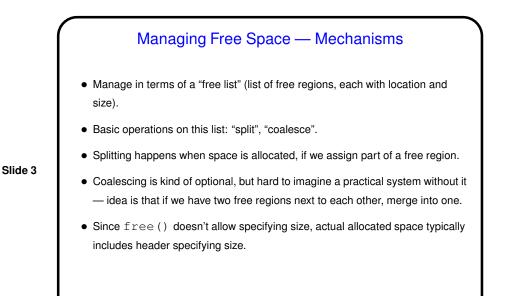


Slide 1

## Managing Free Space Problem of managing varying-size chunks of memory arises in more than one context — space within process (malloc() and free()), space among processes. (Note in passing that if using the textbook's version of base/bound method of allocating space to processes, problem doesn't really arise — all chunks are same size and we only have to keep track of which are free.) Since chunks vary in size, "external fragmentation" a problem. Compaction can sometimes help, but not always: Consider whether you can do that with space managed by malloc() and free(). How can you find all references?



Managing Free Space — Mechanisms, Continued
Free list itself has to be kept somewhere! Linux puts it within the space being managed(!). Details interesting but could be skipped/skimmed.
Where does the space come from in the first place? Call to mmap(). Interesting function in that it can either copy contents of a file (all or part) directly into memory or just reserve an area of memory. Worth noting that mmap() is specific to Linux and some other UNIX-like systems. But other operating systems likely have something alone the same lines.

Slide 4

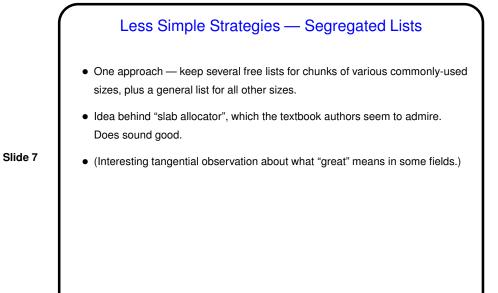
2



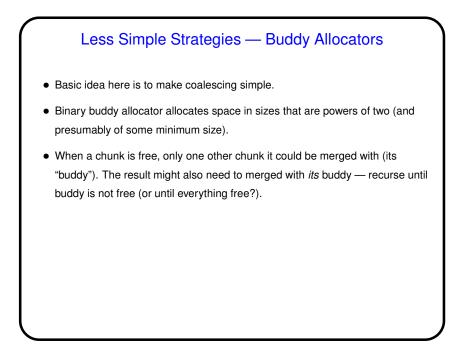
Simple Strategies
Best fit.
Worst fit.
First fit.
Next fit.

Slide 5

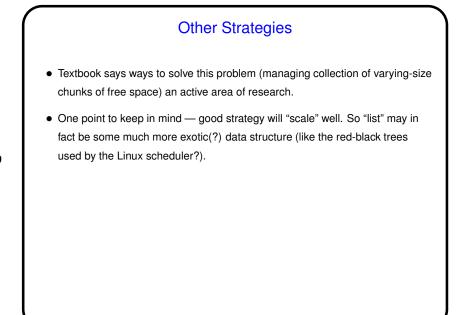
Slide 6



Slide 8



4



Slide 9

