CS 326-01 Process Shared Memory

project04 due Mon May 8th 11:59 pm

Kernel Free Pages Kalloc.c

Page Directies

Process Shared Memory

Kernel Free Pages

128 MB how many pages?

1 page = 4KB = $\boxed{2^{12}}$ = 4×2^{10} = 4096

1 KB = 2^{10}

2^{10} = 1 KB

2^{20} = 1 MB

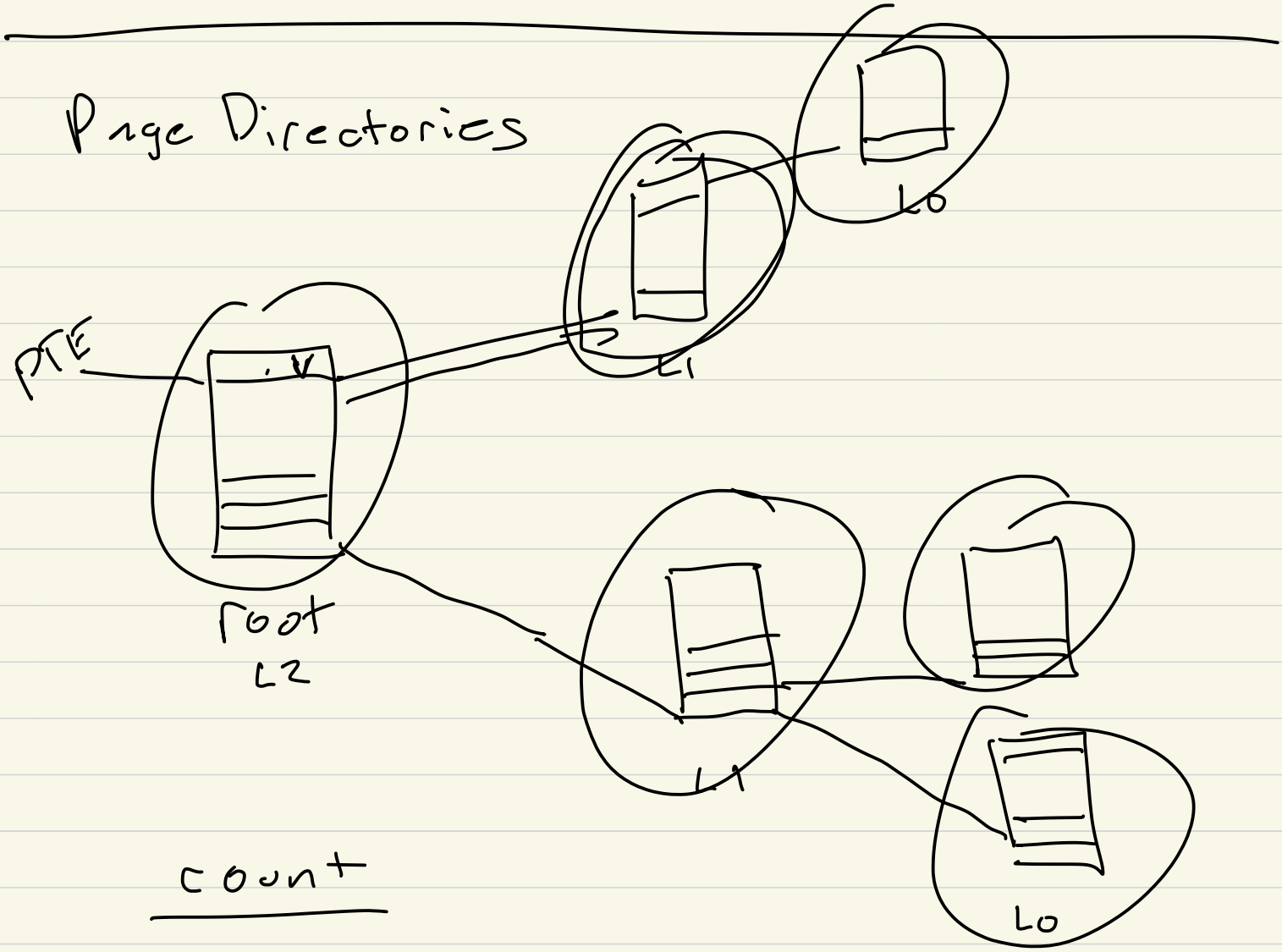
128×2^{20}

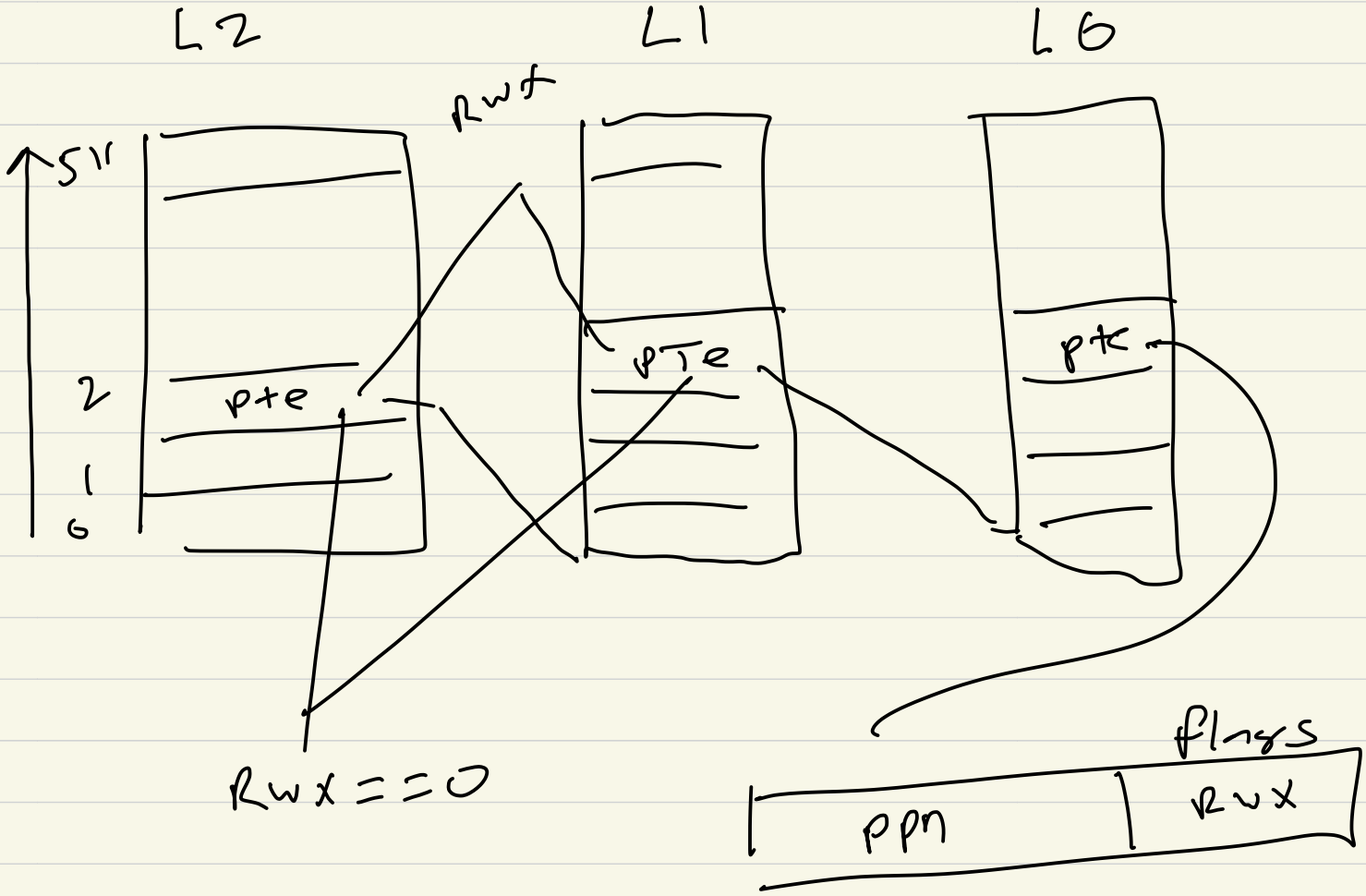
$2^7 \times 2^{20} = \boxed{2^{27}}$

$2^{27} / 2^{12} = 2^{15} = \boxed{32768}$ pages in 128 MB

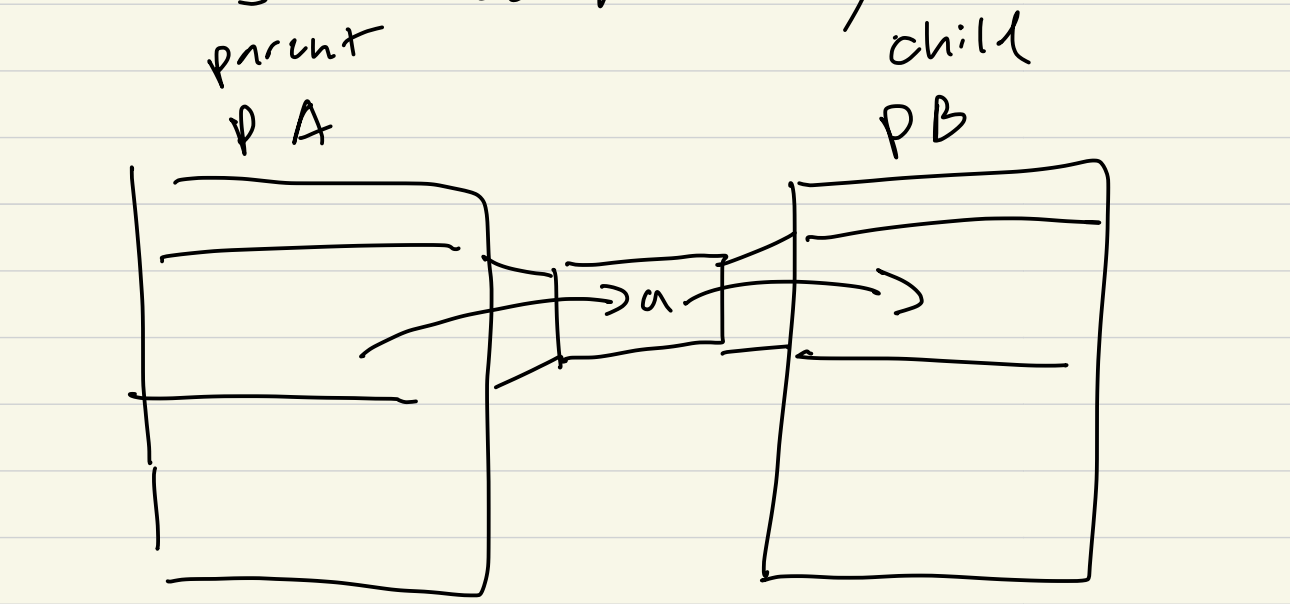
After startup

32768 - 32553 = 215 pages



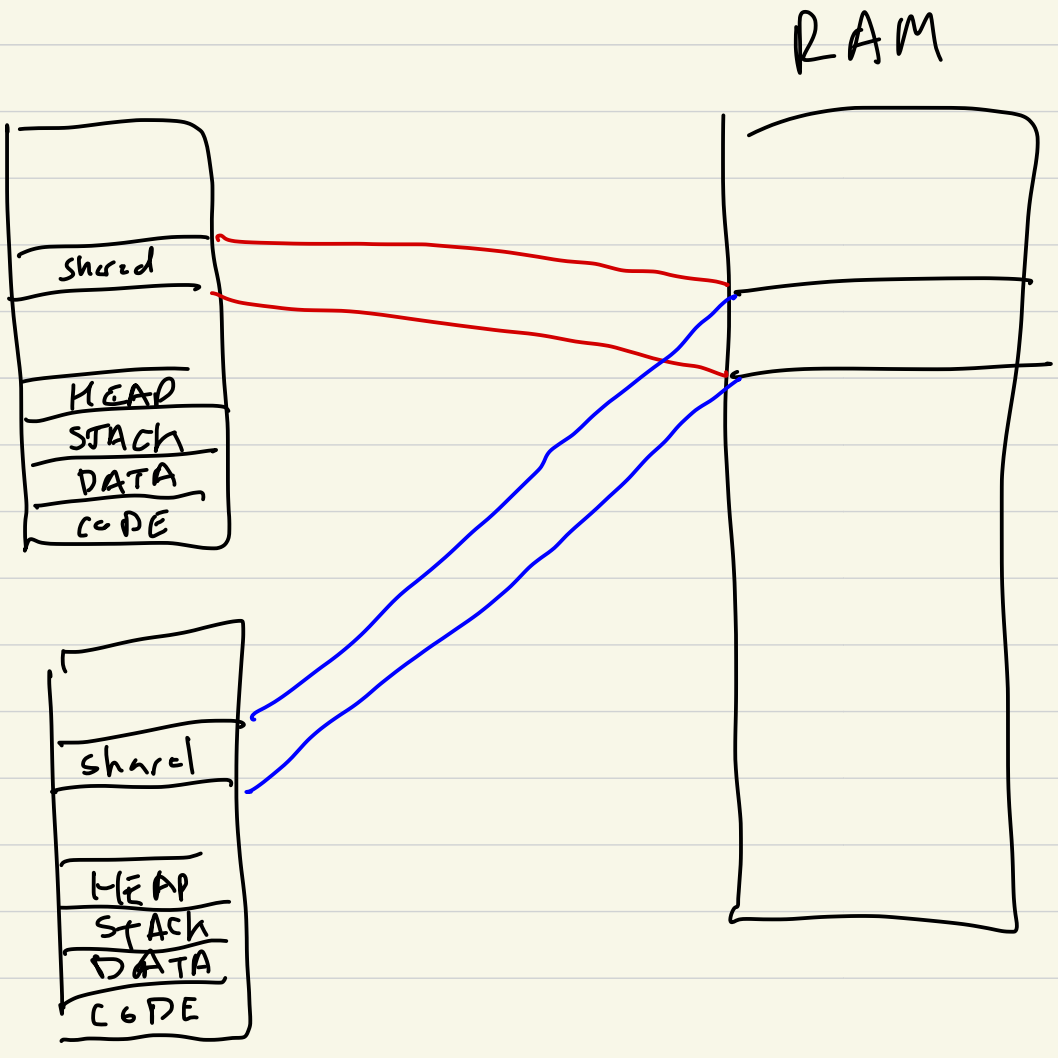

L2
L1
L0
RWX
SV
2
1
0
pte
pte
ptr
RWX==0
flags
ppn
RWX

Process Shared Memory



User

Kernel



Things to consider

syscall

smem(addr, size)

1. check that addr and size are multiples of 4096
2. allocate pages with Kalloc()
3. check pagetable for conflicts
4. mappages to addr in process's pagetable

Concept: a shared memory region will have an owner

In proc struct

- addr
- size
- owner_pid

---

fork()

if parent has a shared mem region then have child inherit the region

map same shared virtual address region in child's page table.

freeproc()

if proc has shared mem

if proc is owner

unmap and deallocate page

else

unmap