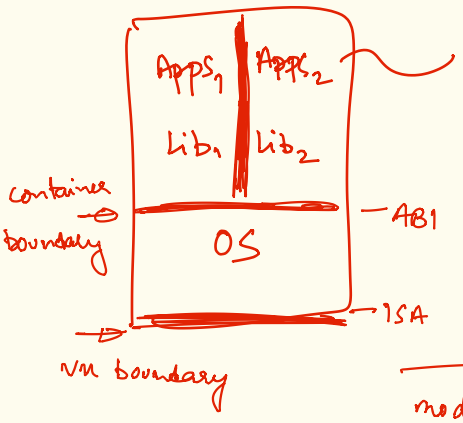
Lecture 12 CS 695

Containers



root file system
dependencies
utilities
libraries
OS-interface/support.

chroot
namespaces
cgroups
capabilities

modules isolation!
resources
runtime envs.

(i) chroot — changes root to a mount point & executes program.

chroot /home/user/temp bash
mount point

H.W
try w/ mv, ps, top

will work if all dependencies of an appl^n. are available relative to the new root for the program.

dependencies in container-world

* via image file — raw disk specification. w/ a file system wrapper.
  (docker world)
  lxc

(ii) namespaces

mechanism to provide isolated view of global resources.

atleast 6 diff. types of namespaces

via 3 system calls
unshare
setns
clone

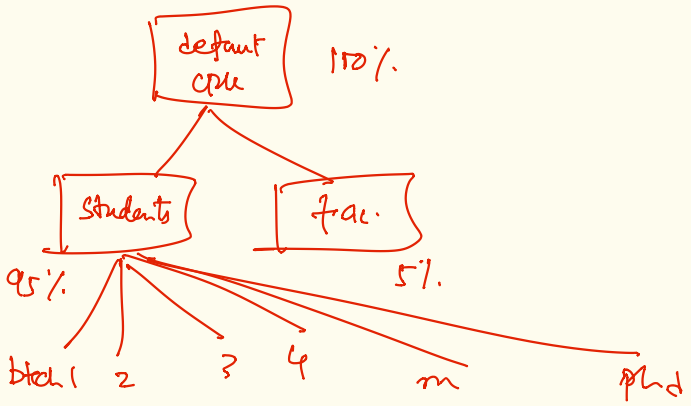
mnt — mount points of FS
pid — processes
net — n/w handling & state
ipc — ipc primitives, queues.
uts — domain/host name
user — uid/gid ranges/sets

* container-world
all elements of a container are in the same namespace!

(iii) cgroups — control groups
(resource control) (of processes)

- resource allocation & accounting & mgmt. at the ~~modest~~ granularity of a (group of processes)
- divide processes in groups (hierarchy of groups)
- attach resources to each node of the hierarchy.

linux: 12 (?) cgroups.
- cpu, memory, blk-io, cpusets, ...



the interface to interact with cgroups is via /sys/fs/cgroups