

* cloud computing

computing-as-a-service

models - pay-on-use
on-demand
elasticity

IaaS

PaaS

SaaS

FaaS

all of these need to be "provided"

with client requirements mechanisms.

② efficiency & costs features
③ abstractions / impl.
tools optimization & resource mgmt.

* IaaS → virtualization / VMs
Containers

② what is the VM abstraction?

~ what is an abstraction?
= logical entity w/ defined functionality and usage interface.

why abstractions?

- reuse, hide complexity
layering functionality

examples: process, API, ~~memory~~ address space, files
cars, bank accounts, switch

Programming Assignments

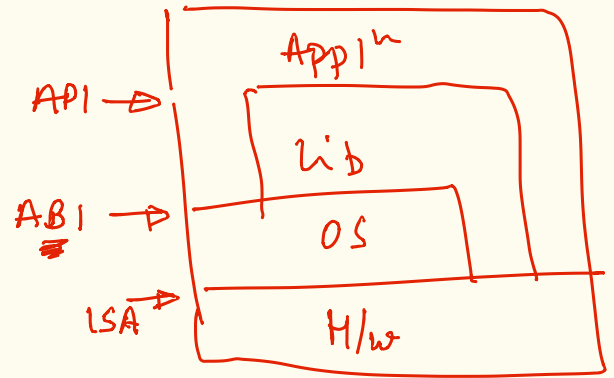
1 — flask

2 — linux peekaboo

3 — data structures
code
extensions
syscalls
LKM

Simple KVM based VM

transistors
gates
logic units
interconnects
ISA



ABI - Appⁿ binary interface
why binary?

drives OS

libraries

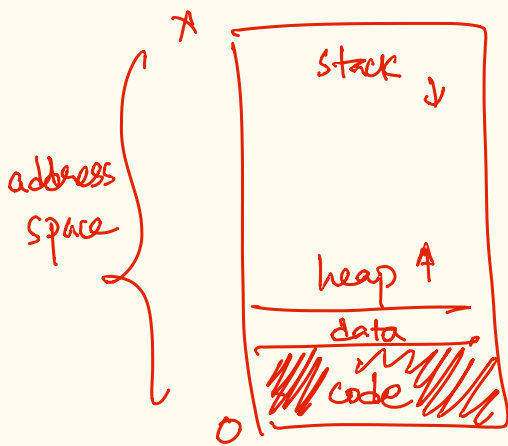
more libraries

applications!

process abstraction

decoupling of program & program instance
(binary)

- how does the OS support the process entity?



meta data

PCB - process control block

store info. about the process

pid

pgd - page directory

files

- allocates memory to store
program + runtime state

⊛ when a process executes what is CPU state?

PC/eip ~ program counter - ptr. to the next/current instⁿ.

SP/esp ~ top of the stack

↳ storing fn parameters (arguments) & return values

CR_x ~ control registers, e.g: CR3 points to the page table

eflags ~ flags register

eax/ebx/ecx/... ~ general purpose registers.

process context on the CPU.

⊛ save & restore process context enables multiplexing of processes ~ many processes can time share the CPU.

⊛ process-view

- ~ no other processes
- ~ CPU owner
- zero starting address space
- files, fs
- n/w endpoints

OS-view (system-view)

- CPU type
- ISA
- ~ disks, disk types
- physical addresses (NUMA 0, NUMA 1)
- ...

main-building blocks of an OS for providing abstractions?

(i) - interrupts / interrupt-driven execution

(ii) - (privileged) modes of execution (per instruction)

(iii) - the system call interface

OS is the sole owner of all resources!