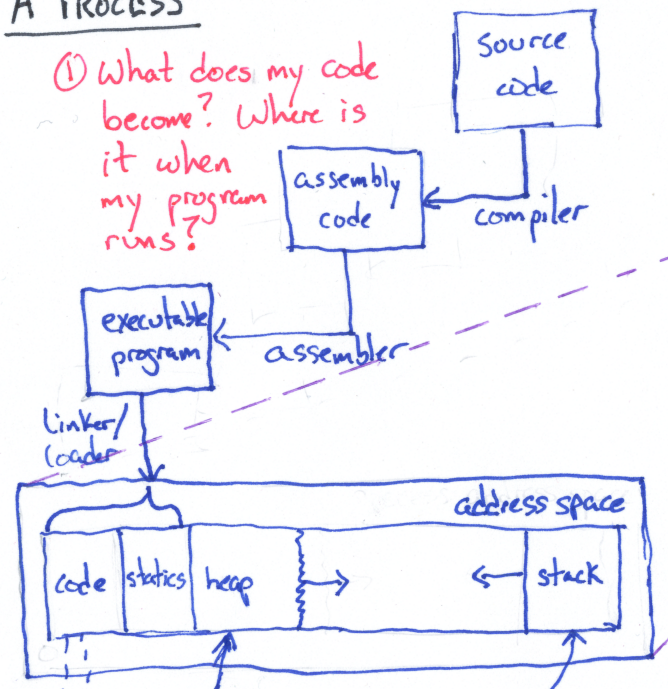
A Process



① What does my code become? Where is it when my program runs?

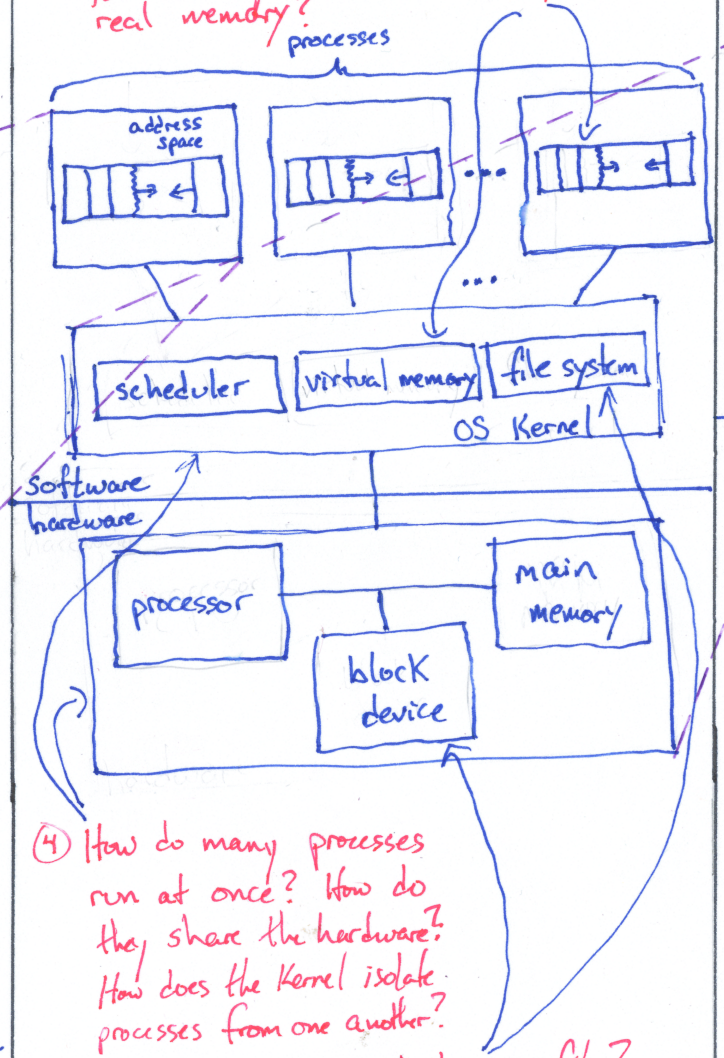
② How does my program do calls & returns and keep them ordered?

Compiled functions & calls create frames that represent each call.

Allocator functions manage dynamic memory requests (arrays & objects), organizing live & dead blocks of memory.

③ How is memory allocated? How is it returned when no longer needed? How does garbage collection work?

A System: Processes on a Kernel

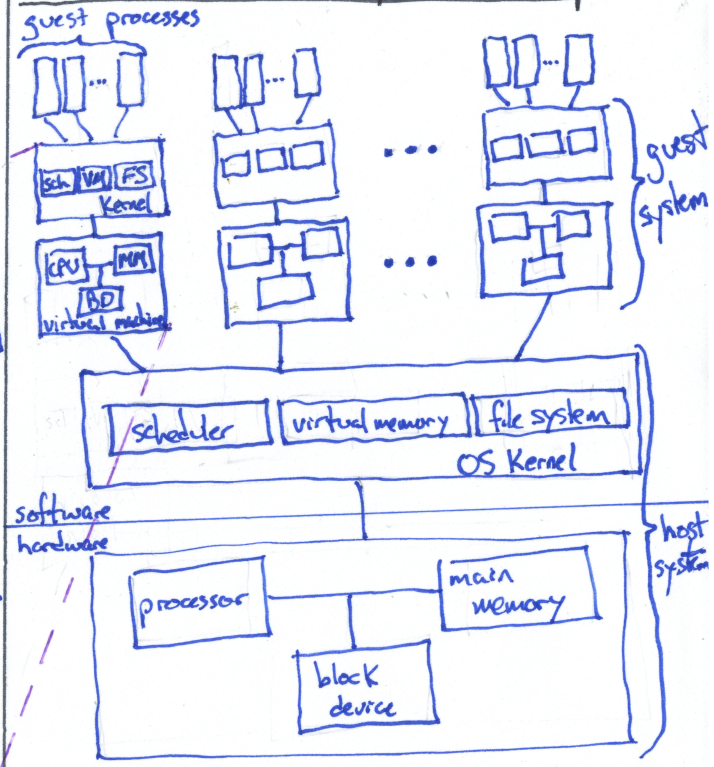


⑤ How can there be separate address spaces for each process? How are they allocated real memory?

④ How do many processes run at once? How do they share the hardware? How does the Kernel isolate processes from one another?

⑥ What is a file? How are they organized on a block device? How are they a general abstraction?

Virtual Machines: Systems within processes



⑦ How can a process fake being hardware to provide a virtual machine? How can one file be used to create a virtual block device? How can one system be shared by many virtual systems? How can any of this be efficient?

Key Concepts:

- Abstracting & implementing capabilities.
- Layering abstractions on abstractions.
- Digging through layered abstractions to find how they interact.

COSC-171: Computer Systems

Course Map (more or less)