Variable-Partition Multiprogramming

Variable-Partition Characteristics

-The queue at the top of the figure contains available jobs and information about their memory requirements. The operating system makes no assumption about the size of a job except that it does not exceed the size of available main memory.

-The system progresses through the queue and places each job in memory, where there is available space, at which point it becomes a process. This organization does not suffer from internal fragmentation, because a process's partition is exactly the size of the process.

-The waste does not become obvious until processes finish and leave holes in main memory. The system can continue to place new processes in these holes. Every hole eventually becomes too small to hold a new process. This is called external

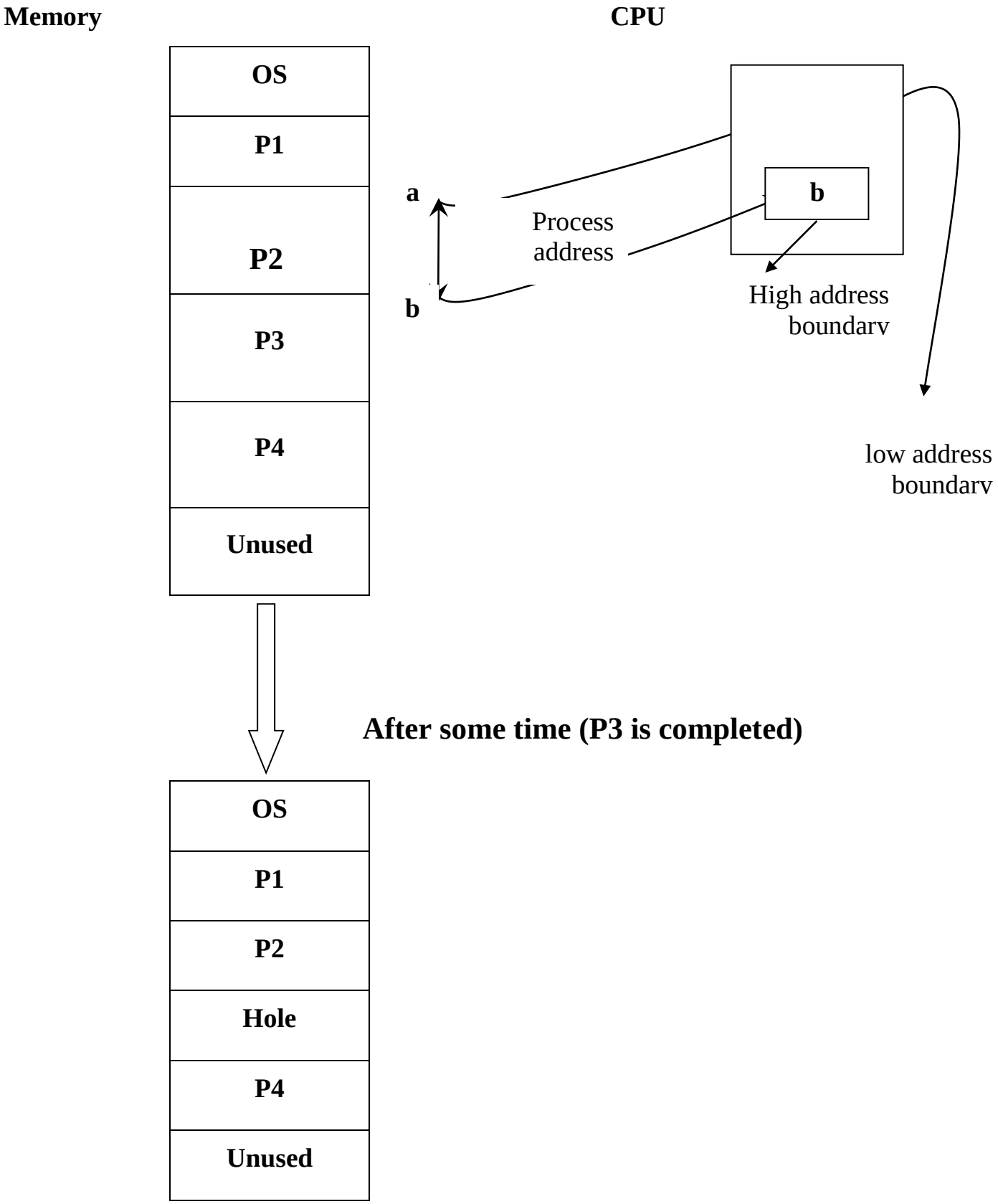
fragmentation, where the sum of the holes is enough to accommodate another process.

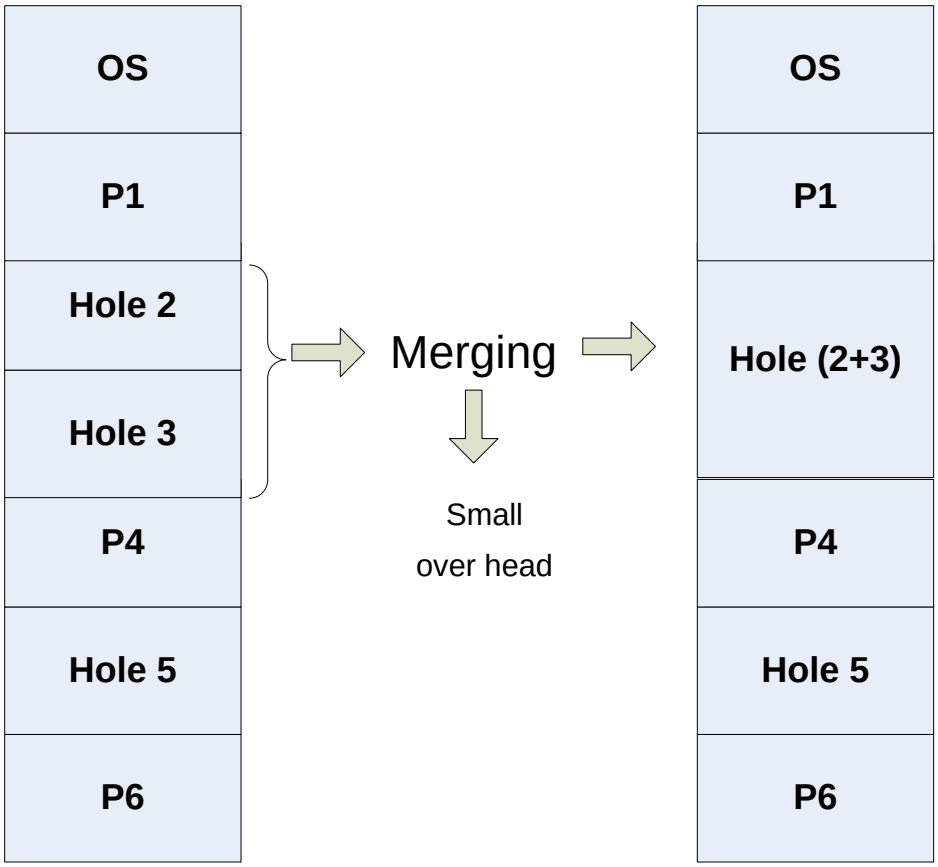
-The system can determine whether the newly freed memory area is adjacent to other free memory areas. The system then records in a free memory list either (1) that the system now has an additional hole or (2) that an existing hole has been enlarged.

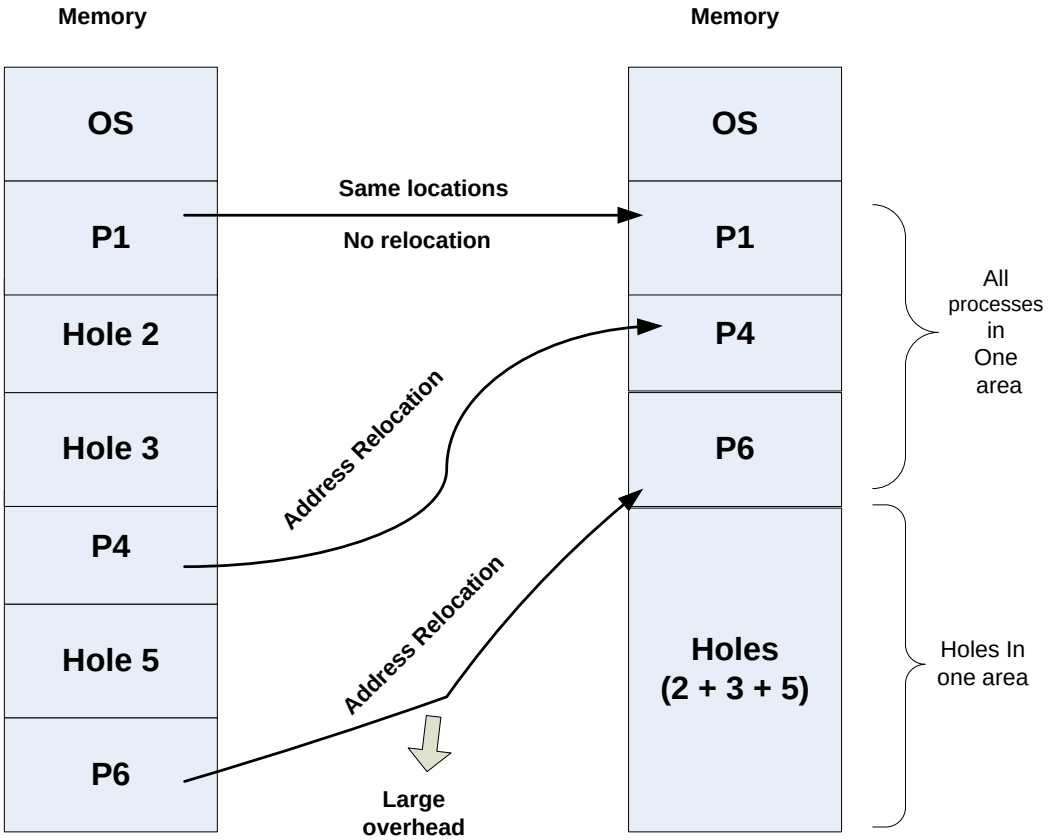
-The process of merging adjacent holes to form a single, larger hole is called coalescing (merge things). The system reclaims the largest possible contiguous blocks of memory.

-Another technique for reducing external fragmentation is called memory compaction (burping the memory or garbage collection). This relocates all occupied areas of memory to one end or the other of main memory. Now all of the available free memory is contiguous. The drawbacks are:

- Overhead consumes system resources.
- The system must cease (stop something) all other computation during compaction which results in erratic response times for interactive users.
- Compaction must relocate the process.


Memory
CPU
OS
P1
P2
P3
P4
Unused
a
b
Process address
b
High address boundary
low address boundary
After some time (P3 is completed)
OS
P1
P2
Hole
P4
Unused


OS
P1
Hole 2
Hole 3
P4
Hole 5
P6
Merging
Small
over head
OS
P1
Hole (2+3)
P4
Hole 5
P6


Memory
Memory
OS
P1
Hole 2
Hole 3
P4
Hole 5
P6
Same locations
No relocation
Address Relocation
Address Relocation
Large
overhead
OS
P1
P4
P6
Holes
(2 + 3 + 5)
All
processes
in
One
area
Holes In
one area

Memory Placement Strategies

-Determines where in main memory to place incoming programs and data. The main strategies are:

- **Best fit:** place the job in the smallest possible hole. The disadvantage is that the rest of hole will not be enough for new job.
- **First fit:** place the job in the first suitable hole. The advantage is low overhead i.e. small CPU wasted time in implementing the strategy.
- **Worst fit:** place the job in the largest available hole. The rest of hole may be still enough for new job