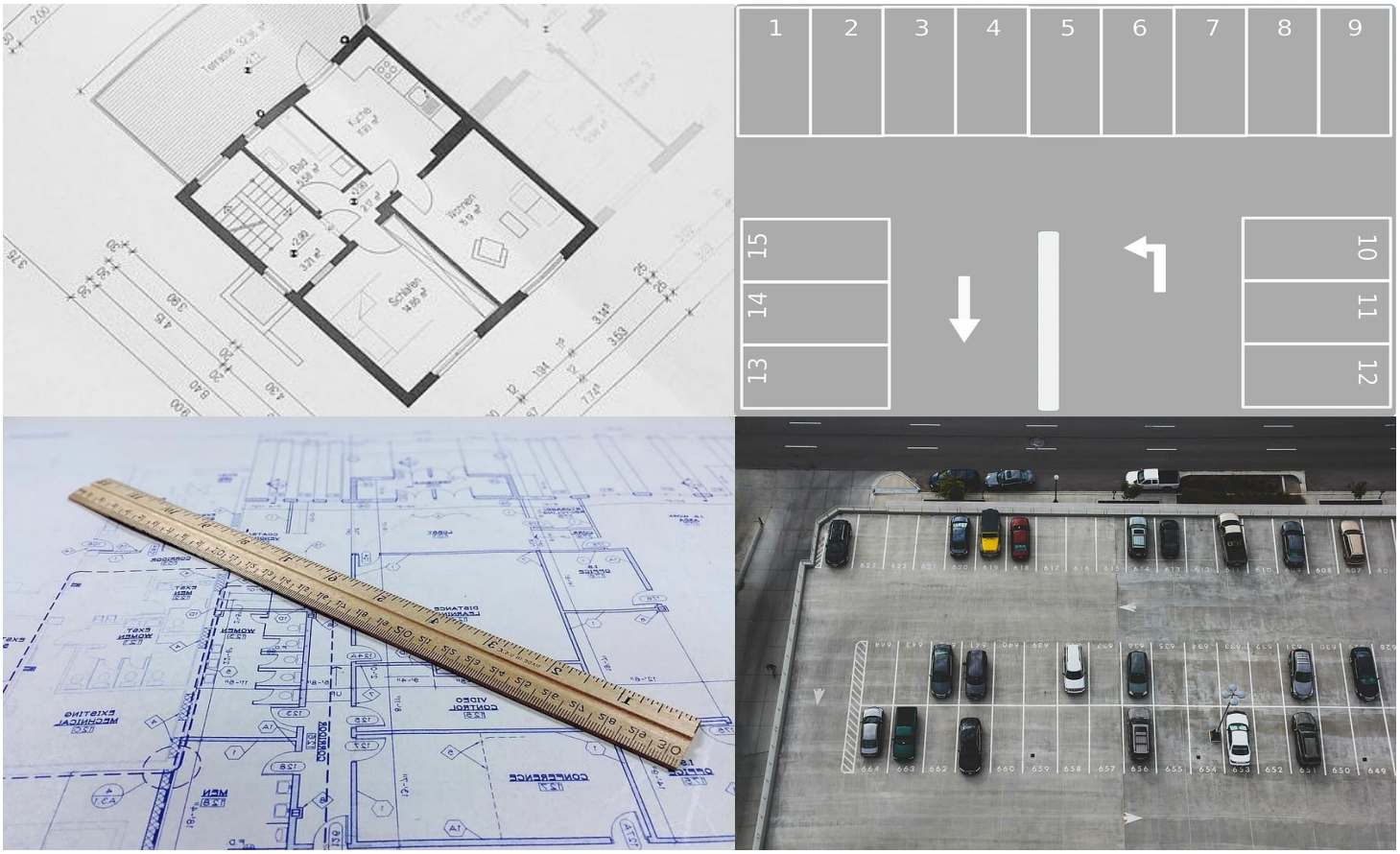
Contiguous Allocation

Contiguous Memory Allocation

- Goal: shared RAM by many processes
- Memory Partitions
  - Place one process in one partition
- **Fixed-Sized** Partitions
  - Location and size of each partition are predefined by the OS
- **Variable-Sized** Partitions
  - Location and size of each partition are determined on-demand, per process' request

Fixed-Size Partitions



Dynamic/Variable-Sized Partitions

Fixed-Sized Partitions

- RAM is divided into **N partitions** (of different sizes but fixed)
  - Configured at boot time
- Only max **N processes** can reside in RAM at any time
  - One partition = one process
  - Fixed multiprogramming level
- When a process must be loaded to RAM to run, the system selects a free partition big enough to fit the process
  - When it terminates, the partition is free again
- ***(Internal) Fragmentation***: unused portion (wasted space) of a partition
- Scheduling Issues? Ready Queue(s)?
- Advantages/disadvantages?

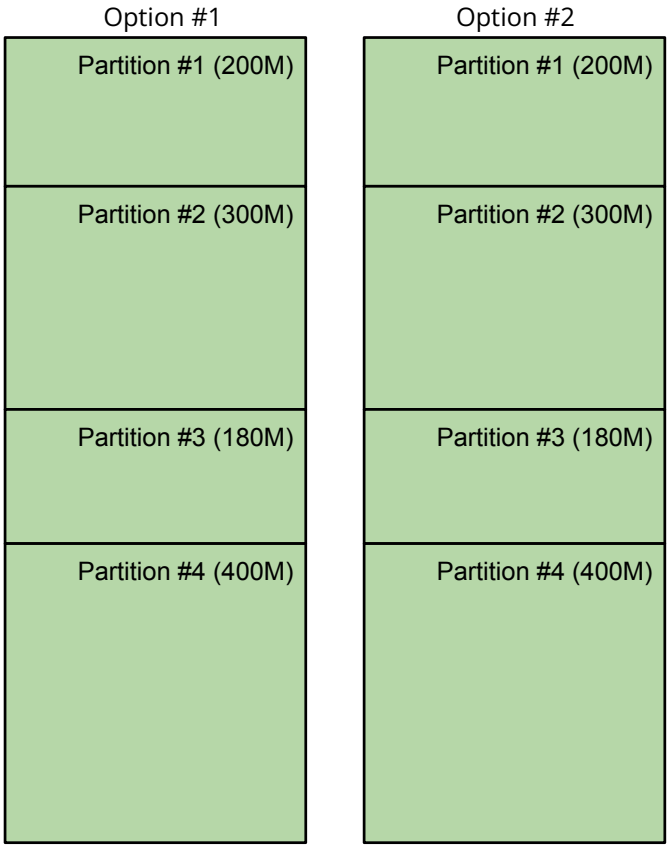
Internal Fragmentation = unused chairs

Fixed-Sized Partitions

Constraint: *Only one process per partition*

1. Process A (100M) begins
2. Process B (310M) begins
3. Process C (200M) begins
4. Process B ends
5. Process D (80M) begins
6. Process A ends
7. Process E (195M) begins
8. Process E ends
9. Process D ends
10. Process C ends

Partitions are configured by the OS, may require system reboot to change

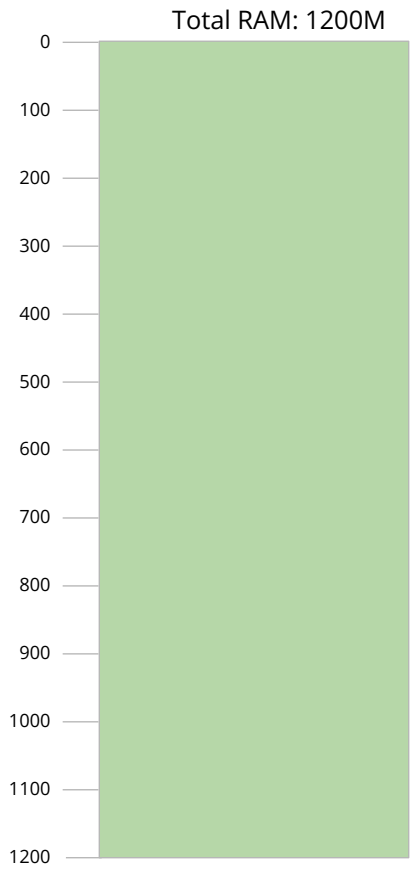


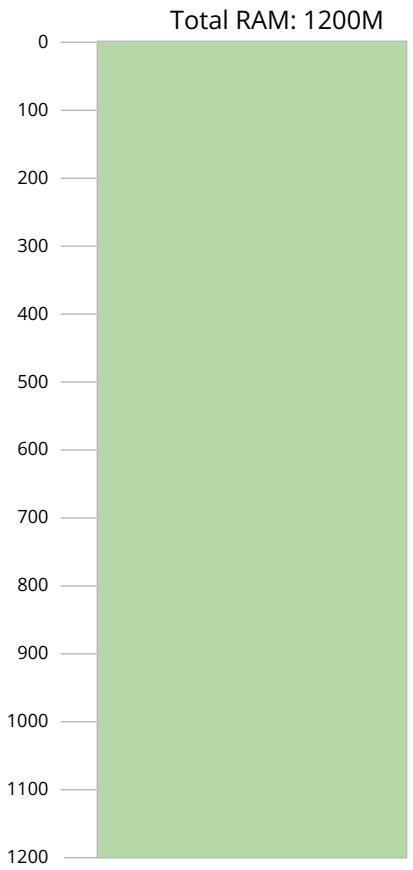
Variable-Sized Partitions

- Partitions are created **on-demand** as processes are loaded
  - Partition size = size of the process just loaded
  - **No internal fragmentation**
- The RAM beings as *one giant hole* (the entire free space)
- ***But***, when a process terminates, it leaves a **hole** (**free partition**)
  - Adjacent holes combine into a larger hole
- Necessary OS data structures
  - A list of **allocated partitions**
  - A list of **holes** (free partitions) link the "holes" themselves into a linked list
- When a process must be loaded, the OS looks for a hole big enough for the process
  - *Unused portion of the hole becomes a smaller hole*

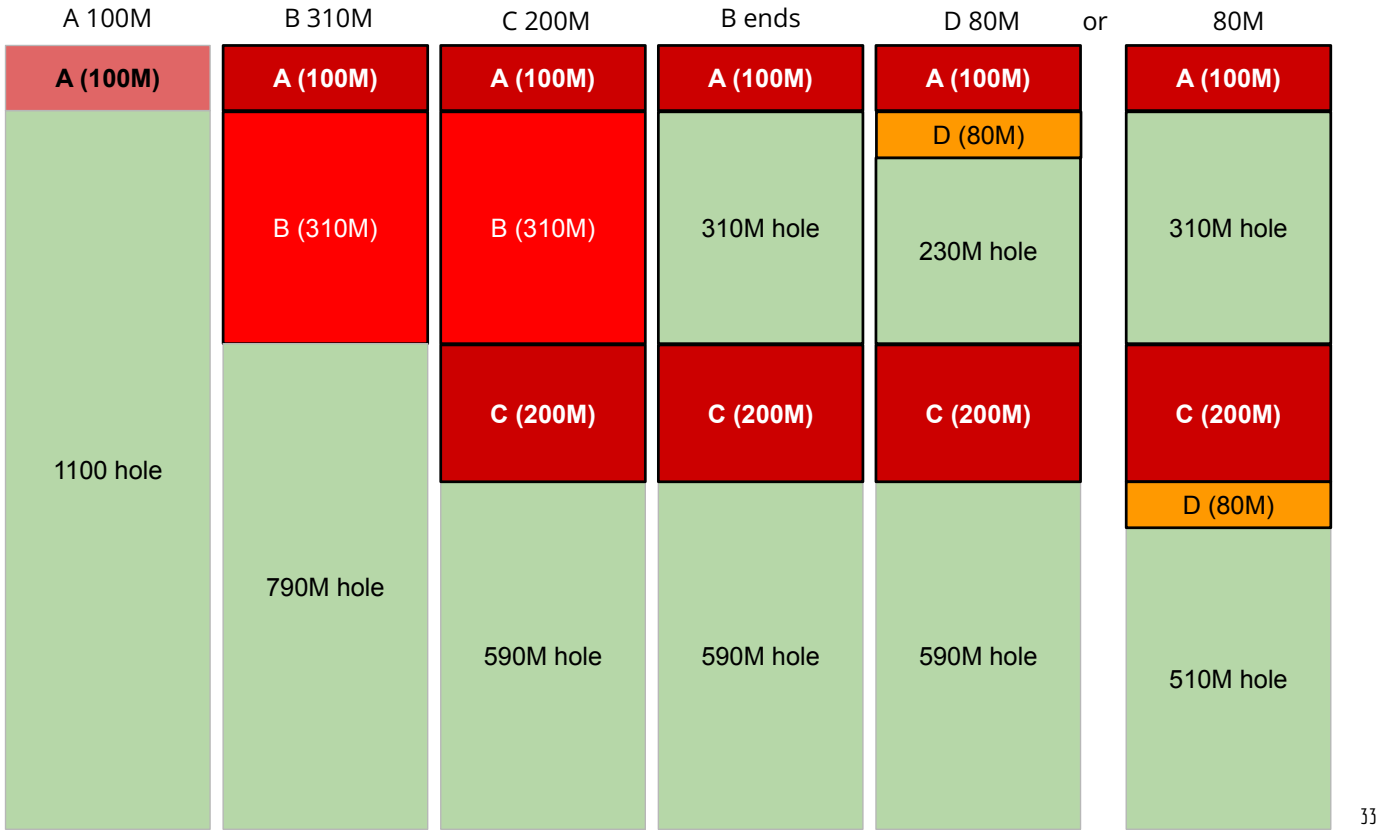
Variable-Sized Partitions

1. Process A (100M) begins
2. Process B (310M) begins
3. Process C (200M) begins
4. Process B ends
5. Process D (80M) begins
6. Process A ends
7. Process E (195M) begins
8. Process E ends
9. Process D ends
10. Process C ends

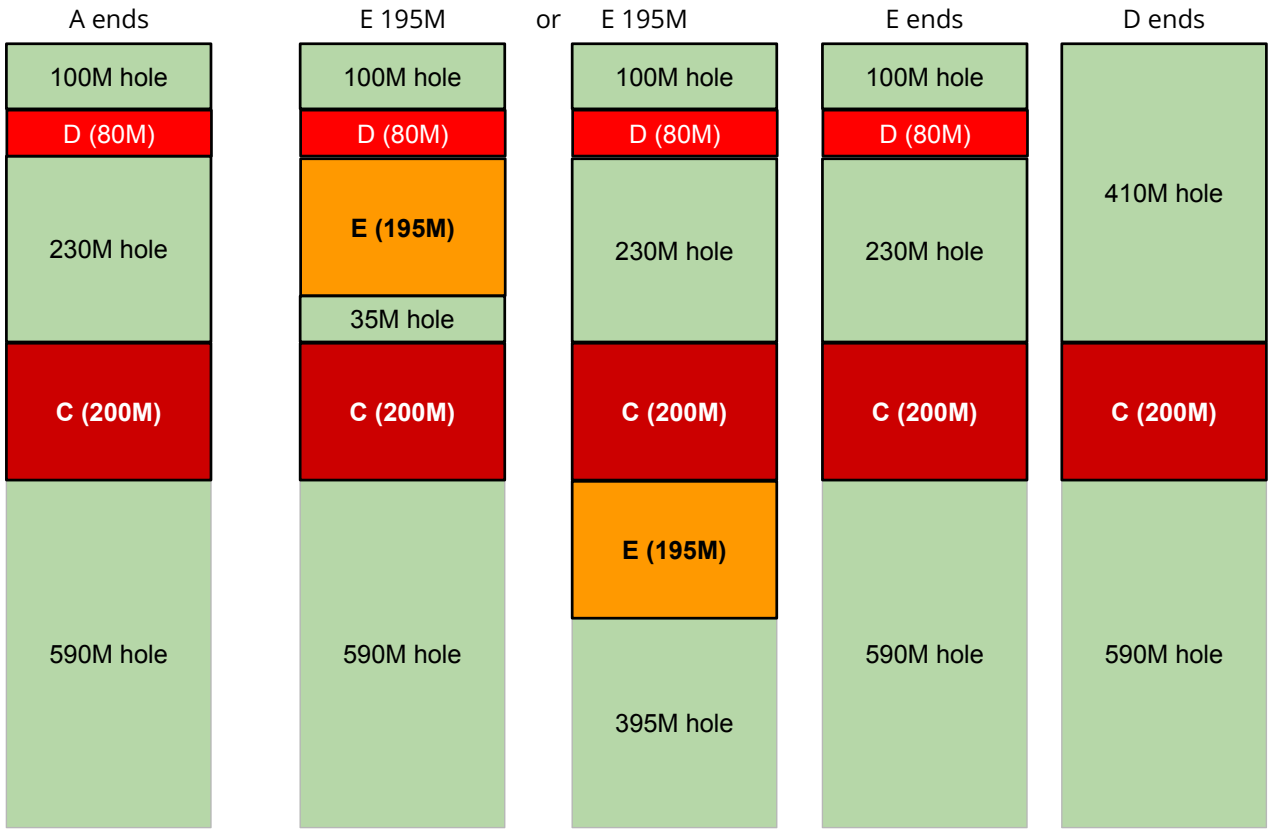




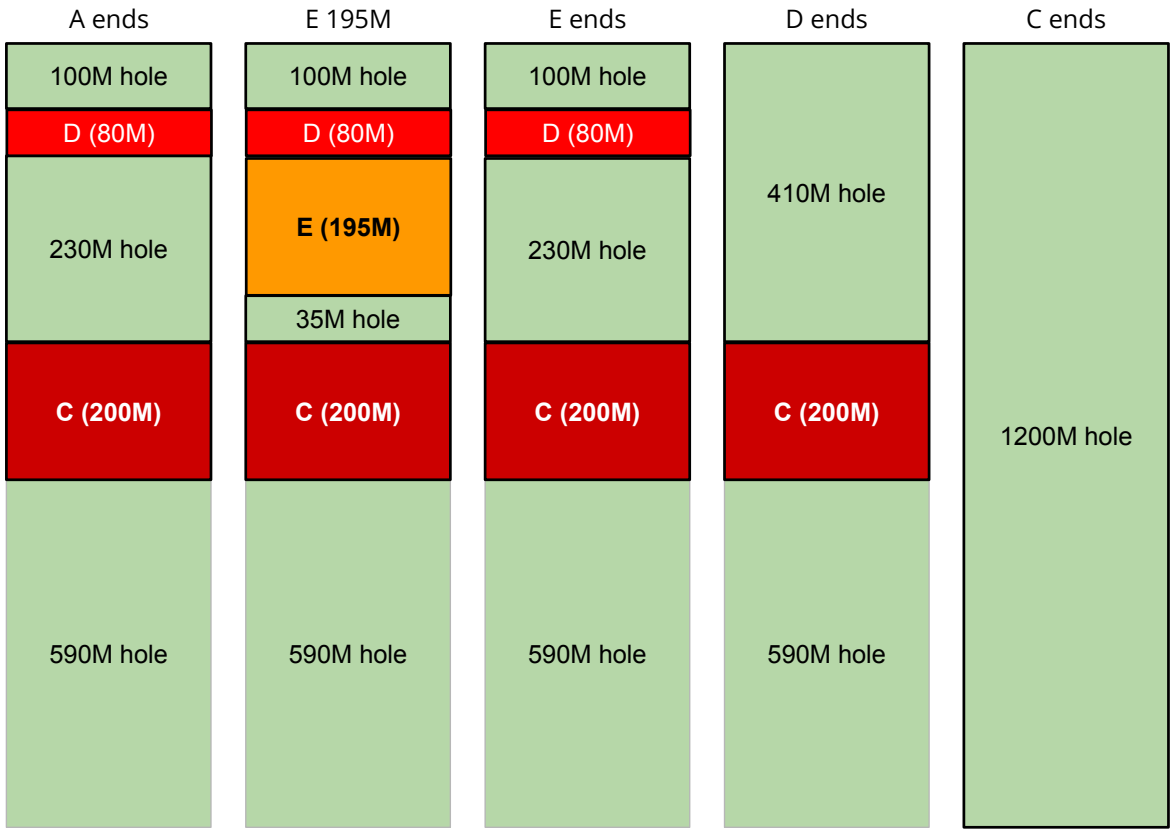
Variable-Sized Partitions (Total RAM 1200M)



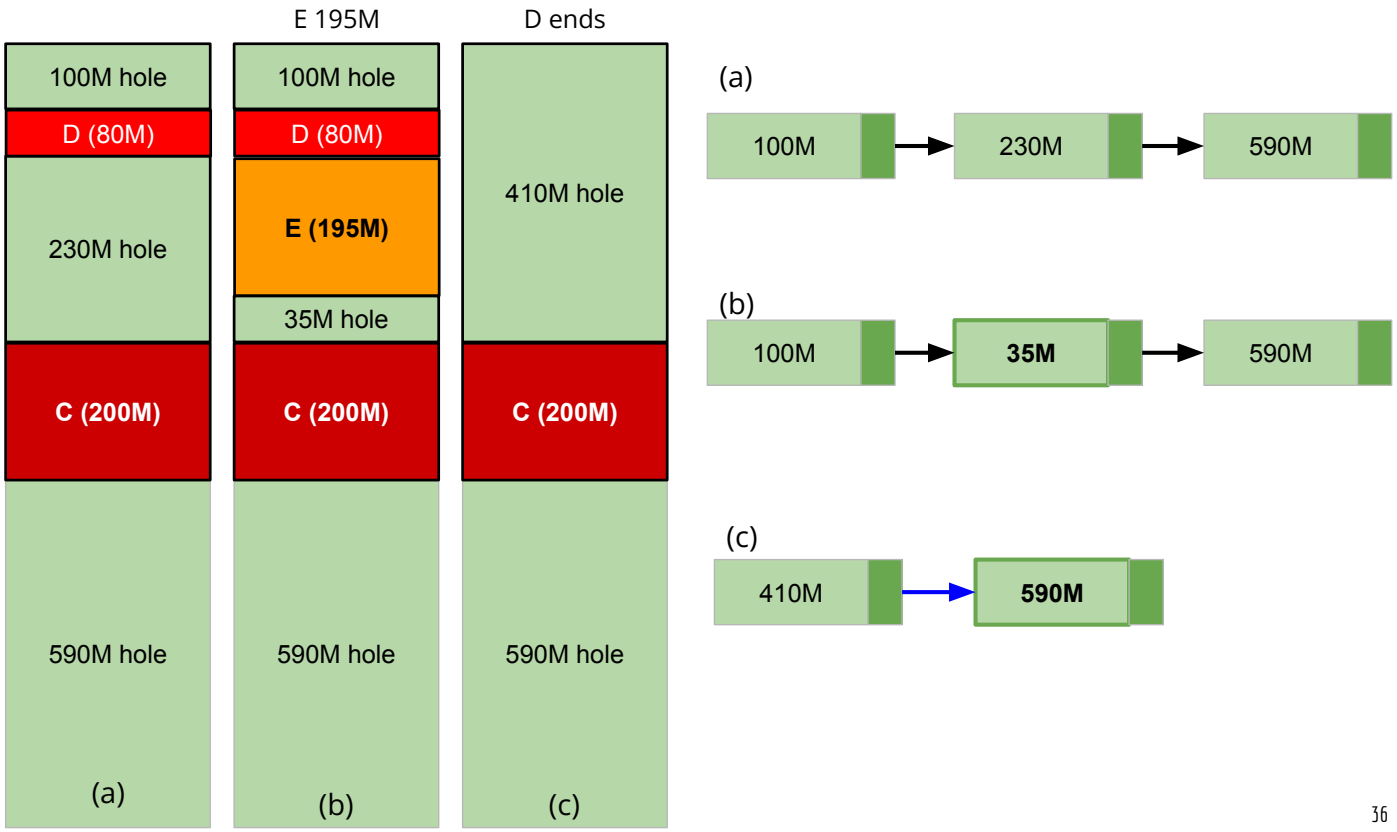
Variable-Sized Partitions



Variable-Sized Partitions



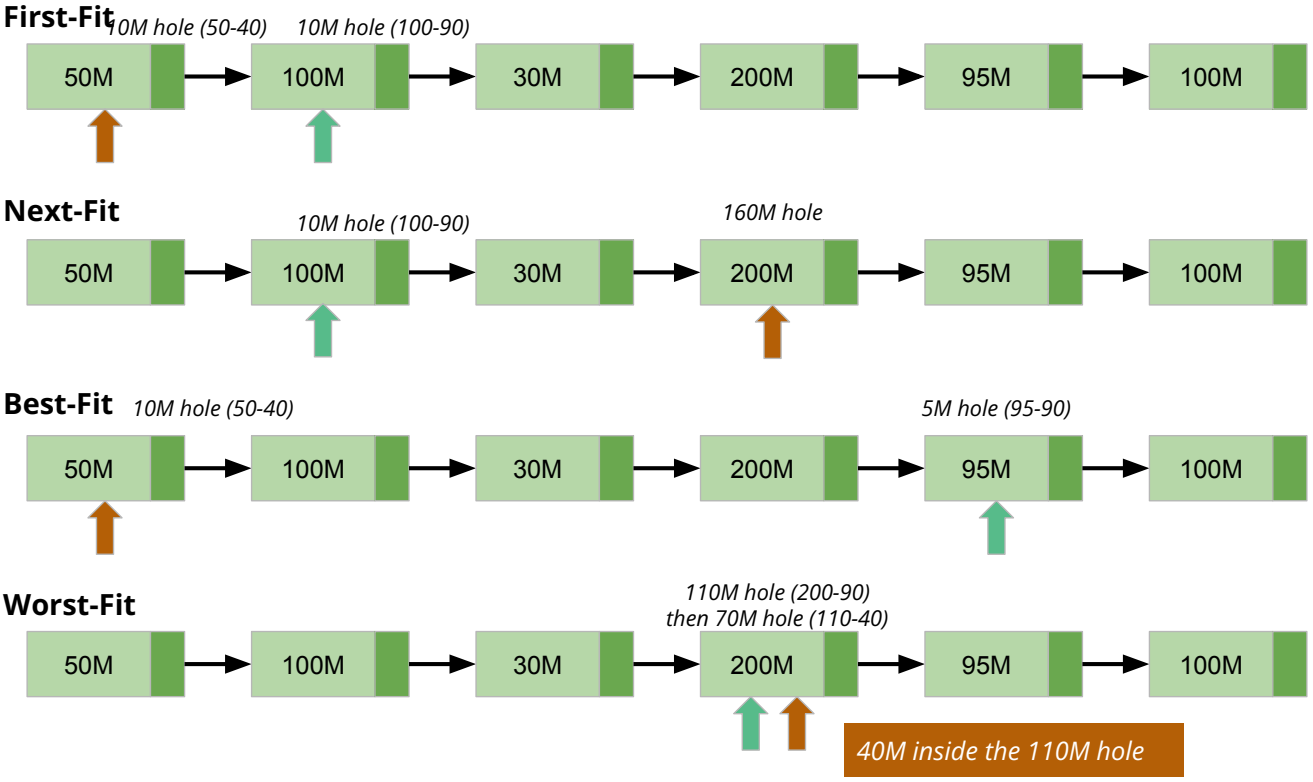
Variable-Sized Partitions: List of Holes



Variable-Sized Partitions: Allocation Algo.

- Allocation Algorithms (how to select a free partition)
  - **First-Fit**: allocate the first hole big enough to accomodate the request. Start searching from the **beginning** of the hole list
  - **Next-Fit**: allocate the first hole big enough to accomodate the request. Start searching from the **last position of previous search**
  - **Best-Fit**: search the entire list (from the beginning) for the hole that gives **minimum leftover**
  - **Worst-Fit**: search the entire list (from the beginning) for the hole that gives **maximum leftover**
- Advantages/Disadvantages?
- Big-O of these algorithms?

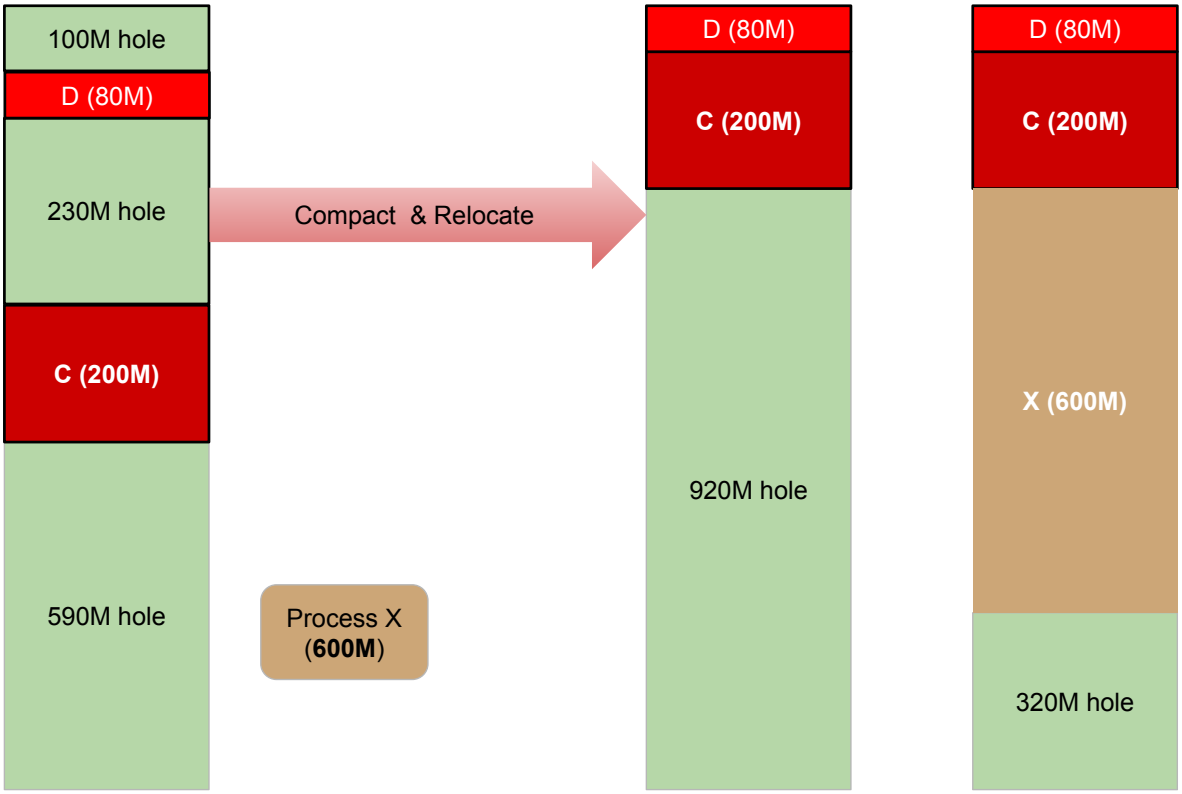
Requests: 90M and then 40M



(In|Ex)ternal Fragmentation

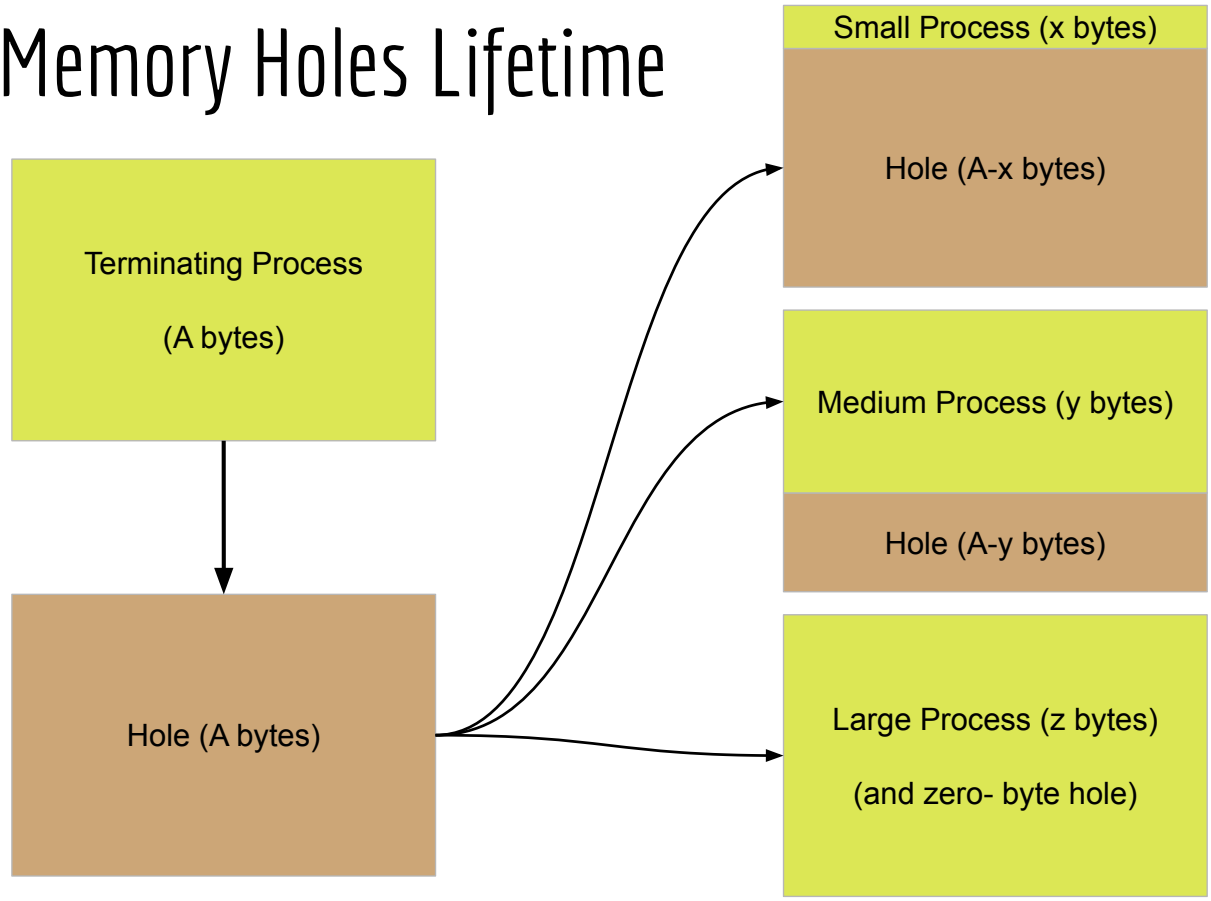
- *Internal Fragmentation*: unused space *within* a partition (of **Fixed-Sized Partitions**)
- *External Fragmentation:* (of **Variable-Sized Partitions**)
  - holes are scattered throughout RAM
  - **The TOTAL size of holes is big enough** to hold the next process
  - **none of the holes** is big enough to hold the next process
  - Solution: run compaction algorithm (consolidates scattered holes into one free contiguous partition)

External Fragmentation & Relocation



50-percent rule:
On average, fragmentation results in
⅓ of RAM wasted in holes

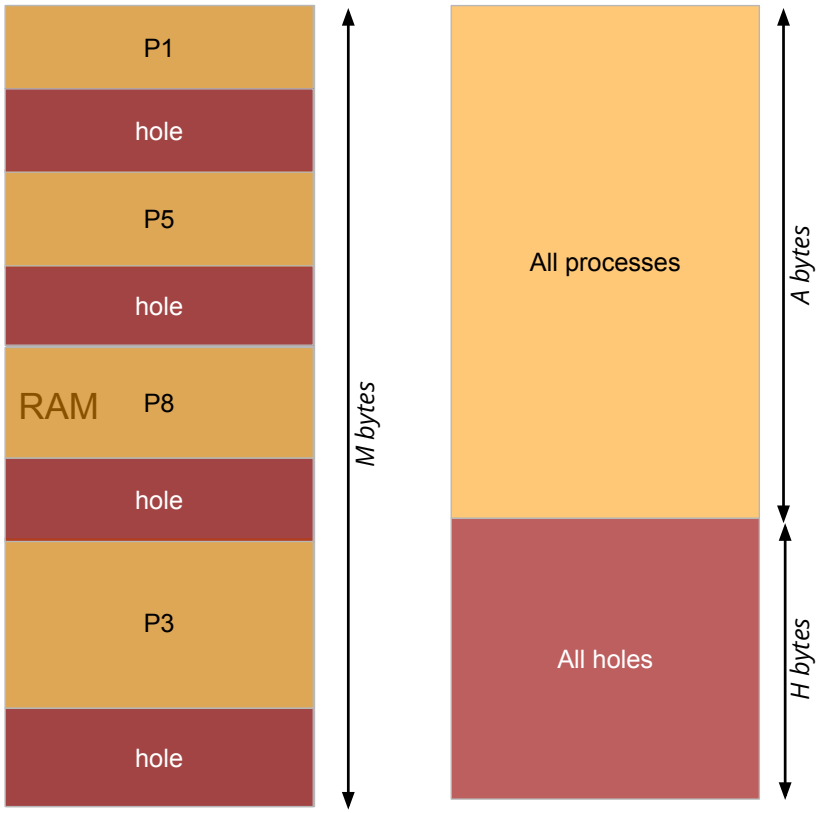
Memory Holes Lifetime



A process of size A bytes turns into holes of size 0 bytes (min) to A bytes (max)

A process of size A bytes turns into holes with average size of ½ A bytes

Wasted RAM Space in Holes?



$$M = A + H$$
$$= A + \frac{1}{2}A$$
$$M = \frac{3}{2}A$$
$$A = \tfrac{2}{3}M \quad \text{and} \quad H = \tfrac{1}{3}M$$

⅓ of RAM is wasted across memory holes