Lecture 13: Inverted page tables

- Review
- Inverted page tables / hashed paging
- Quiz
Python:

```python
class MyMonitor:
    def __init__(self):
        self.lock = Lock()
        self.cv1 = Condition(self.lock)
        self.cv2 = Condition(self.lock)

    def f(self):
        with self.lock:
            while not self.pred:
                self.cv1.wait()
```

Java:

```java
class Object {
    private Lock lock = Lock();
    private Condition cv = Condition(lock);
}

class Foo extends Object {
    synchronized void f() {
        try {
            this.notifyAll();
            while (!this.pred)
                this.wait();
        }
    }
```
(multimeter)
Big addr space
Page size: 4K
Log addr space: 14 bits (16K pages)
Phys mem: 32K

$2^{32}$ bytes

$2^{30}$ frames

20 bits for frame # round to 4 bytes

$2^{10}$ entries / POPT

49-bit addr:

<table>
<thead>
<tr>
<th>12 bits</th>
<th>10 bits</th>
<th>10 bits</th>
<th>12 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 bits</td>
<td>10 bits</td>
<td>10 bits</td>
<td>12 bits</td>
</tr>
</tbody>
</table>

Entry in LUT: POPT to look up
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.

IPT

entry / frame smaller than forward PT

Search?