

Lecture 13: Inverted page tables

- Review
- Inverted page tables / hashed paging
- Quiz

Python:

```
class MyMonitor
    __init__():
        lock = Lock()
        cv1 = Condition(lock)
        cv2 = Condition(lock)
```

f():

with lock:

while not pred:
cv1.wait()

Java

```
class Object {
    priv. Lock lock = Lock()
    priv. Condition cv = Condition(lock)
}
```

```
class Foo /* ext. Object */ {
    synchronized void f() {
```

```
        this.notifyAll()
        while (!pred) {
            this.wait()
        }
    }
```

only one CV
per monitor:
need to
wake up all
threads when
any pred.
becomes
true

Matchmaker

```

class MM:
    __init__():
        self.lock = Lock()
        self.next_game = 0
        # num. of games that have started.
        self.total_players = 0
        # num. of threads that have
        # called join()
        # Invariant: next_game = total_players / 4
        # (rounded down)
        self.game_started = Condition(self.lock)
        # predicate: my_game < self.next_game
        #
    
```

```

# wait until this thread's game
# has started, then return.
def join(self):
    with self.lock:
        self.total_players += 1
        my_game = self.next_game
        self.next_game = self.total_players / 4
        self.game_started.notify_all()
        # wait until my game
        # has started.
        while not (my_game < self.next_game):
            self.game_started.wait()
        # know my game has started
        return.
    
```

Could do:

```

self.game_started = Condition(self.lock)
# invariant: game_started [next_game]
exists.
# predicate [i]: next_game > i
    
```

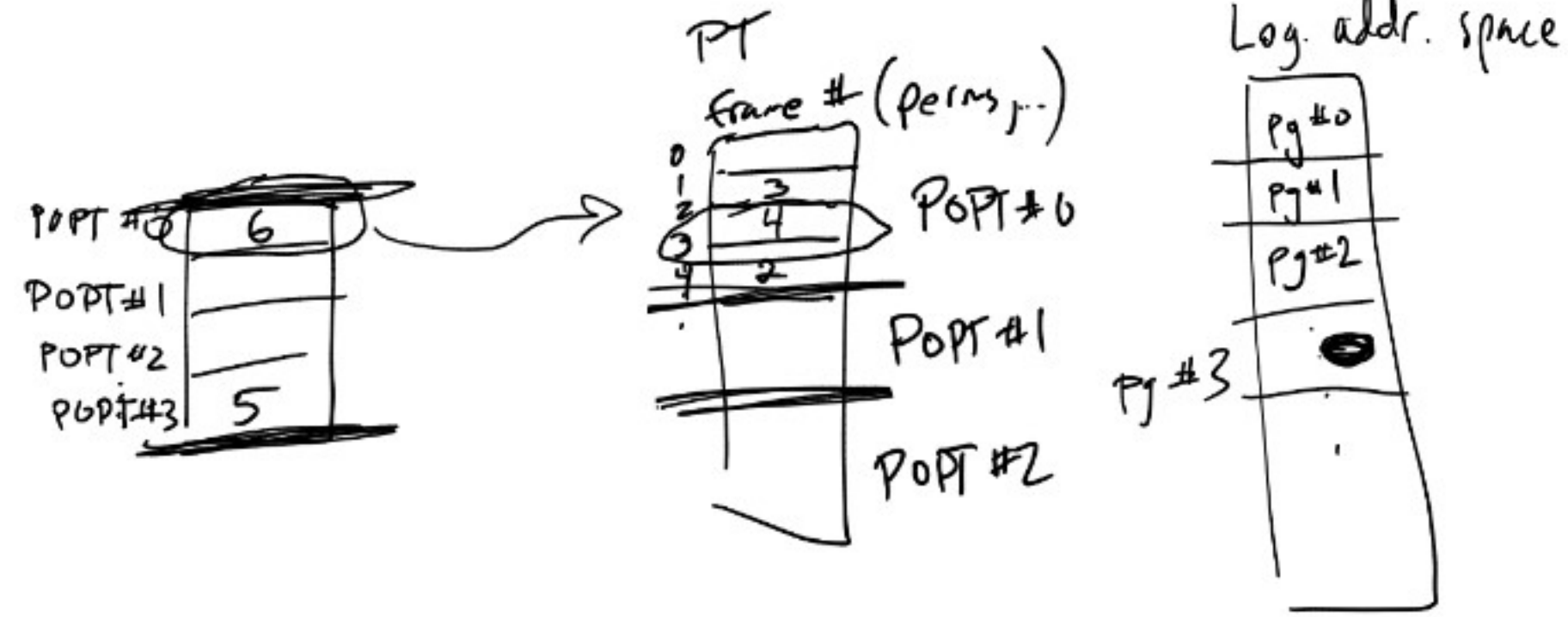
(overkill)
 instead: share 1
 CV,
 wake
 everyone

could optimize here
 (only not All
 if game actually starts)

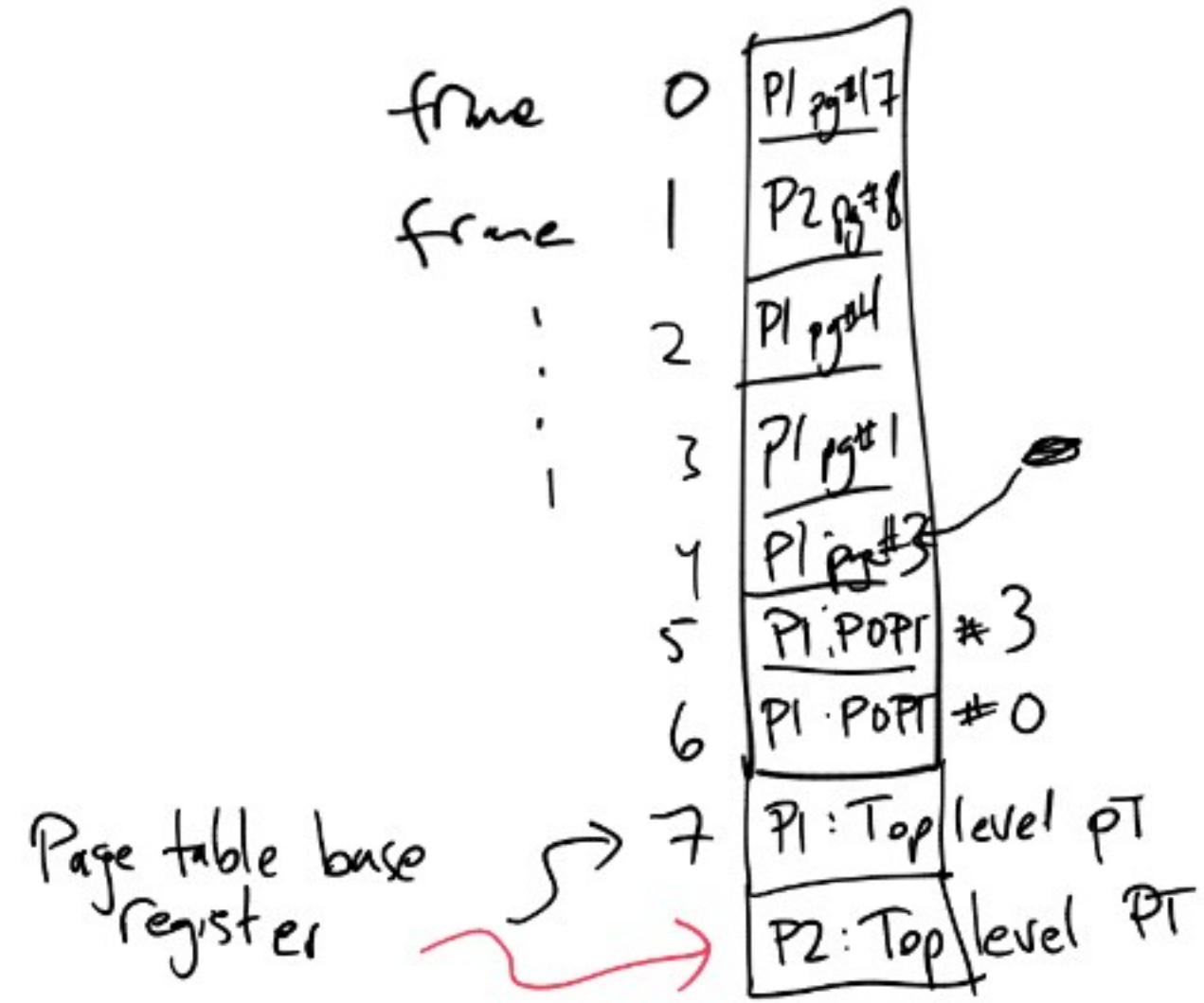
establish here.

after one thread leaves,
 condition is still true:
 other threads should go.
 (need not All())

Logic
(virtual)



Phys. Memory



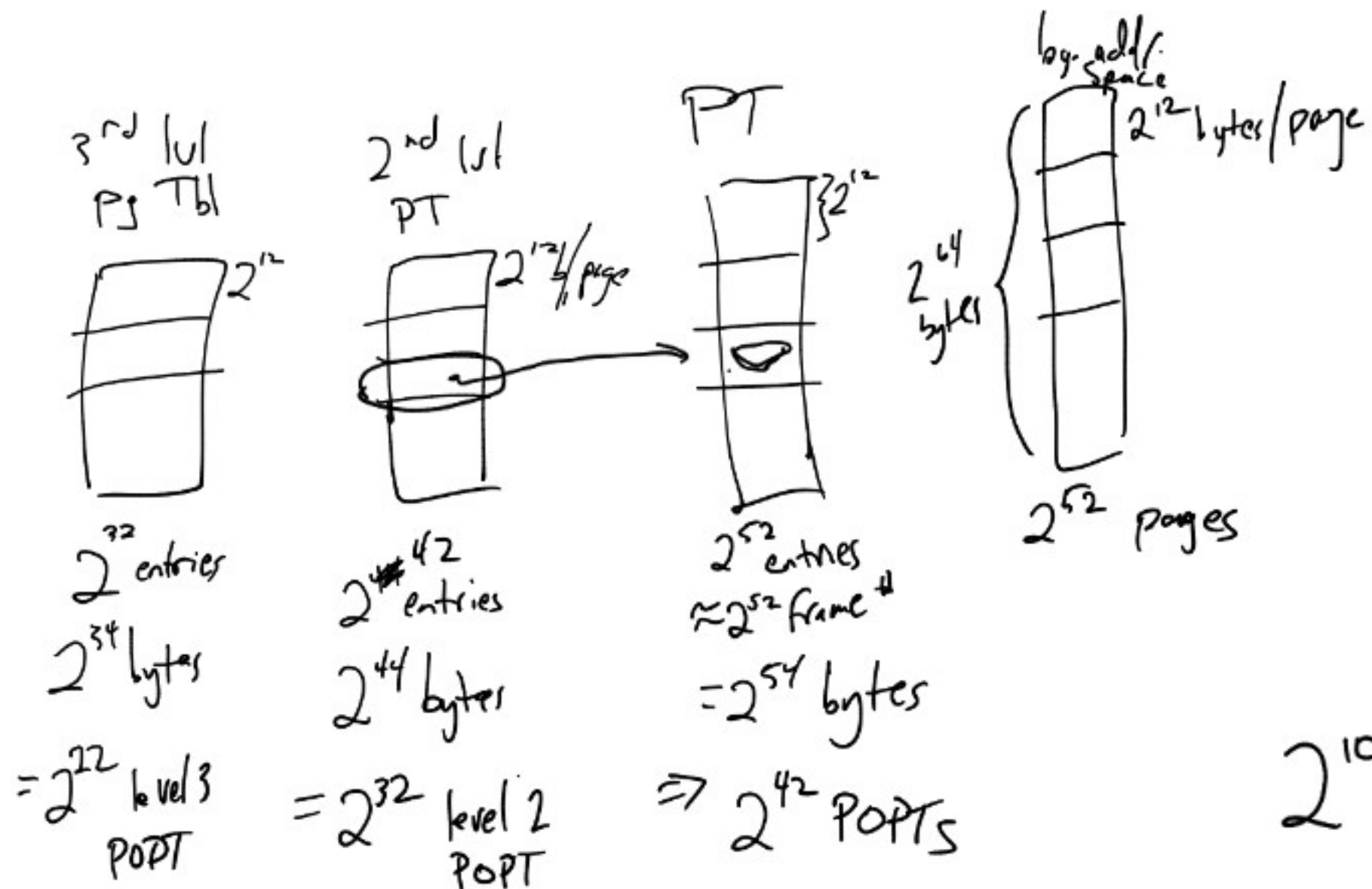
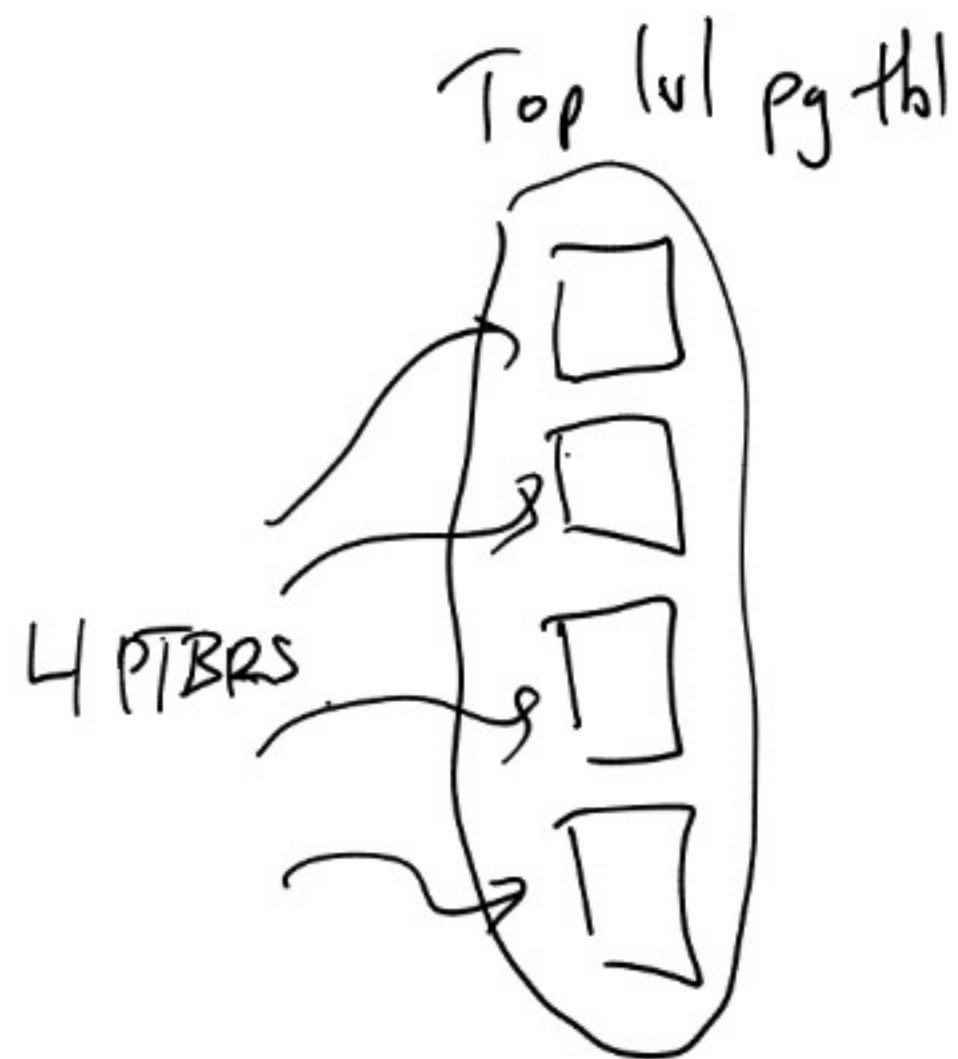
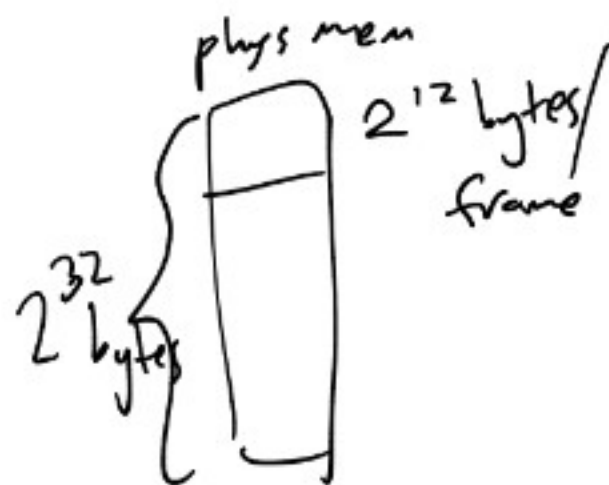
Big addr. spaces

Page size: 4kb

log. addr space: 64 bit (16,000 Petabytes)

phys. mem: 32 bit

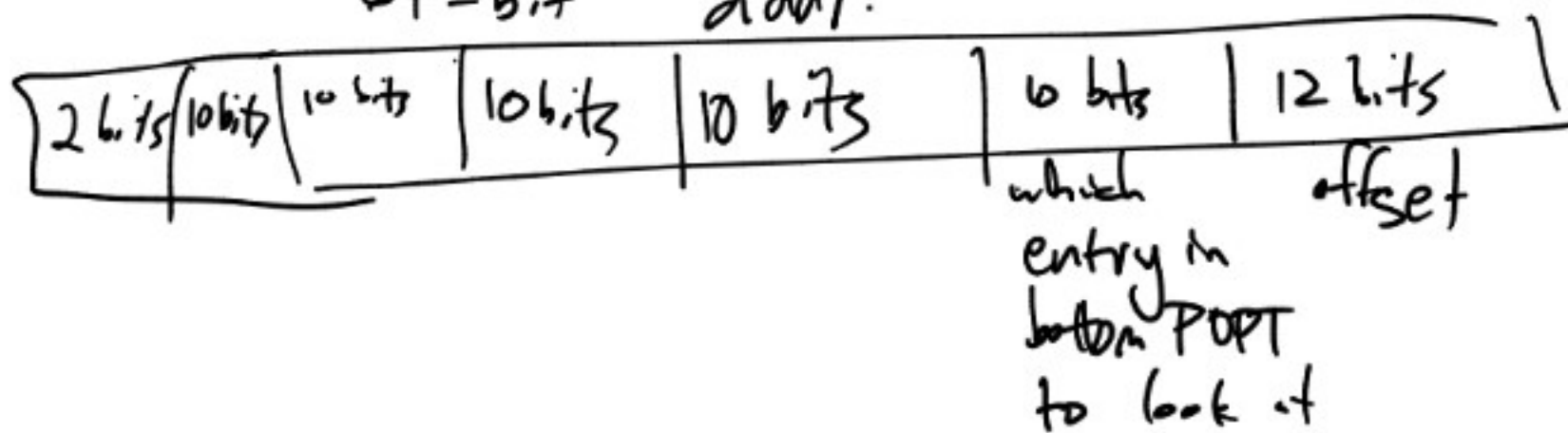
- 2^{10} K
- 2^{20} M
- 2^{30} G
- 2^{40} T
- 2^{50} P
- 2^{60}



20 bits for frame #
round to 4 bytes

2^{10} entries / POPT

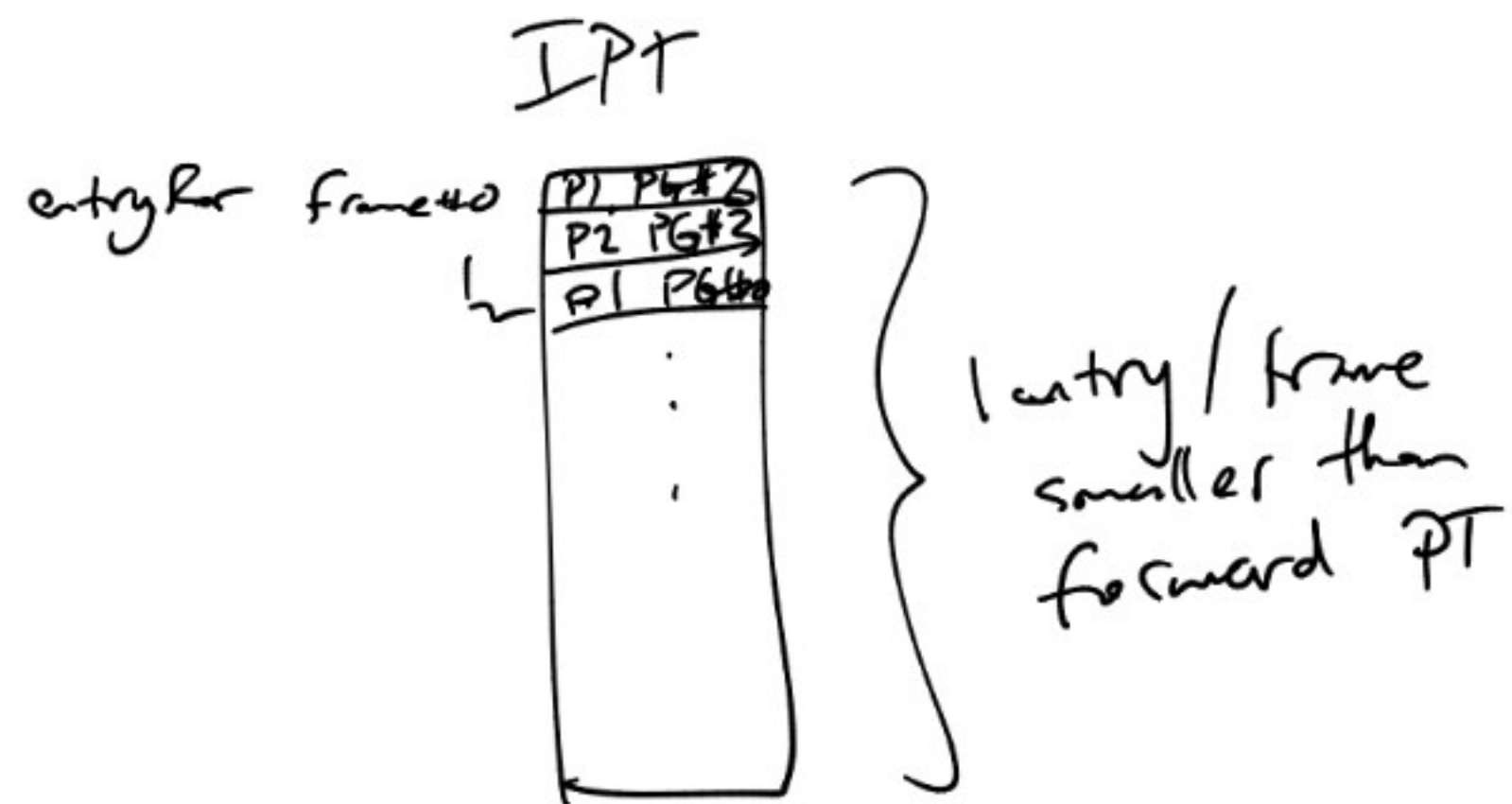
64-bit addr.



Notice: Many many more pages than frames. for 64-bit log. addr space

Page table: entry per page, telling us which frame it is in

Inverted page table: one entry per frame. tells us what page is in that frame.



Search?