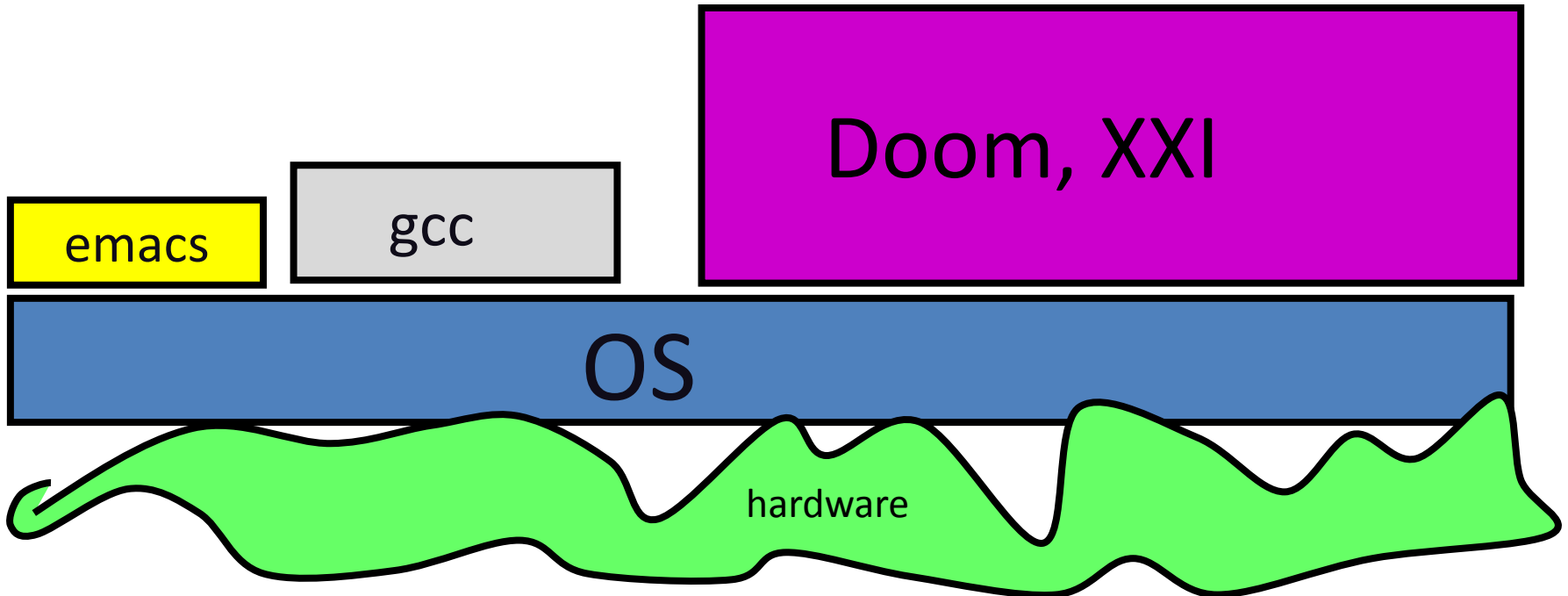


What is an OS?



- software between applications and reality:
 - abstracts hardware and makes portable
 - makes finite into (near)infinite
 - provides protection

OS evolution: step 0

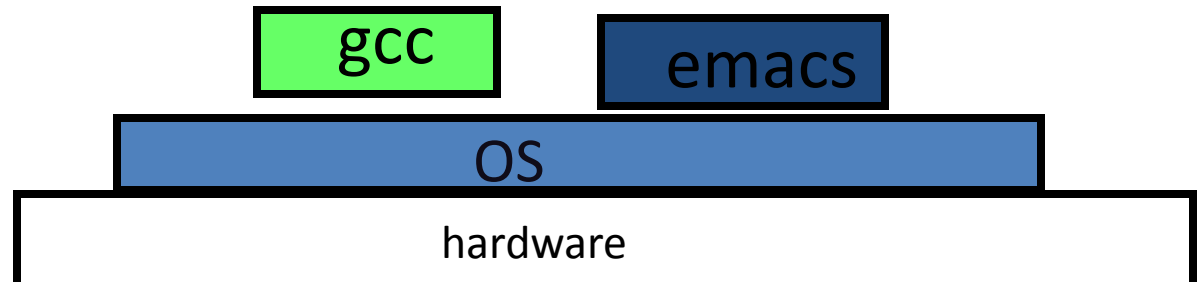
- Simple OS: One program, one user, one machine
 - Examples: early PCs, nintendo, cars, elevators, ...



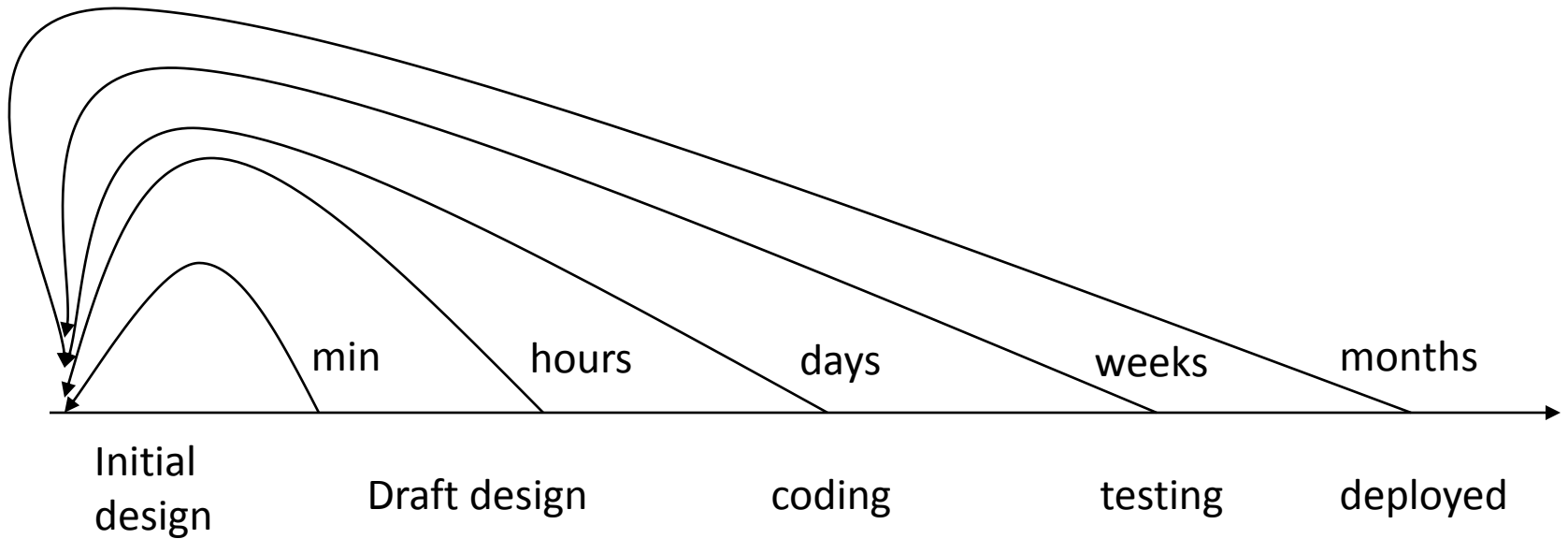
- OS just a library of standard services. Examples: standard device drivers, interrupt handlers, I/O.
- Non-problems: No bad people. No bad programs. A minimum number of complex interactions.
- Problem: poor utilization, expensive

OS evolution: step 1

- Simple OS is inefficient
 - If process is waiting for something, machine sits wasted.
- (Seemingly) Simple hack:
 - Run more than one process at once
 - When one process blocks, switch to another
- A couple of problems: what if a program
 - Infinite loops?
 - Starts randomly scribbling on memory?
- OS adds protection
 - + Interposition
 - + preemption
 - + privilege



The design loop



- Find flaws fast!