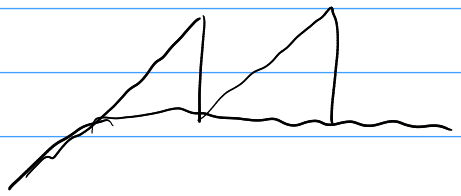
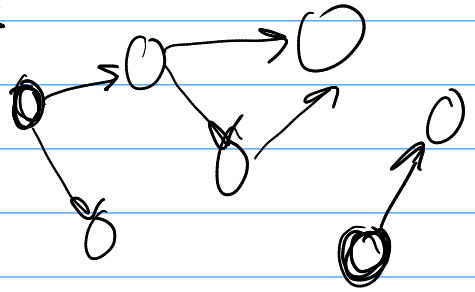
CS 3214

GC wrap up
virtual memory

recap: manage heap memory
malloc (manual mm)
{ automatic memory management
- garbage collection
- mark + sweep



Out of memory error
- heap size is too small
→ objects that are supposed to be dead are not cleaned up
"memory leak"



caching
≈
weak references

→ Boxed Integer int

"bloat"

GC for C/C++
{ "managed languages" knew where all pointers are
"precise GC"
don't know where all pointers are for sure
but we can guess
"conservative GC"
some compilers will try to emit code to make it more clear

Reference Counting

each obj keeps count of how many refs to it

get ↑ refcnt

put ↓ refcnt

if refcnt == 0

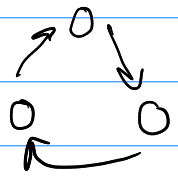
clean up object

programmer only thinks "locally"

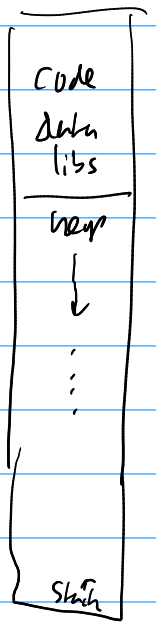
Automatic: Smart Pointers (C++, (Rust)) ownership

when ref created, assigned, copied, overwritten

< get / put >

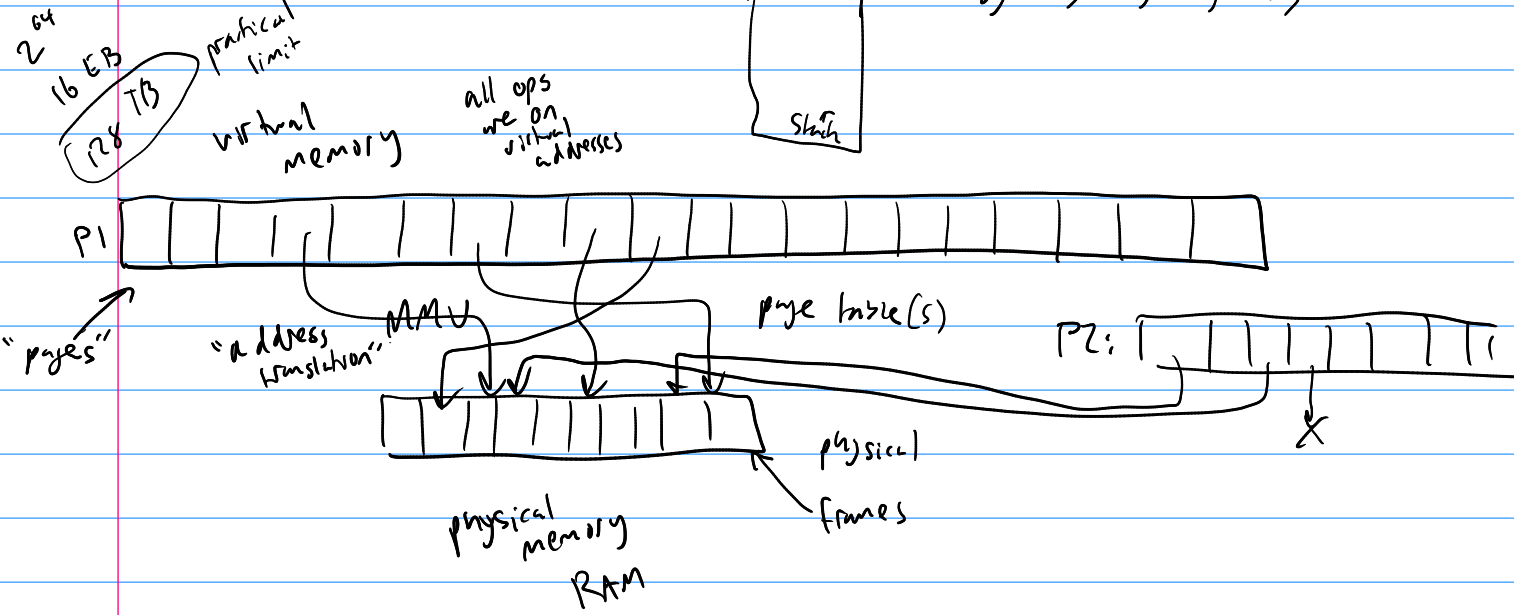


Virtual memory



2^{64} ~ 16 exabytes

KB, MB, GB, TB, PB, EB



Goals of virtual memory?

- virtualisation
  - process thinks it has its own memory
  - can use more than phys. machine has ←
- protection
  - no access to other process address space
  - kernel is protected from user
- security
  - address space layout randomization
  - sharing