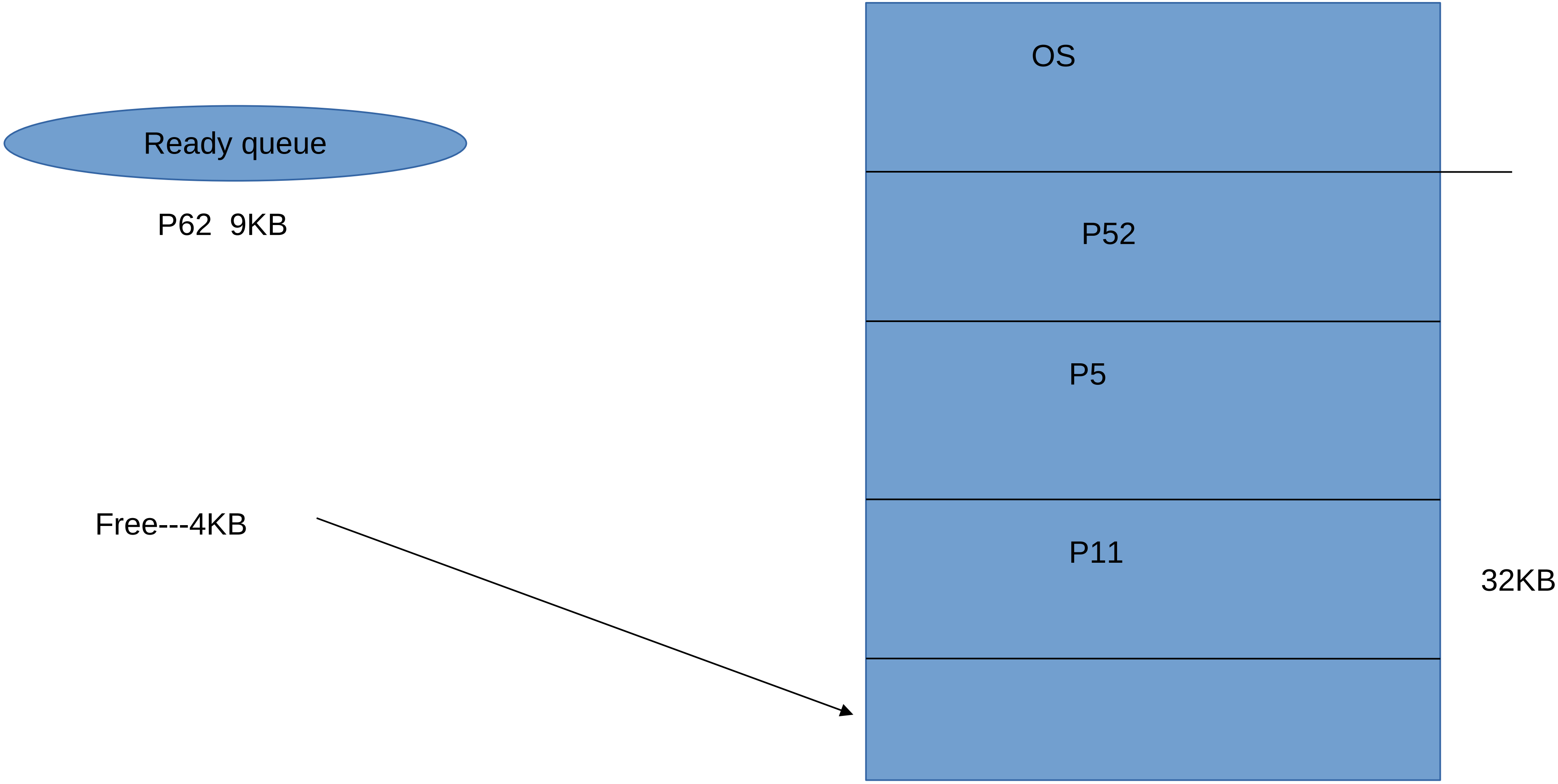
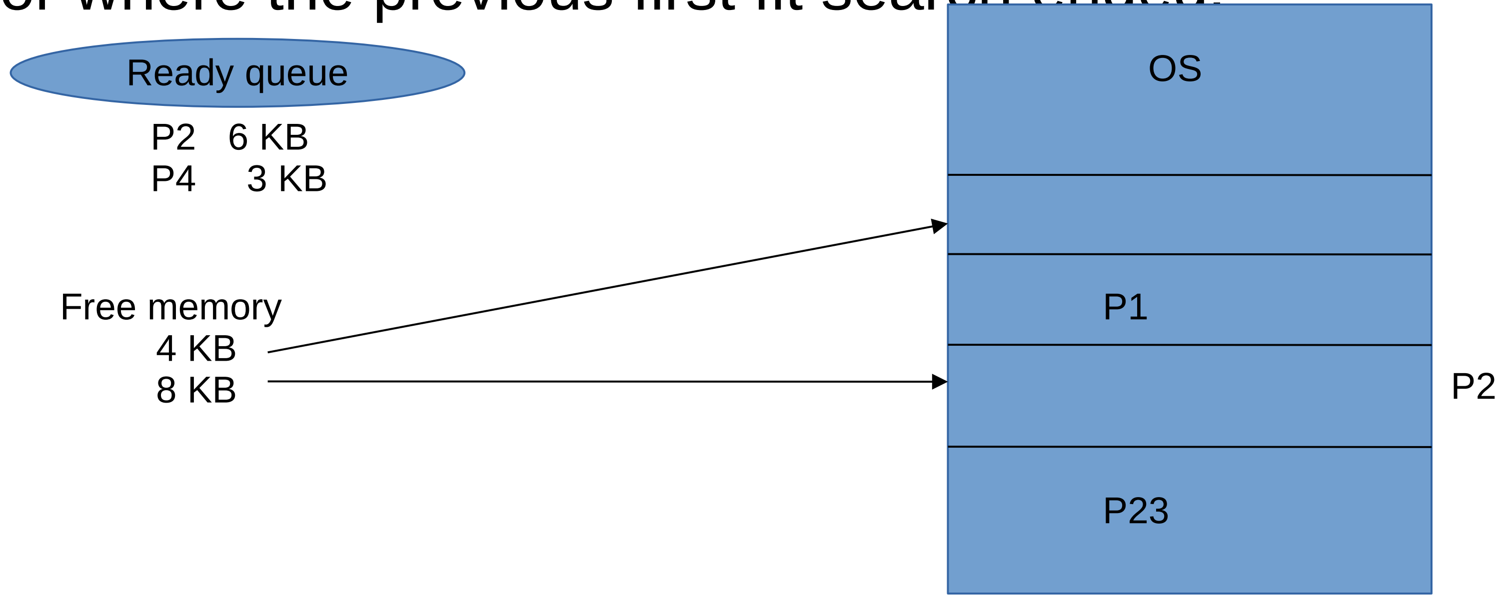

Ready queue
P62 9KB
Free---4KB
OS
P52
P5
P11
32KB

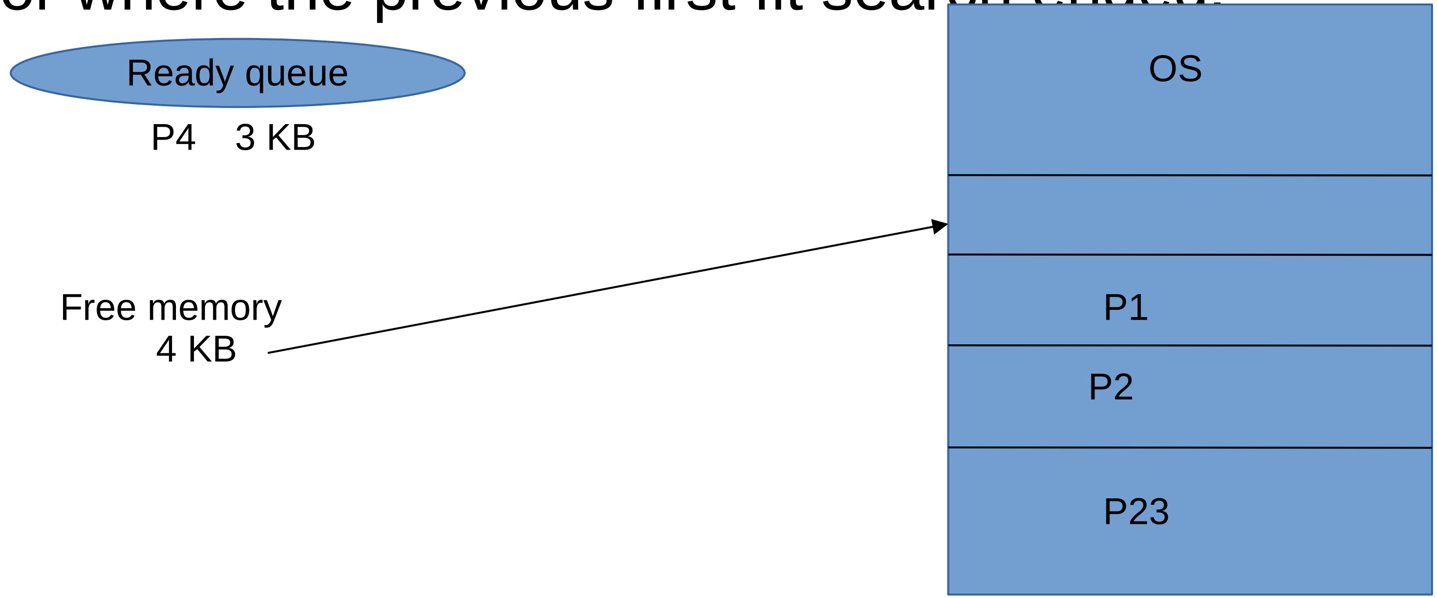
Selection of a hole to hold a process

First fit: allocate the first hole that is big enough.
Searching can start at the beginning of the set of holes or where the previous first fit search ended.



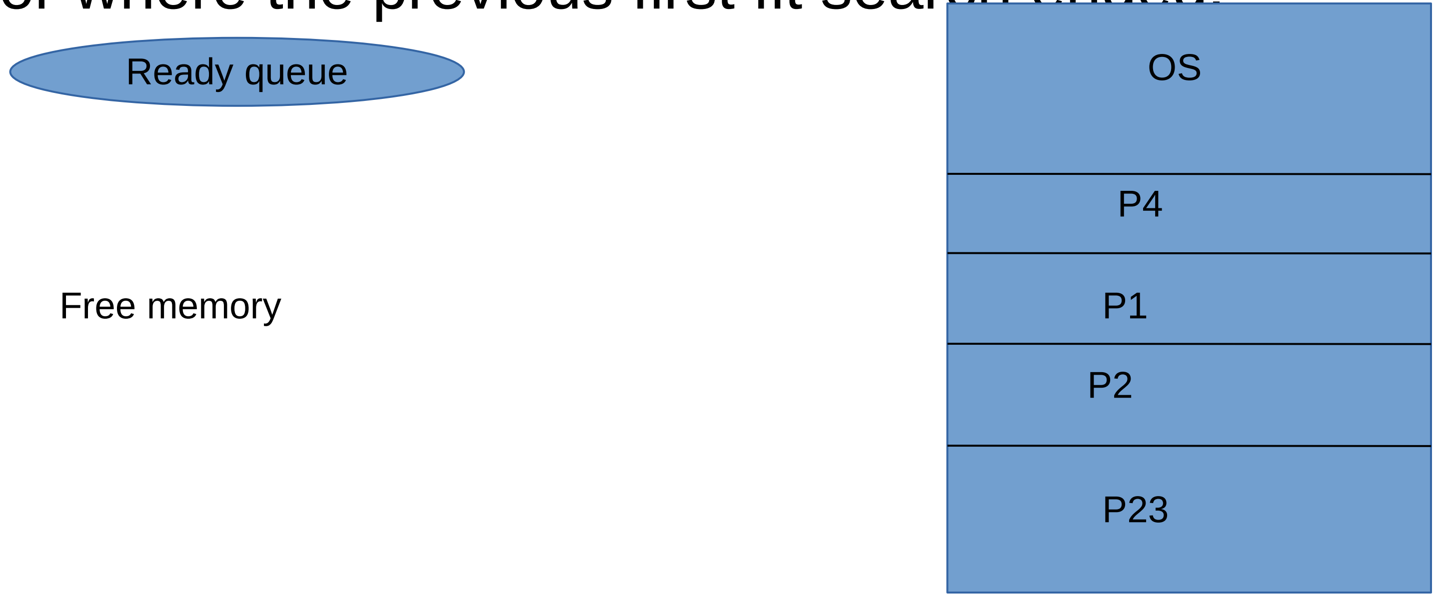
Selection of a hole to hold a process

First fit: allocate the first hole that is big enough. Searching can start at the beginning of the set of holes or where the previous first fit search ended.



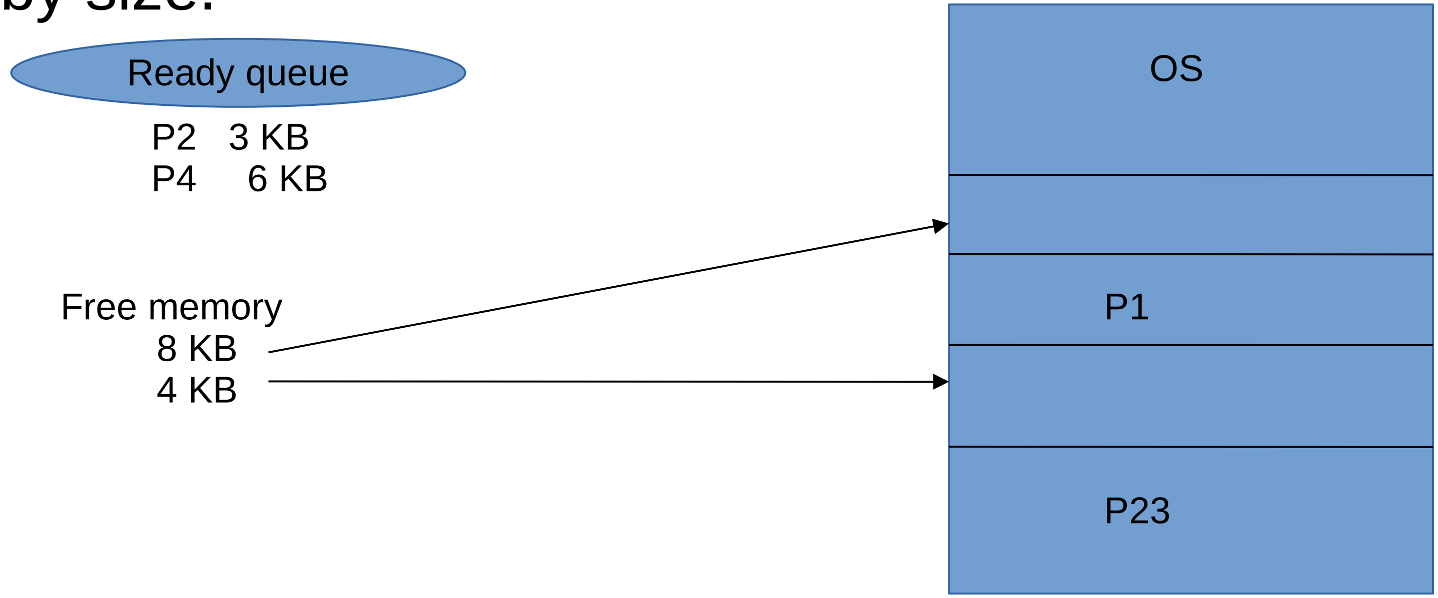
Selection of a hole to hold a process

First fit: allocate the first hole that is big enough. Searching can start at the beginning of the set of holes or where the previous first fit search ended.



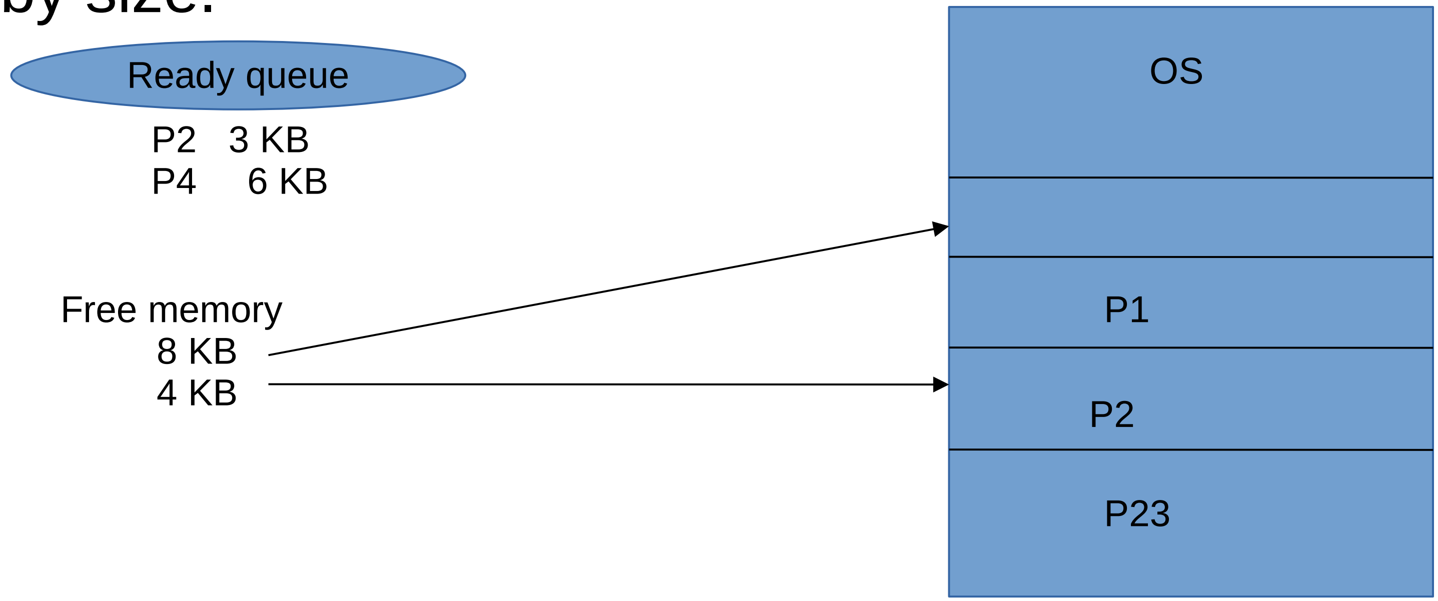
Selection of a hole to hold a process

best fit: allocate the smallest hole that is big enough. We must Search the entire list, unless the list is kept ordered by size.



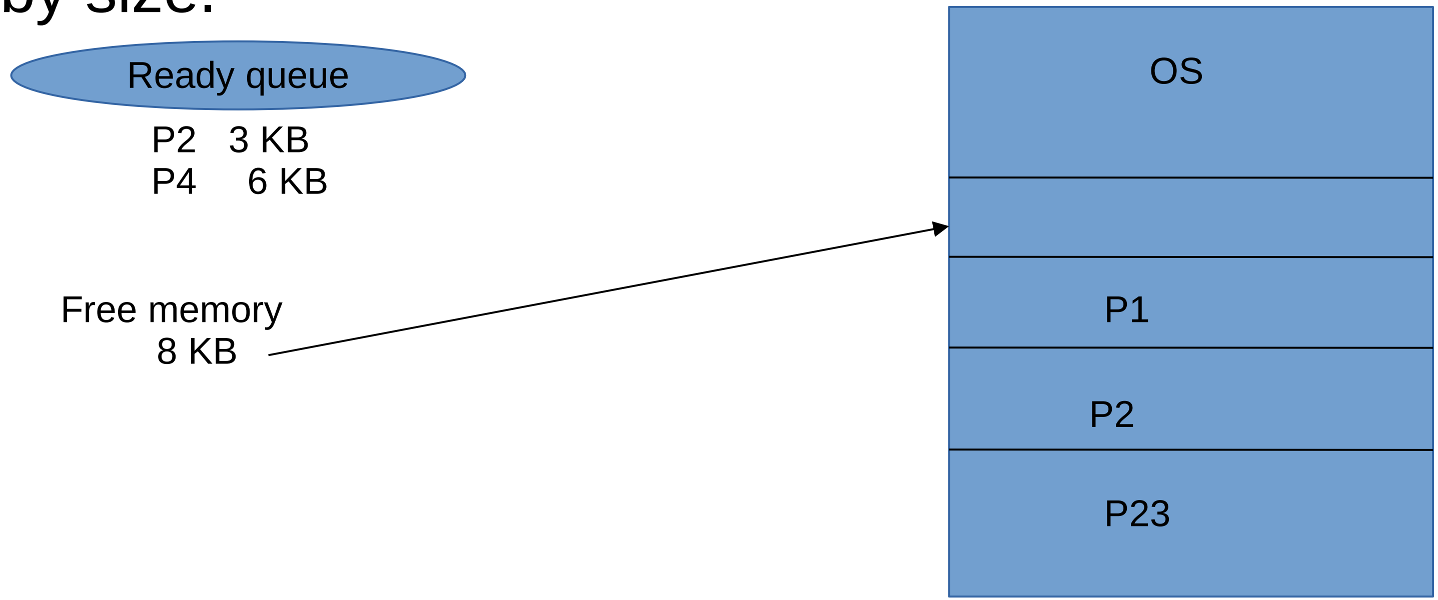
Selection of a hole to hold a process

best fit: allocate the smallest hole that is big enough. We must Search the entire list, unless the list is kept ordered by size.



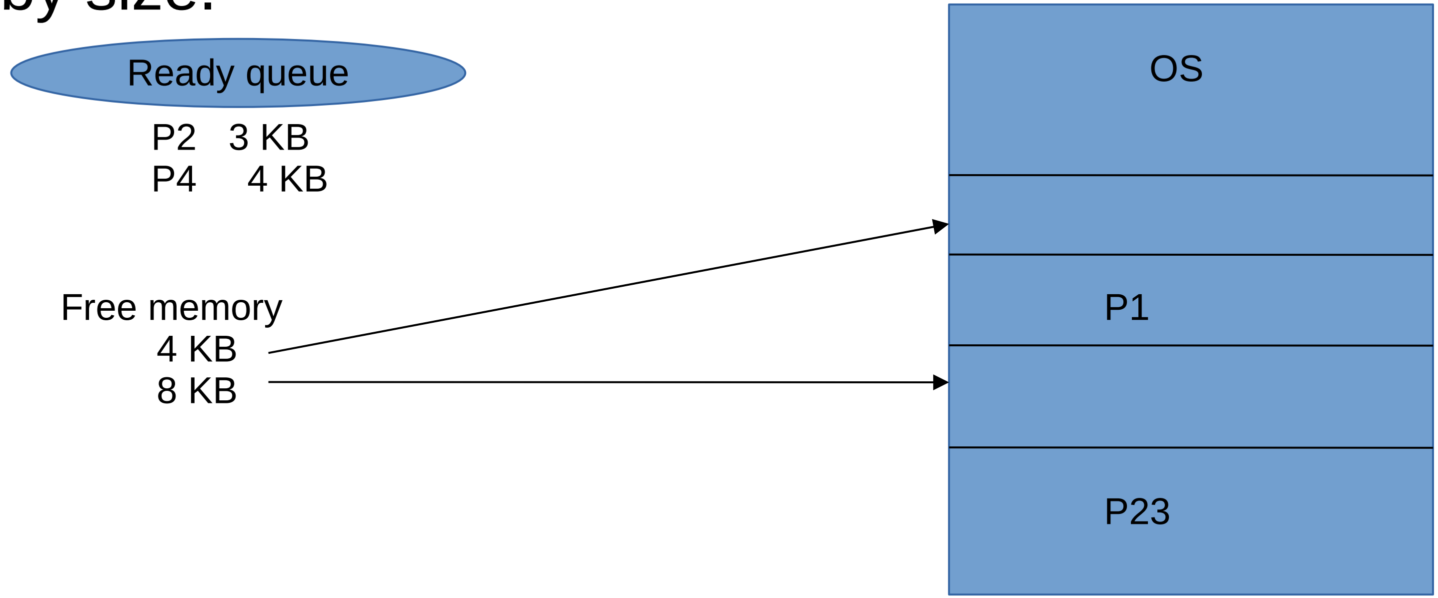
Selection of a hole to hold a process

best fit: allocate the smallest hole that is big enough. We must Search the entire list, unless the list is kept ordered by size.



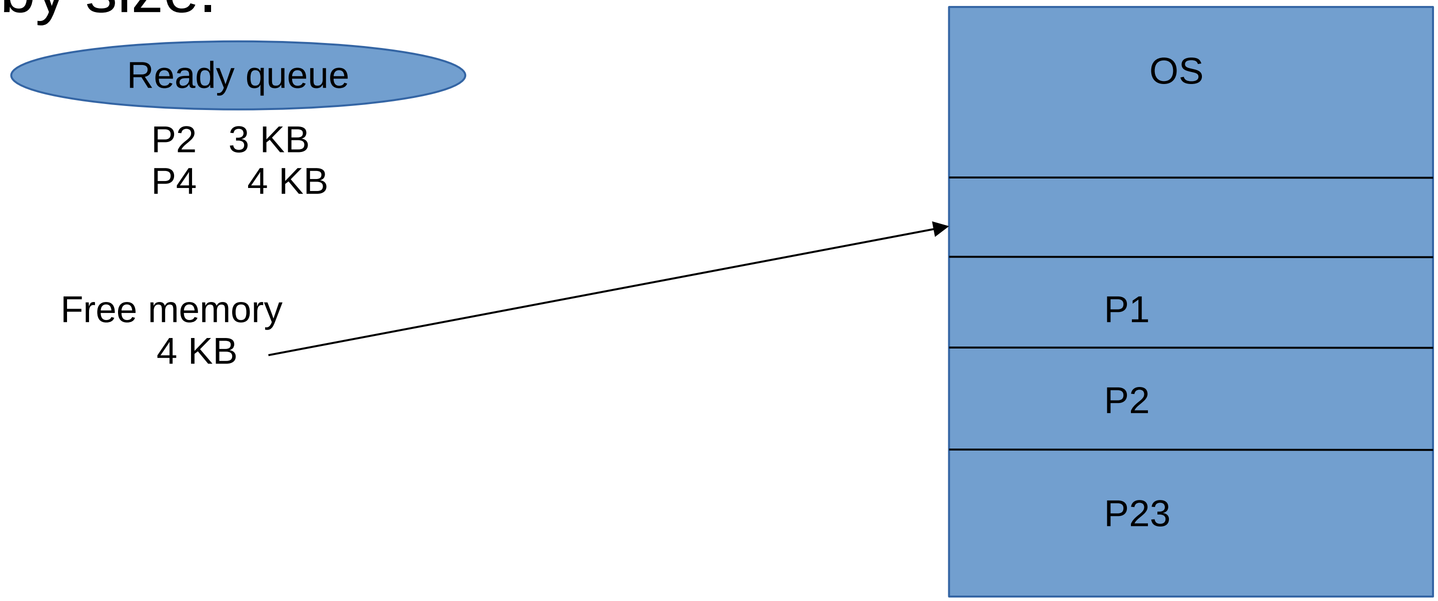
Selection of a hole to hold a process

Worst fit: allocate the largest hole that is big enough. We must Search the entire list, unless the list is kept ordered by size.



Selection of a hole to hold a process

Worst fit: allocate the largest hole that is big enough. We must Search the entire list, unless the list is kept ordered by size.



Simulations have shown that both first fit and best fit are better than worst fit in terms of decreasing both time and storage utilization.

Neither first fit nor best fit is clearly better in terms of storage utilization but first fit is generally faster.

External Fragmentation

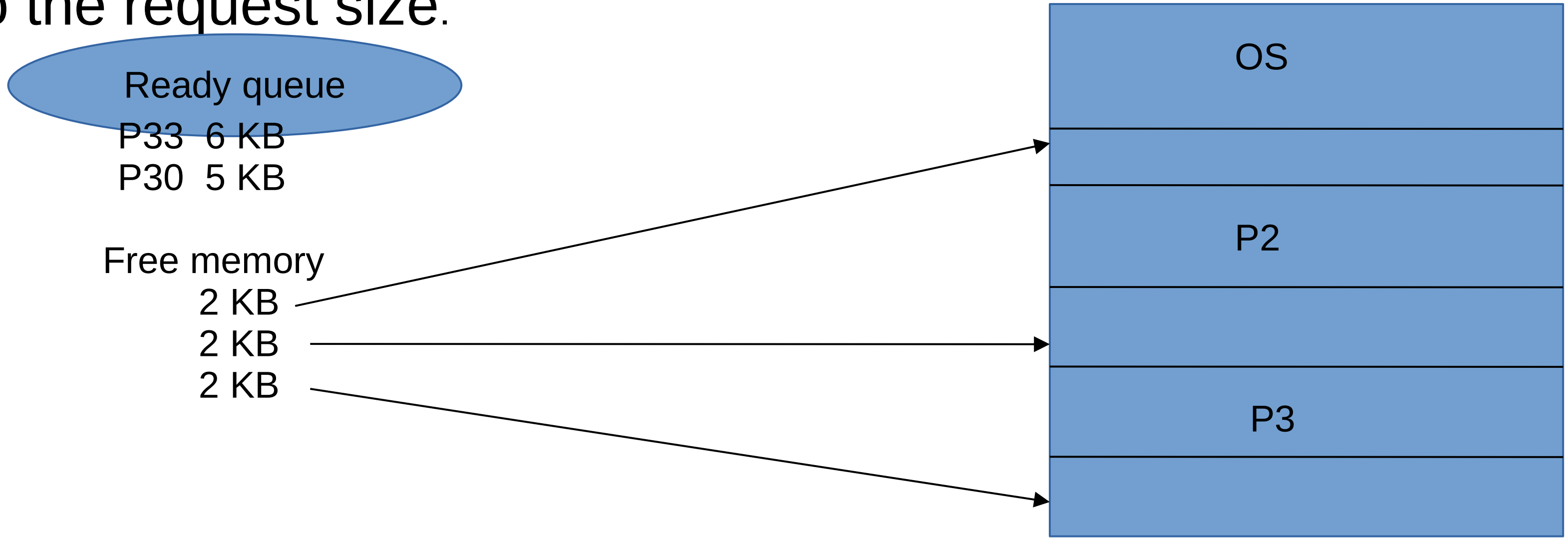
External fragmentation exists when enough total memory space exists to satisfy a request, but it is not contiguous.

In other words there are a number of small holes none of which is large enough to satisfy a request, but the sum of their sizes is greater than or equal to the request size.

External Fragmentation

External fragmentation exists when enough total memory space exists to satisfy a request, but it is not contiguous.

In other words there are a number of small holes none of which is large enough to satisfy a request, but the sum of their sizes is greater than or equal to the request size.



Solution?

Compaction

The goal is to shuffle the memory contents to place all the free memory together in one large block.

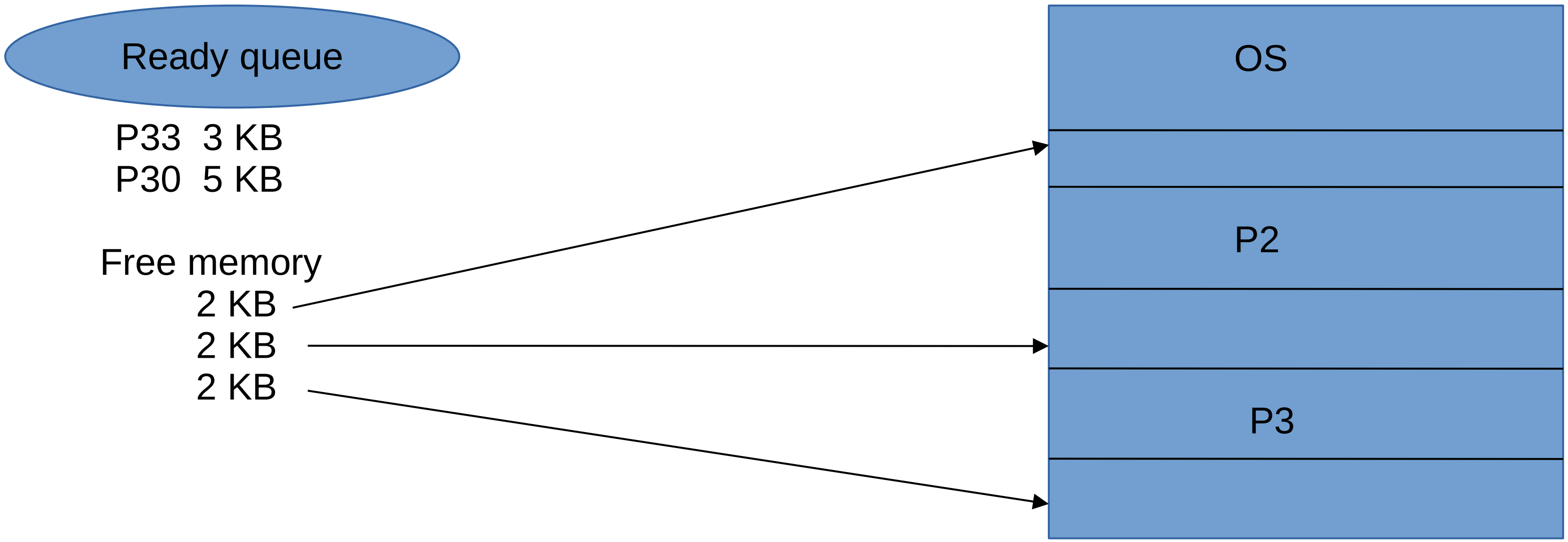
For a relocatable process to be able to execute in its new location, all internal addresses must be relocated.

If the relocation is static (Binding at compile time) compaction can not be done.

If relocation is dynamic (Binding at load time) compaction can be done.

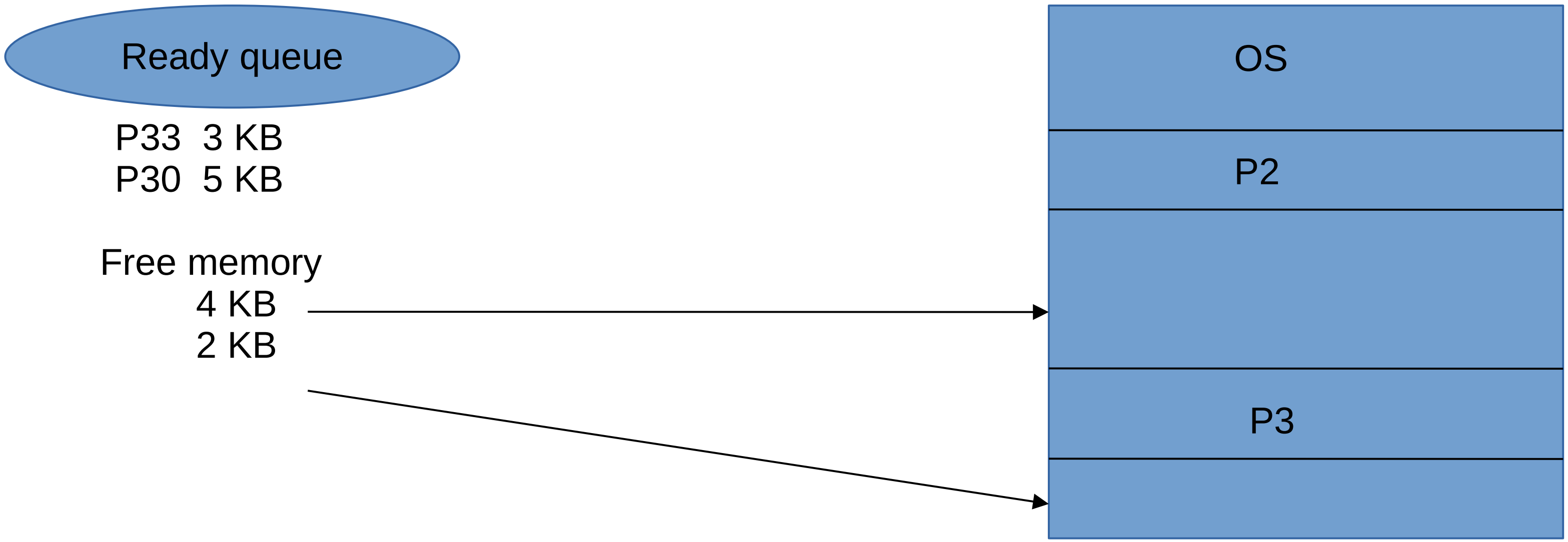
Compaction Technique

A. simply move all processes towards one end of memory all holes move in the other direction producing one large hole of available memory.



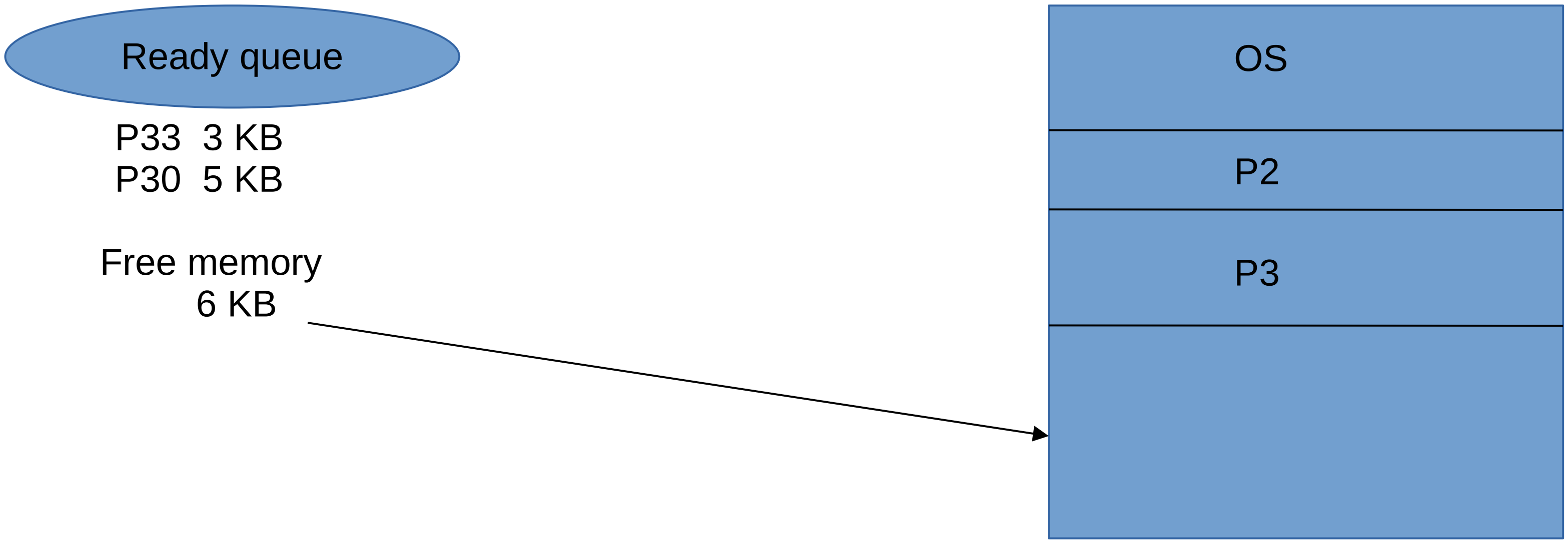
Compaction Technique

A. simply move all processes towards one end of memory all holes move in the other direction producing one large hole of available memory.



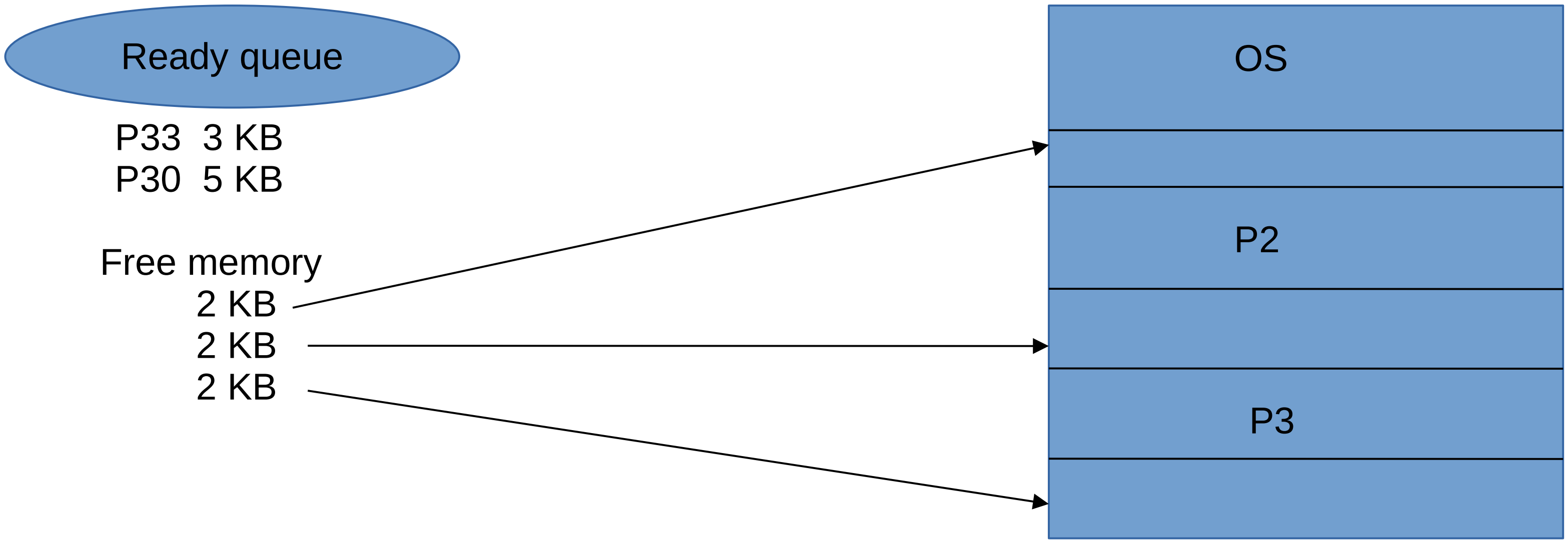
Compaction Technique

A. simply move all processes towards one end of memory all holes move in the other direction producing one large hole of available memory.



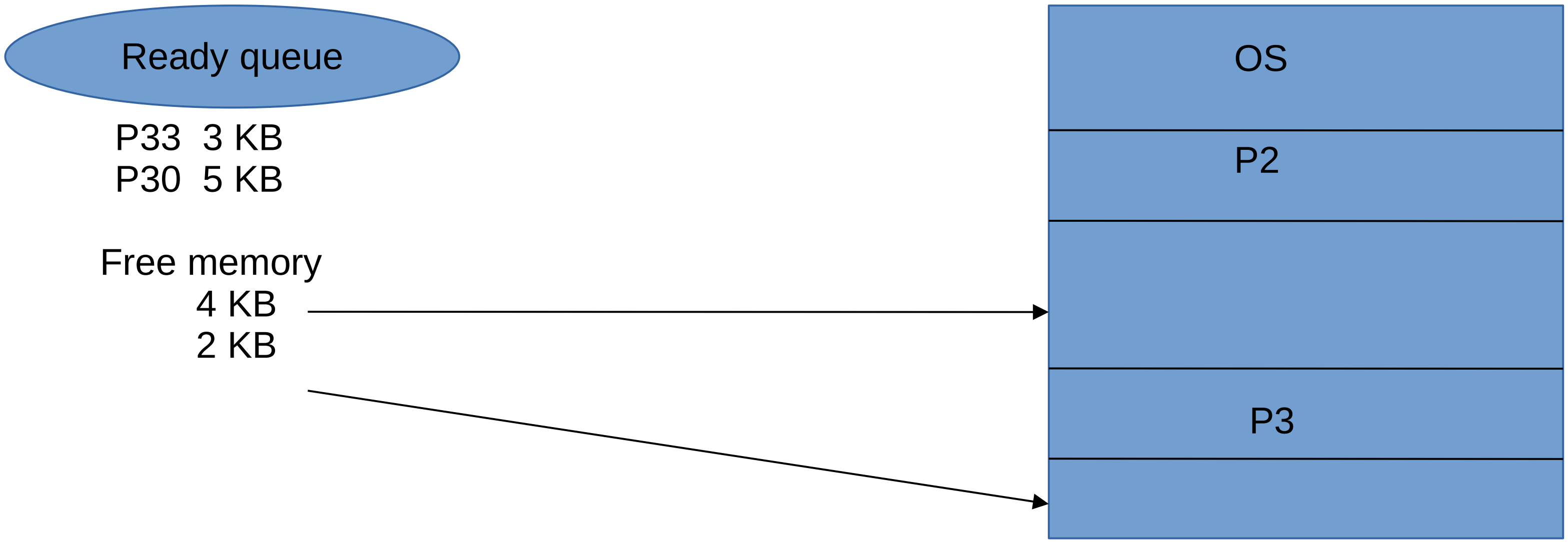
Compaction Technique

B. Create a large hole big enough anywhere to satisfy the request.



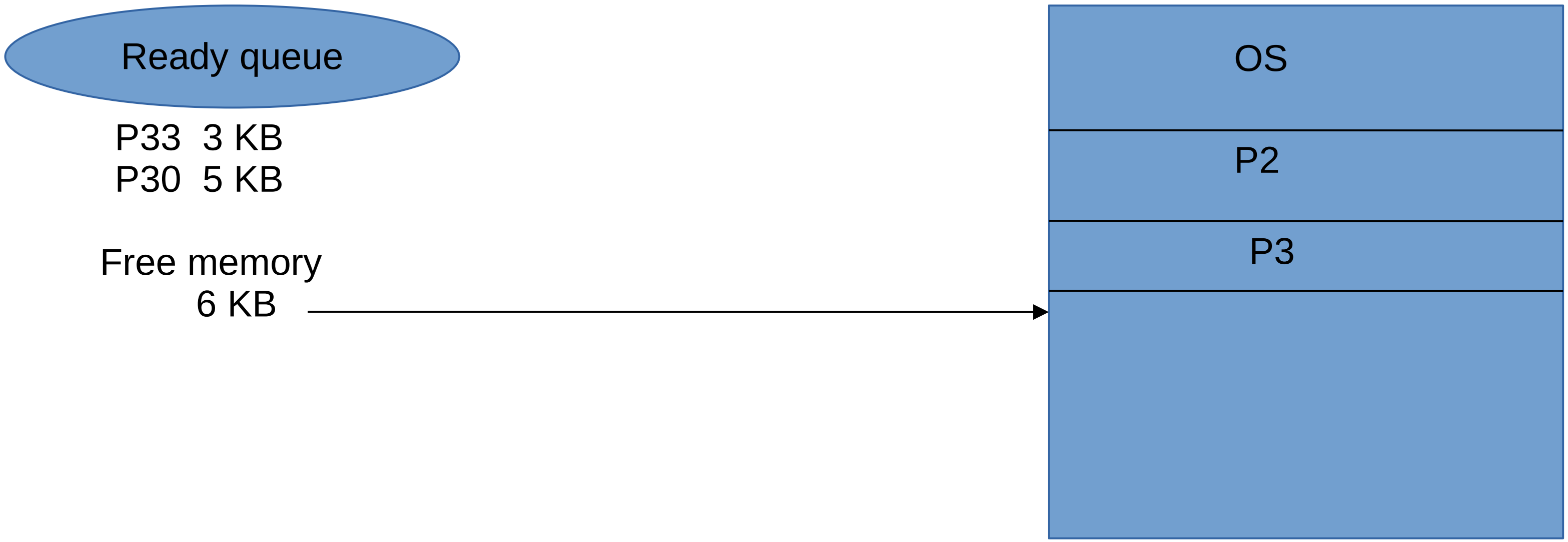
Compaction Technique

B. Create a large hole big enough anywhere to satisfy the request.



Compaction Technique

B. Create a large hole big enough anywhere to satisfy the request.



Internal Fragmentation

Consider a hole of 2002 bytes and let us say this is the only available at the moment.

Suppose the next process requests 2000 bytes.

If we allocate exactly the requested block, we are left with a hole of 2 bytes.

The overhead to keep track of this hole will be substantially larger than the hole itself.

The general idea is to allocate this hole as part of the larger request.

There fore the allocated memory is slightly larger than the request. So a little amount of memory is wasted.

So internal fragmentation exist.

Compaction is really tough ……. solution?