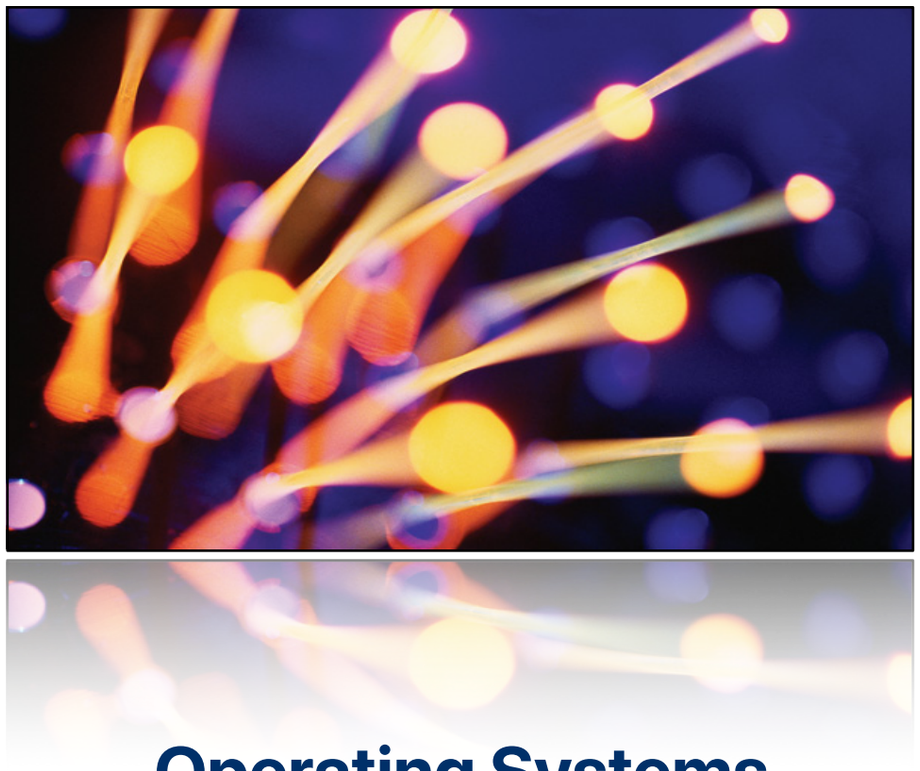
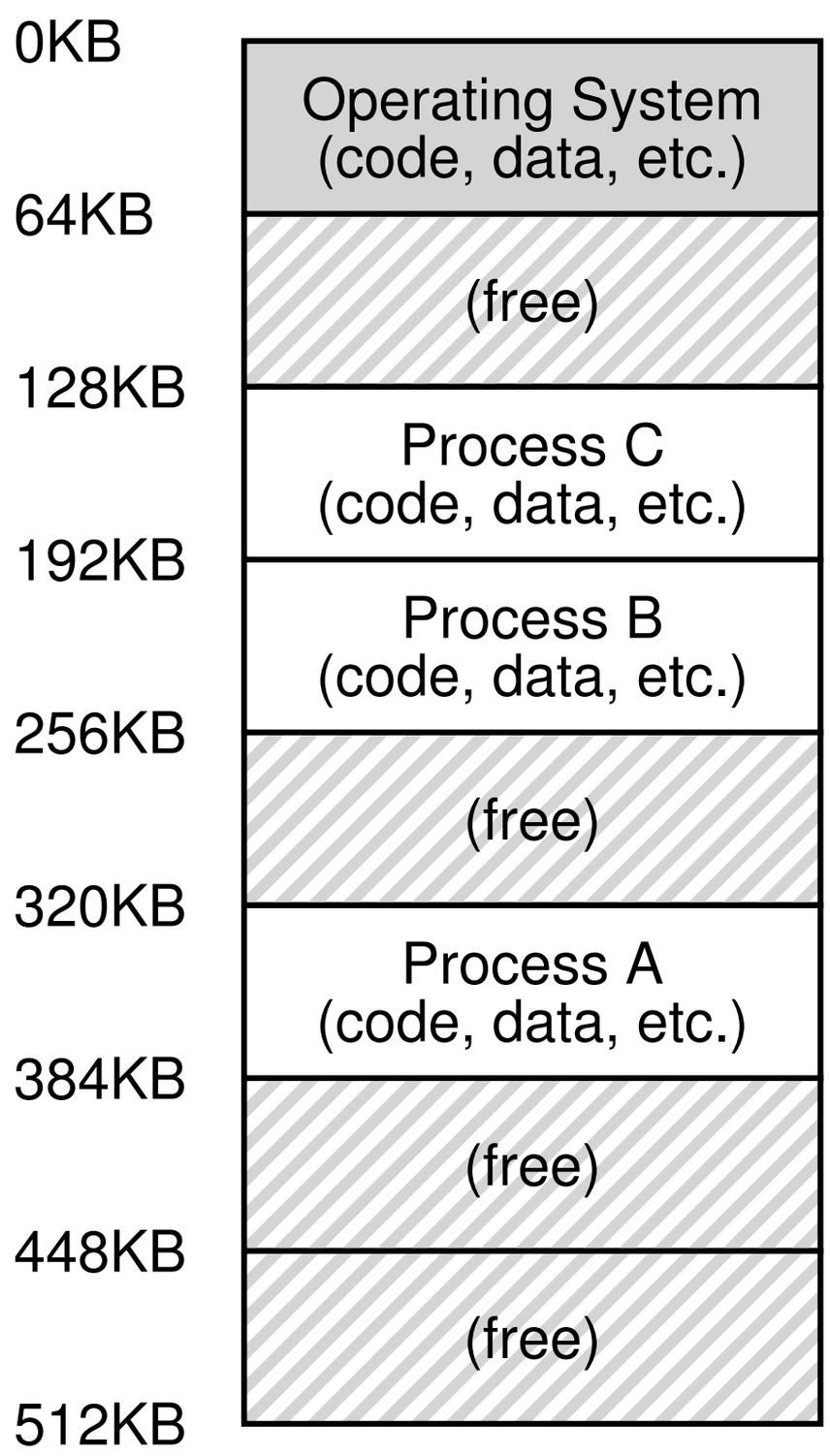
Segmentation



Operating Systems

Baochun Li
University of Toronto

Any problems with base-and-bounds virtualization?



Problems with base-and-bounds virtualization

Internal fragmentation: wasted space between the heap and the stack

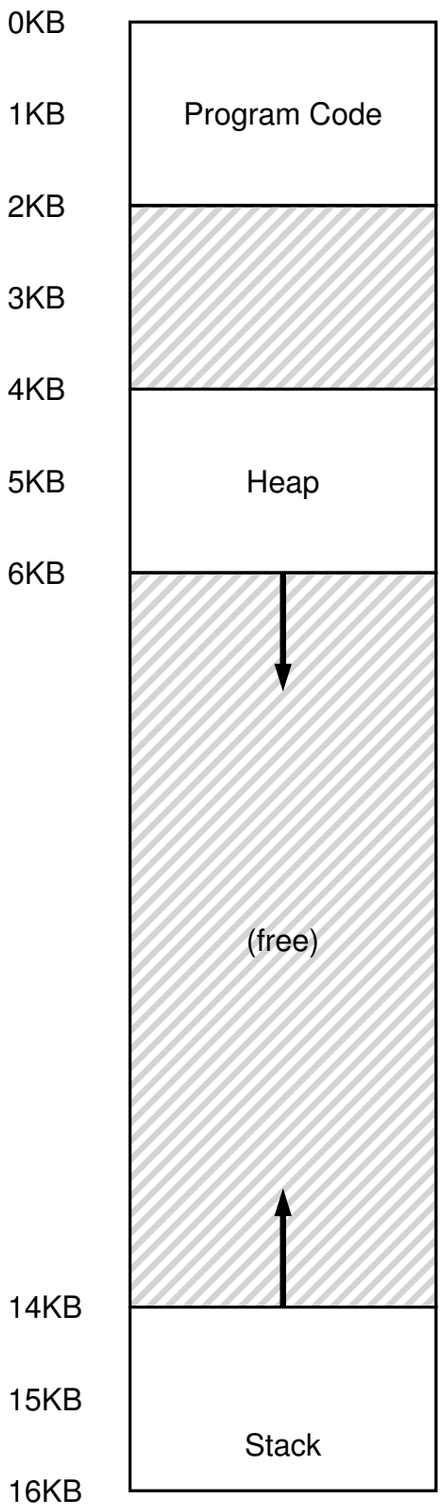
The address space is assumed to fit into the physical memory

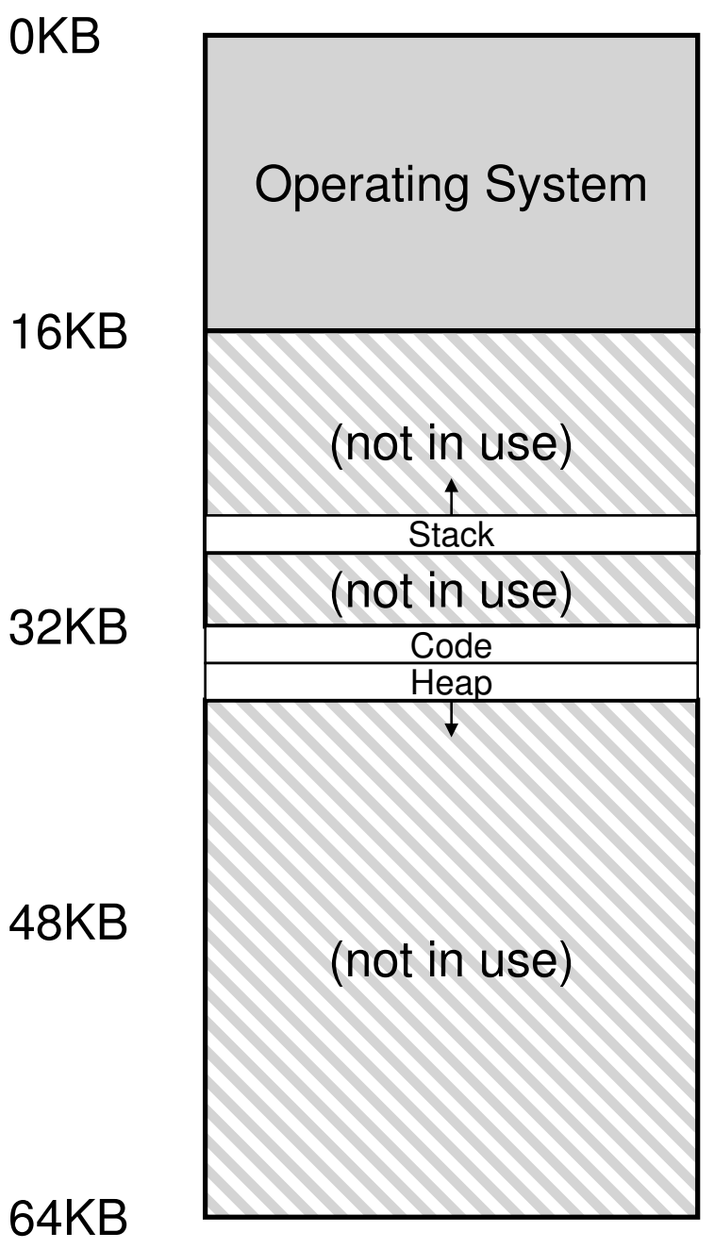
Segmentation

Basic idea: a pair of base and bounds register values for each logical segment in the address space

Accommodates sparse address spaces, with large amounts of unused address space between the heap and the stack

Example



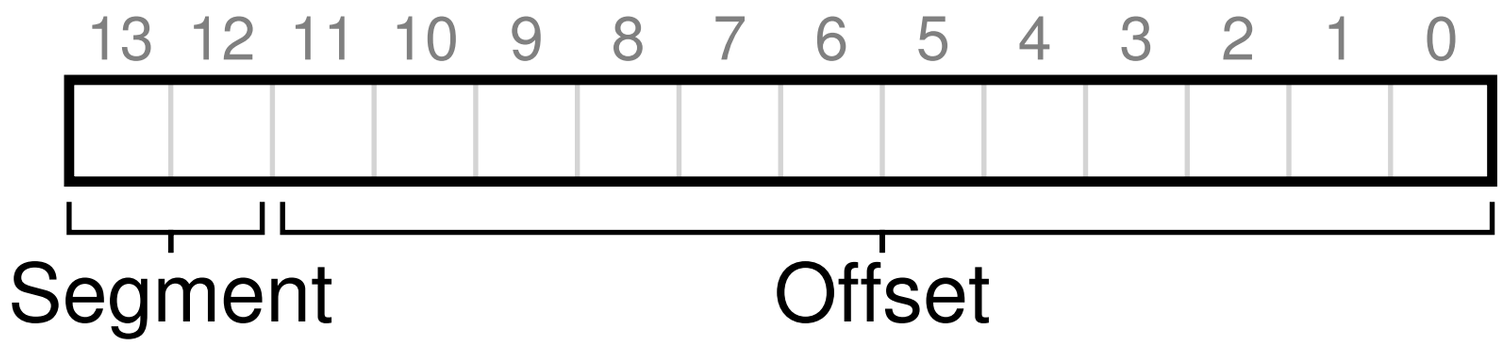


Segment	Base	Size
Code	32K	2K
Heap	34K	2K
Stack	28K	2K

Which segment are we referring to?

The hardware needs to know which segment an address belongs to, in order to compute the offset

An explicit approach: use the top several bits in the virtual address



Example translation

```
Segment = (Virtual Address & SEG_MASK) >> SEG_SHIFT
Offset = Virtual Address & OFFSET_MASK
if (Offset >= Bounds[Segment])
   RaiseException(SEGMENTATION_FAULT)
else
   PhysicalAddr = Base[Segment] + Offset
```

What about the stack segment?

The stack grows backwards to lower addresses

In addition to base and bounds values, the hardware also needs to know which way the segment grows

Segment	Base	Size	Grows Positive?
Code	32K	2K	1
Heap	34K	2K	1
Stack	28K	2K	0

Support for sharing

To conserve physical memory, sometimes it is useful to share certain memory segments between address spaces

Example: code sharing

To support sharing, add a few protection bits

Segment	Base	Size (max 4K)	Grows Positive?	Protection
Code_{00}	32K	2K	1	Read-Execute
Heap_{01}	34K	3K	1	Read-Write
Stack_{11}	28K	2K	0	Read-Write

OS Support

On a context switch, save and restore segment registers

What happens when a user process calls malloc() to allocate an object on the heap?

The heap segment may need to grow, by performing a system call — **sbrk()**

How do we manage free space in the physical memory?

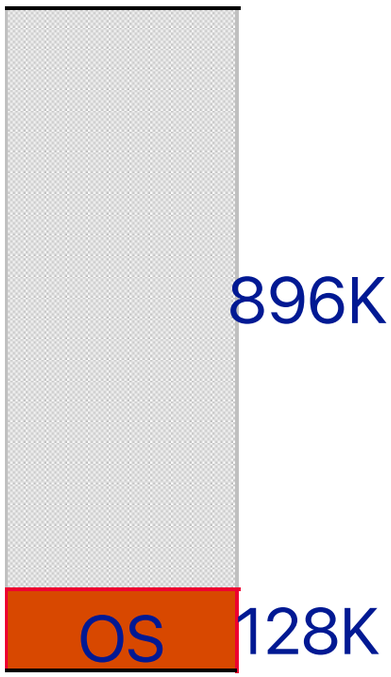
Segment sizes vary, rather than assuming the same size for address spaces

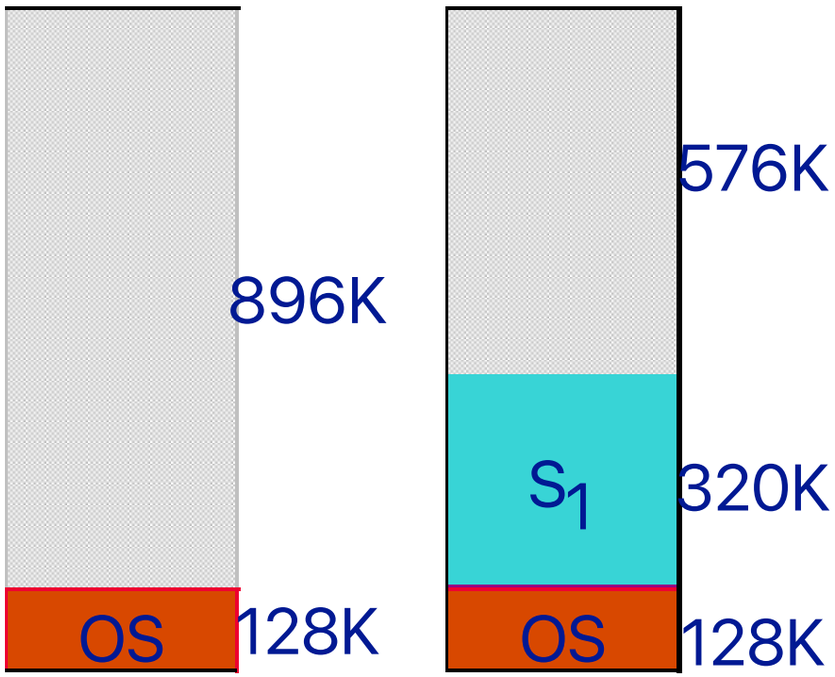
When a new address space is created

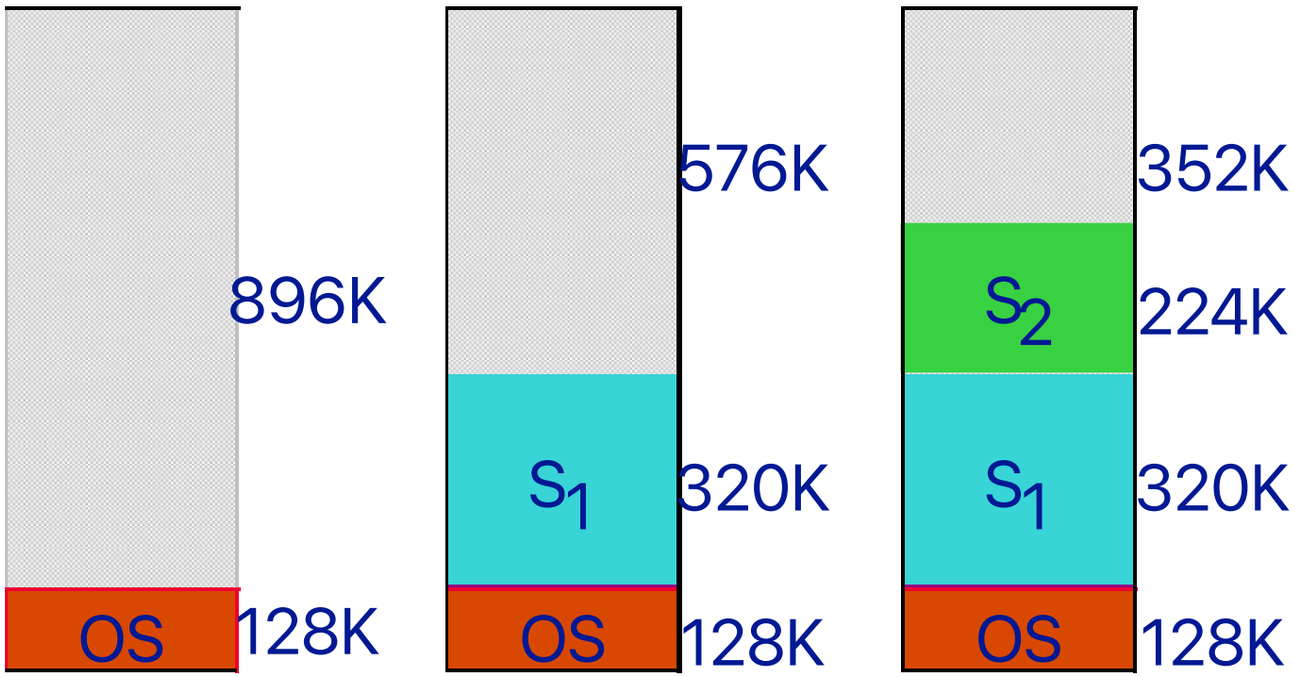
The OS needs to find space in the physical memory for its segments

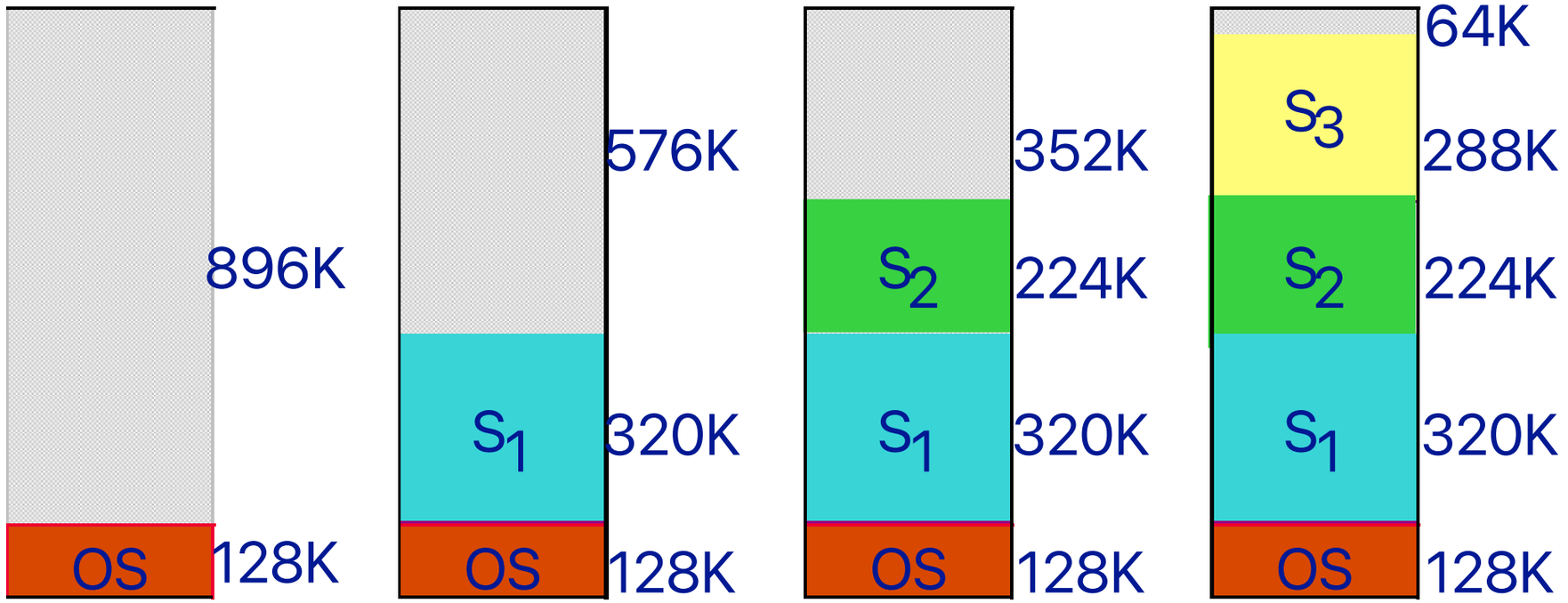
We now have a few segments in each address space, but they may have different sizes

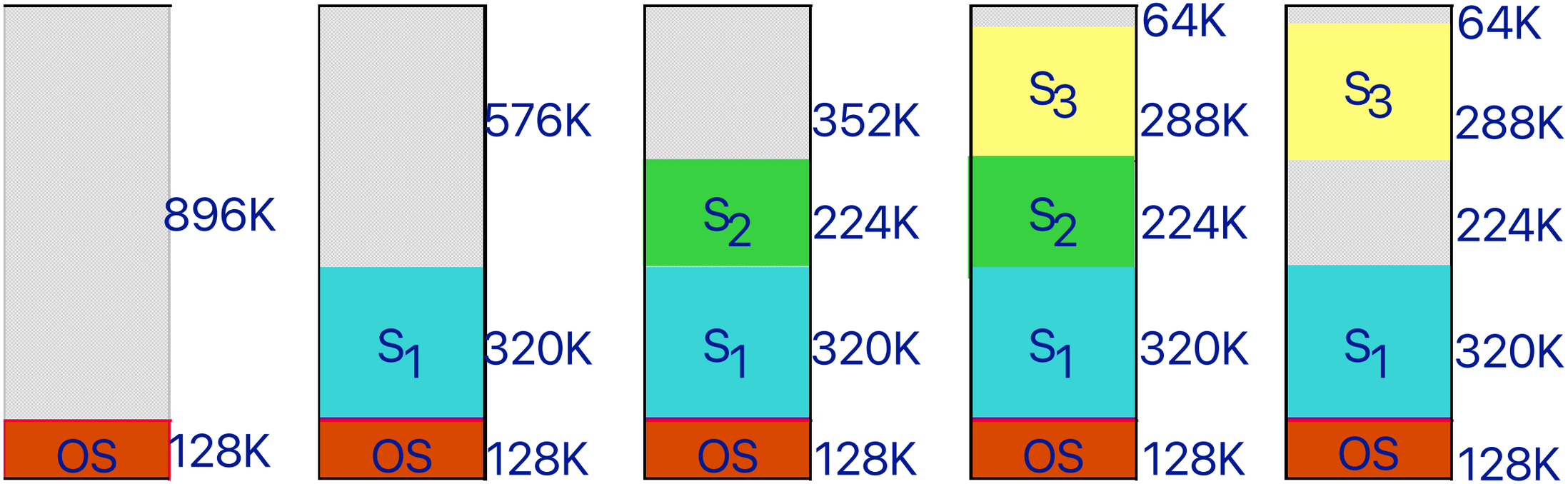
As address spaces are created and removed, the physical memory quickly becomes full of little holes of free space — called external fragmentation

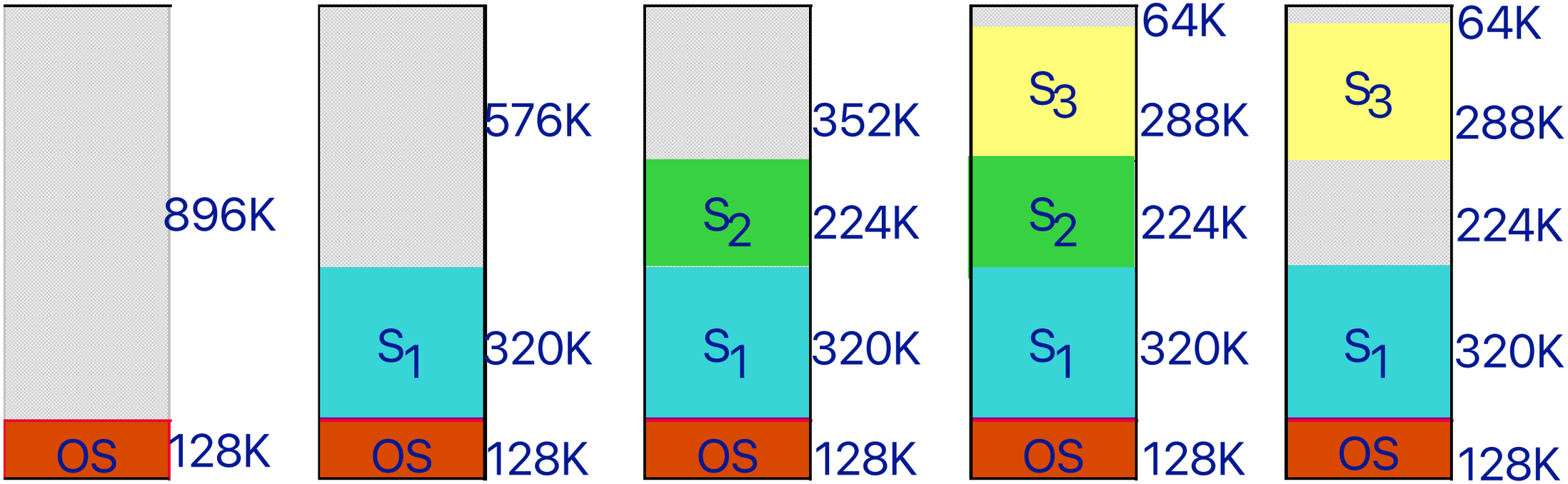

896K
OS
128K

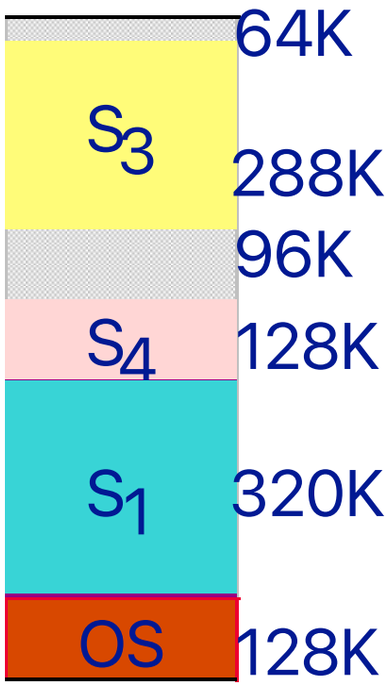

896K
OS
128K
576K
S1
320K
OS
128K

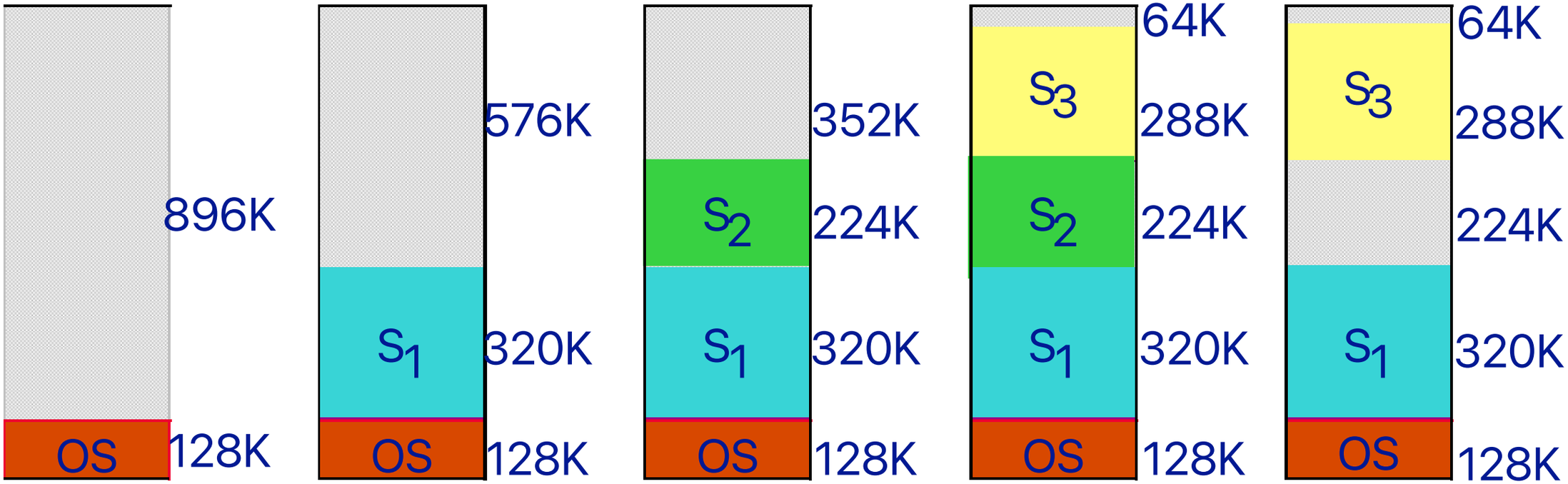

896K
OS
128K
576K
S1
320K
OS
128K
352K
S2
224K
S1
320K
OS
128K

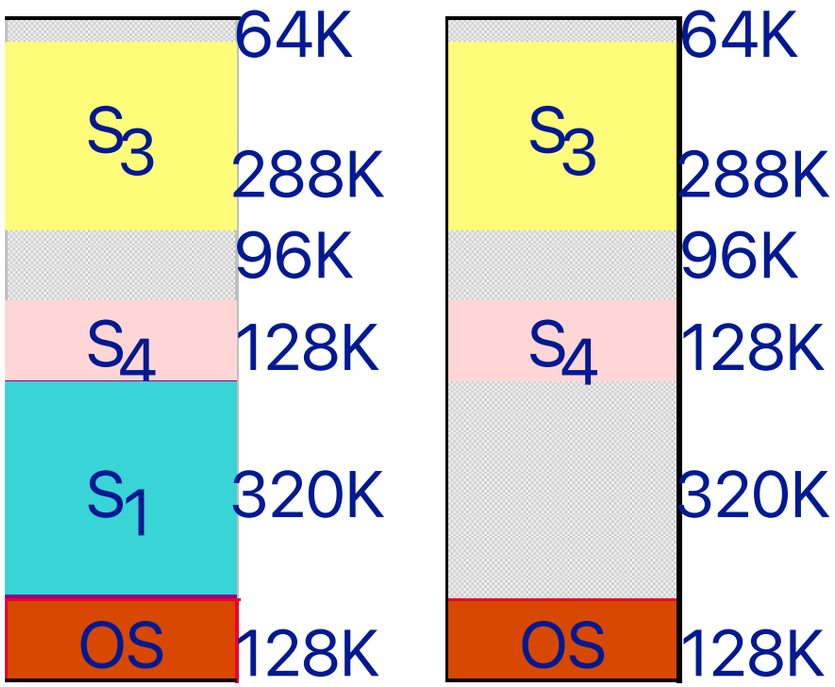

896K
OS
128K
576K
S1
320K
OS
128K
352K
S2
224K
S1
320K
OS
128K
64K
S3
288K
S2
224K
S1
320K
OS
128K

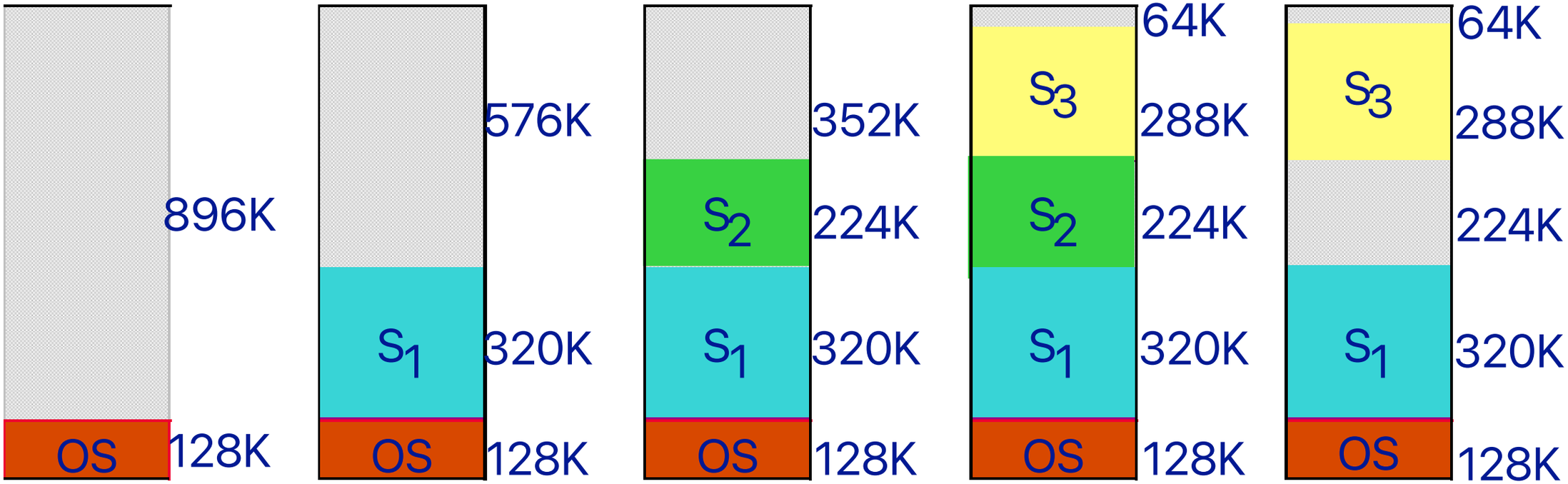

896K
OS
128K
576K
S1
320K
OS
128K
352K
S2
224K
S1
320K
OS
128K
64K
S3
288K
S2
224K
S1
320K
OS
128K
64K
S3
288K
224K
S1
320K
OS
128K

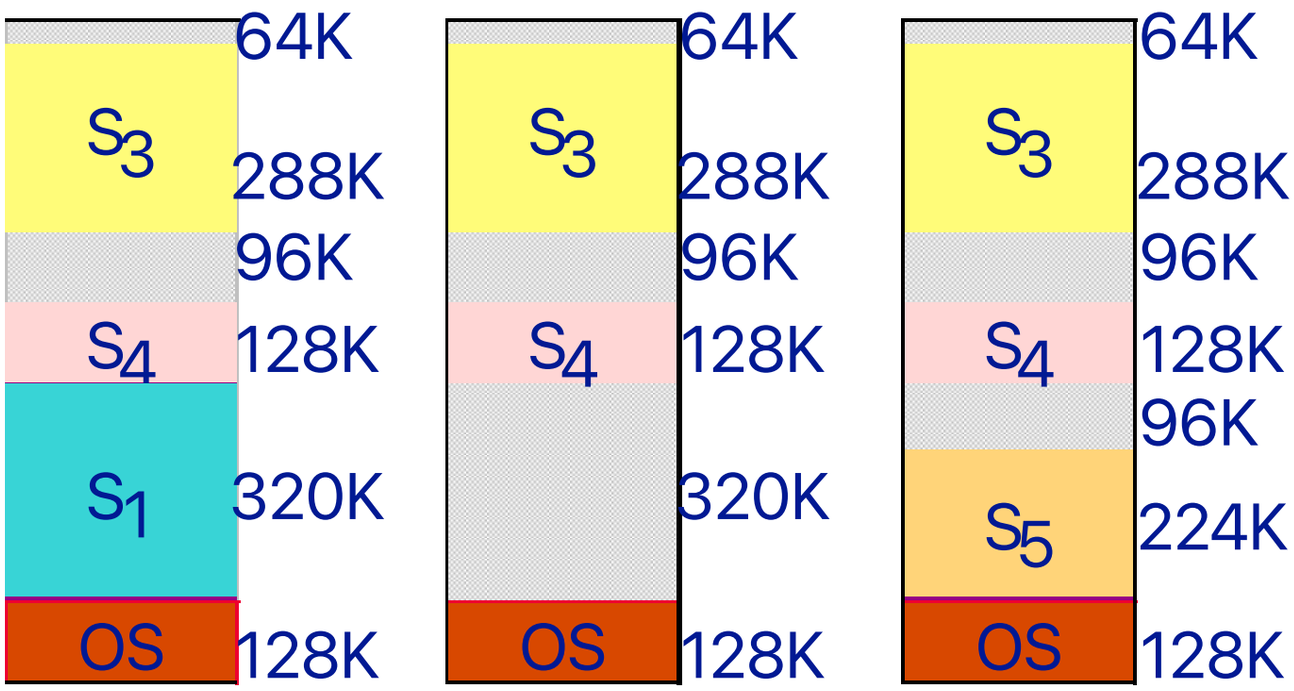

896K
OS
128K
576K
S1
320K
OS
128K
352K
S2
224K
S1
320K
OS
128K
64K
S3
288K
S2
224K
S1
320K
OS
128K
64K
S3
288K
224K
S1
320K
OS
128K

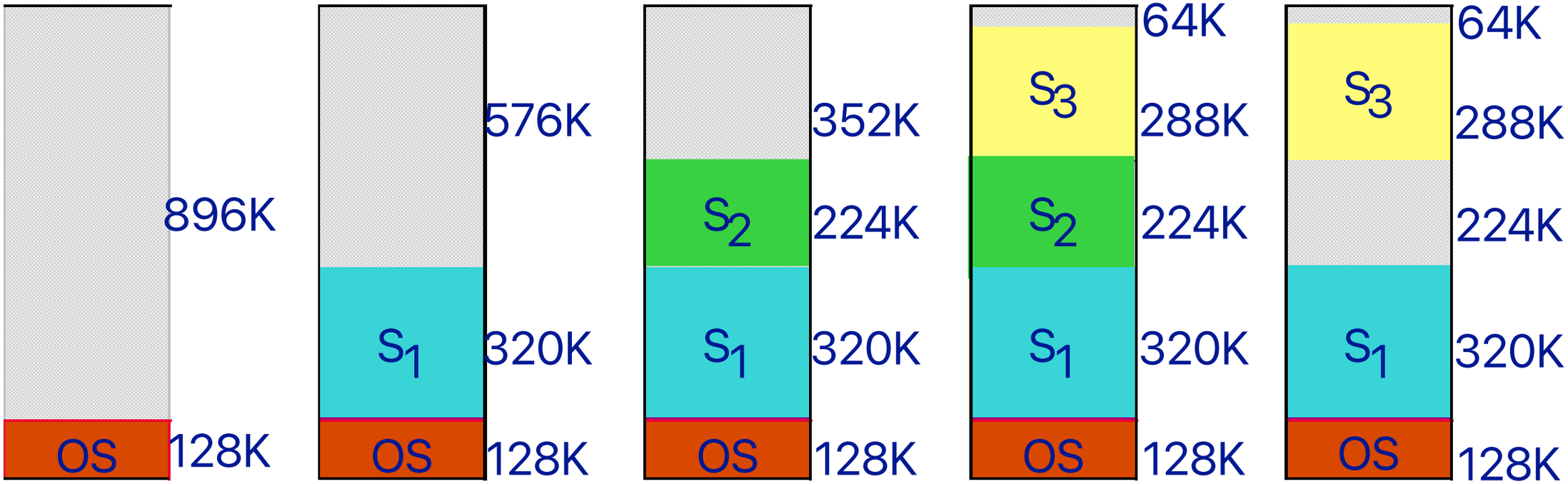

64K
S3
288K
96K
S4
128K
S1
320K
OS
128K

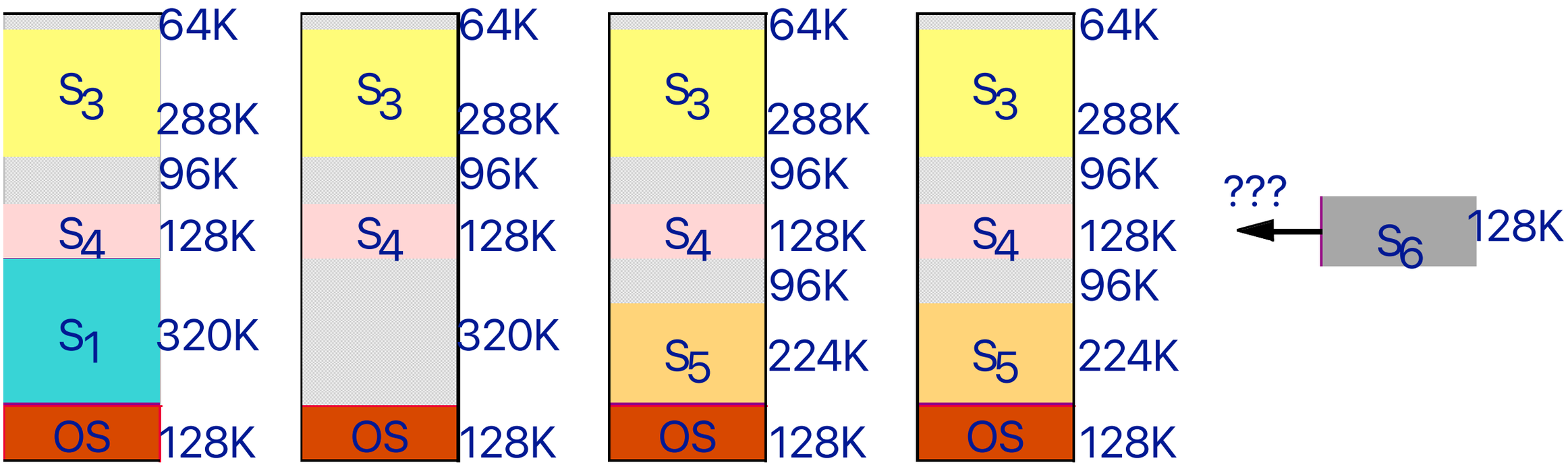

896K
OS
128K
576K
S1
320K
OS
128K
352K
S2
224K
S1
320K
OS
128K
64K
S3
288K
S2
224K
S1
320K
OS
128K
64K
S3
288K
224K
S1
320K
OS
128K


64K
S3
288K
96K
S4
128K
S1
320K
OS
128K
64K
S3
288K
96K
S4
128K
320K
OS
128K


896K
OS
128K
576K
S1
320K
OS
128K
352K
S2
224K
S1
320K
OS
128K
64K
S3
288K
S2
224K
S1
320K
OS
128K
64K
S3
288K
224K
S1
320K
OS
128K

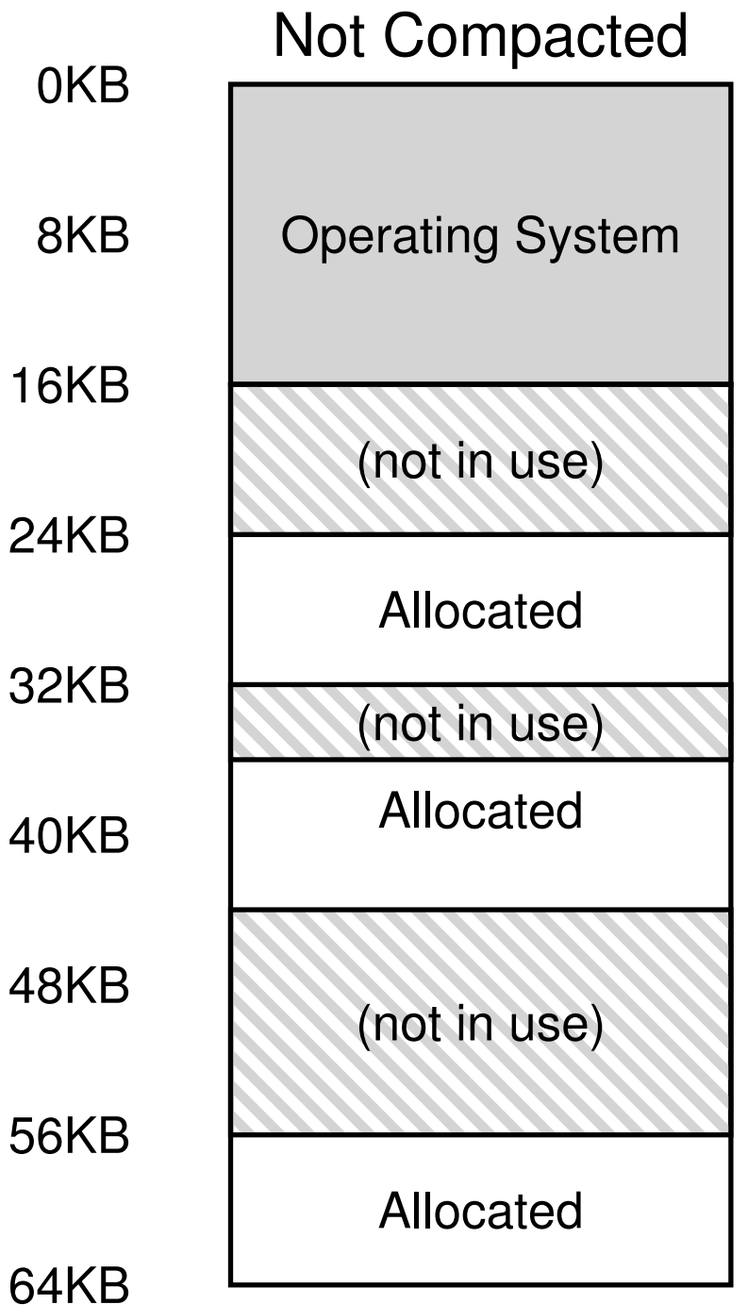

64K
S3
288K
96K
S4
128K
S1
320K
OS
128K
64K
S3
288K
96K
S4
128K
320K
OS
128K
64K
S3
288K
96K
S4
128K
96K
S5
224K
OS
128K

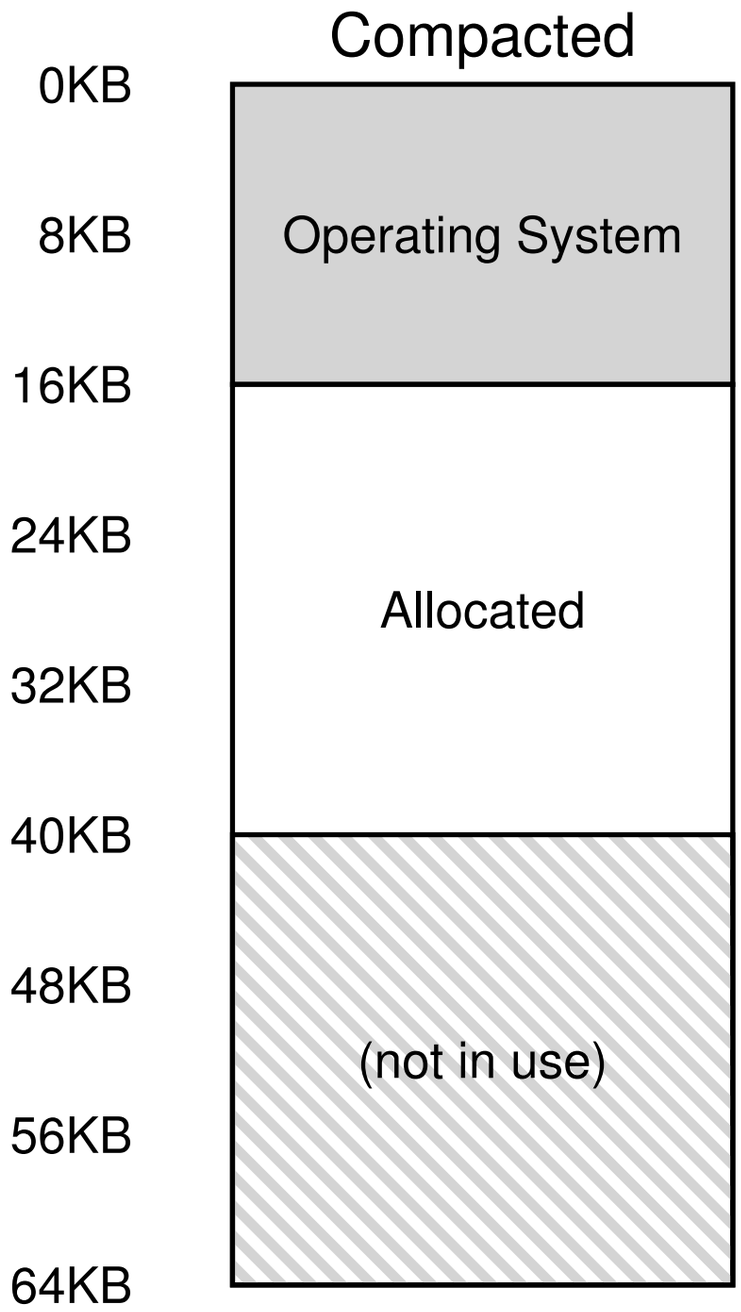

896K
OS
128K
576K
S1
320K
OS
128K
352K
S2
224K
S1
320K
OS
128K
64K
S3
288K
S2
224K
S1
320K
OS
128K
64K
S3
288K
224K
S1
320K
OS
128K


64K
S3
288K
96K
S4
128K
S1
320K
OS
128K
64K
S3
288K
96K
S4
128K
320K
OS
128K
64K
S3
288K
96K
S4
128K
96K
S5
224K
OS
128K
64K
S3
288K
96K
S4
128K
96K
S5
224K
OS
128K
???
S6
128K

Solution: Compaction

But copying segments is memory-intensive — heavy overhead





What we've covered so far

Three Easy Pieces

16 (Segmentation)