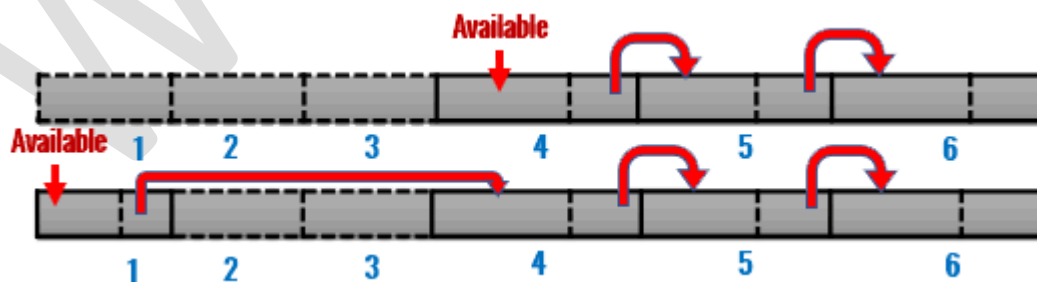


Dynamic Storage Allocation

- The techniques needed to implement dynamic storage allocation is mainly depends on how the storage deallocated. If deallocation is implicit, then the run-time support package is responsible for determining when a storage block is no longer needed. There is less a compiler has to do if deallocation is done explicitly by the programmer.

Explicit Allocation of Fixed-Sized Blocks

- The simplest form of dynamic allocation involves blocks of a fixed size.
- Allocation and deallocation can be done quickly with little or no storage overhead.
- Suppose that blocks are to be drawn from a contiguous area of storage. Initialization of the area is done by using a portion of each block for a link to the next block.
- A pointer available points to the first block. Allocation consists of taking a block off the list and deallocation consists of putting the block back on the list.



Explicit Allocation of Variable-Sized Blocks

- When blocks are allocated and deallocated, storage can become fragmented; that is, the heap may consist of alternate blocks that are free.
- The situation can occur if a program allocates five blocks and then de-allocates the second and fourth.
- Fragmentation is of no consequence if blocks are of fixed size, but if they are of variable size, because we could not allocate a block larger than any one of the free blocks, even though the space is available.
- First fit, worst fit and best fit are some methods for allocating variable-sized blocks.



For More Details Click Here:

<https://www.wikitechy.com/tutorials/compiler-design/dynamic-storage-allocation>