CS20F
paged virtual memory
- Pages
- What's a page table for?
- How do page tables work? (splitting up addresses)
- Multi-level page tables
- How much virtual space does a program really need?
How many virtual pages?

\[
\frac{2^{64} \text{ bytes}}{2^{16} \text{ bytes/page}} = 2^{48} \text{ pages}
\]
MMU's job: translate virtual pg # → physical pg #
Pretend page size = 64KB
= \(2^{16}\) B

0x0000 0000 ABCD 9876

Pretend

Phys page
5F
has virtual
page ABCD
MMU needs to translate

Page Table

Page Table Entry

Page # (ABCD)

48

16

+ some flags

5F
Page size: $2^{11}$

- Address
- Page Table
- Page
- Flags
- Physical Memory

Indexing an array
How big is the page table?

$2^{48}$ entries

Each entry

phys pg #

flags (Read-only
R/W
Shared
OS-only)
\[ \frac{1}{2^n-1} \text{ represents } \frac{1}{48.96\%} \text{ of } \frac{1}{255} \text{ of virtual memory} \]
pretend this table refers to pages of the stack

physical page#
another approach

map virt pg # 
  to phys pg #

hash table
Exam point

We're not using much VM
16 KB page

addr

PG #
48

byte offset

16

4 hexadecimal digits
Virtual memory is neato!