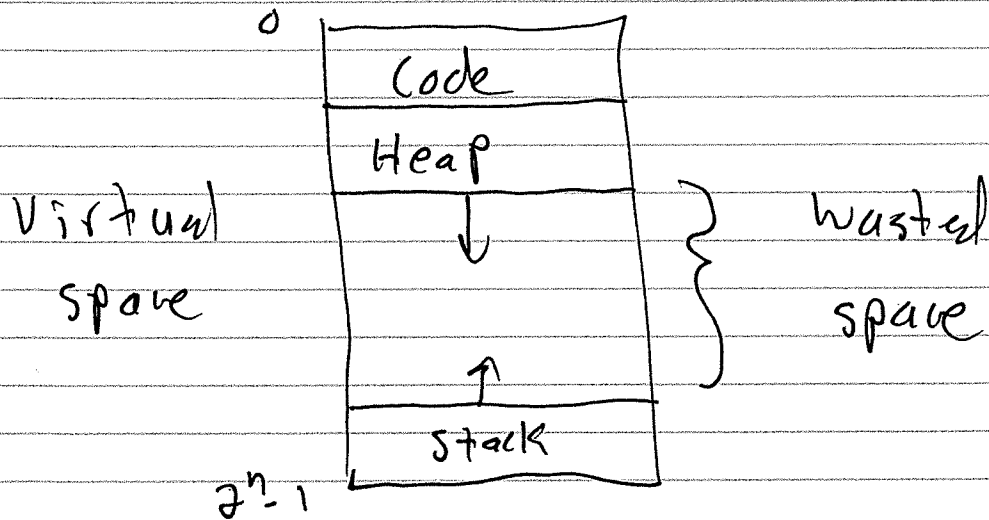


CS 411

Segmentation ~~* Passing~~

(1)

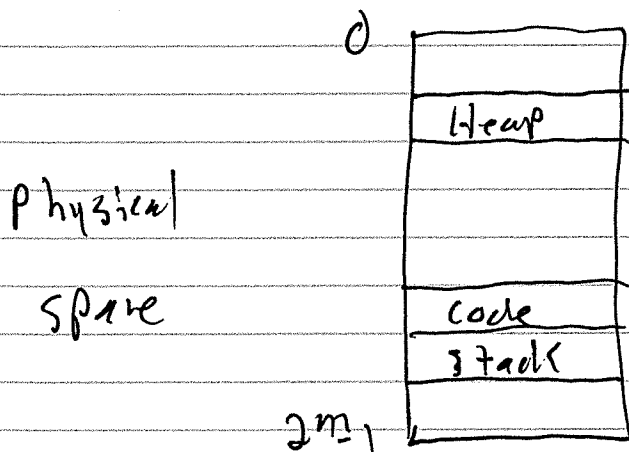
The problem with our previous memory model:



We have 3 segments:

Code Heap Stack

Let's split them, breaking up the single large address space



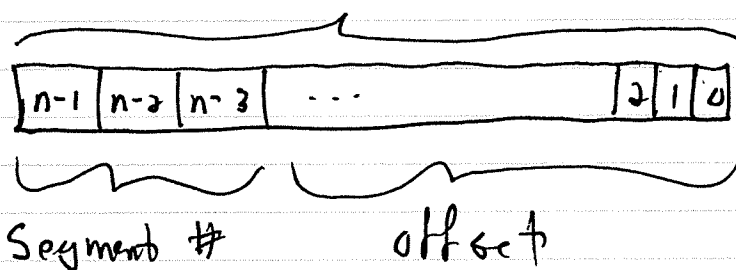
(2)

Assume virtual address space
is 2^n

Physical address space is 2^m

$$m \geq n$$

A virtual address is split into
2 parts Segment # and offset:



In this example we have

$$2^3 = 8 \text{ Segments}$$

The maximum size of a segment

$$\text{is } 2^{n-3}$$

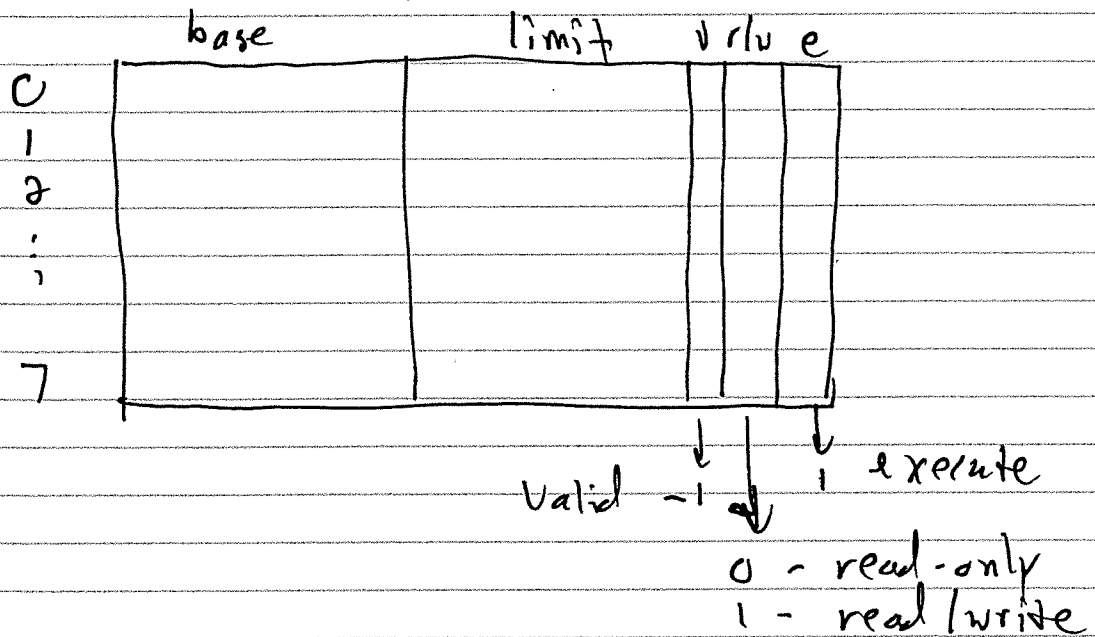
Why ≥ 3 segments

multiple code segments - Program, Library

For each segment we have:

- base + limit registers
- protection bits
 - read/write
 - valid
 - execute

Organise into a segment table:



How many bits for base + limit regs?

How should protection bits be set for code segment? stack segment?

What are the steps for translating + checking a virtual address?

Code segments can be shared
between processes. Yay!

Problem: we still have external
fragmentation

Consider a segmental system
where ~~the~~ a virtual address is
20 bits with the segment # field
occupying 4 bits.

Physical addresses are 28 bits

Which bits are used for the segment
field in a virtual address?

Which bits are used for the segment
offset?

How many segments are there?

What is the maximum size of a segment?

(5)

In the segment table, what is
the size in bits of
segment base
segment limit

Given a 16-bit virtual address
space with 4 segments,
a physical address space of 20 bits,
and this segment table:

Segment	base	limit	v	r/w	e
0	01000	0100	1	0	1
1	02000	3000	1	1	0
2	00E00	1000	0	1	1
3	0A000	0000	0	1	1

Translate each of the following
virtual addresses. If an exception
occurs, indicate why

Instruction fetch	0010
read	4F00
write	0000
write	7001
Read	80C0