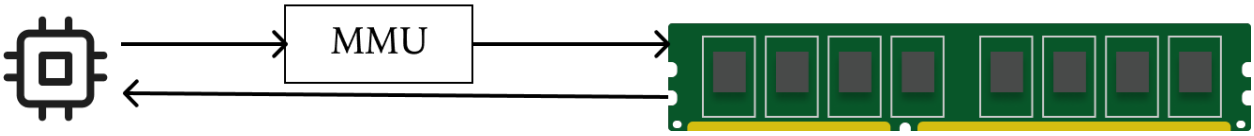
Memory

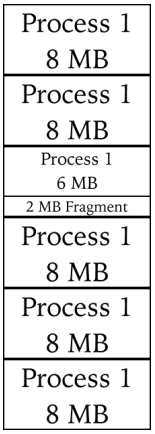
Logical vs. Physical Memory Address Space

- **Logical (Virtual) Memory Address Space**: The memory address space as seen by processes.
- **Physical Memory Address Space**: The actual memory used by the system.
- The **Memory Management Unit (MMU)** translates addresses between logical and physical spaces.



Memory Allocation

Contiguous Allocation of Memory



- Memory is allocated in a continuous block, making it vulnerable to fragmentation:
  - **Internal Fragmentation**: Wasted space within an allocated block.
  - **External Fragmentation**: The sum of internal fragments being large enough to host a process but scattered across memory.

Non-Contiguous Allocation of Memory

1. **Paging**:
   - Divides memory into fixed-size segments called **pages** (logical) and **frames** (physical).
   - **Page Size = Frame Size** to simplify address mapping.
   - **Advantages**: Efficient memory utilization.
   - **Disadvantages**: Still prone to internal fragmentation.
   - Pages and frames are mapped using a **page table**, stored in the **PCB** of each process.
2. **Segmentation**:
   - Divides memory based on the logical structure of a process (e.g., code, stack, data).
   - Uses a **segmentation table** for mapping.
   - **Advantages**: Better representation of process structure.
   - **Disadvantages**: Prone to **external fragmentation** due to multiple reallocations.

Virtual Memory

- **Concept**: Only a portion of a process is loaded into physical memory, while the rest remains on the disk.
- **Benefit**: Allows more processes to be loaded into memory simultaneously, enhancing system utility.

Demand Paging

- Only necessary pages of a process are loaded into memory on demand.
- **Page Fault**: Occurs when a required page is not found in physical memory, prompting the page to be swapped in from the disk.

Thrashing

- **Definition**: Excessive page faults caused by loading too many processes, resulting in frequent disk I/O and decreased CPU utilization.
- **Impact**: Reduces system performance significantly.

Cache Memory

- **Purpose**: Cache memory stores frequently accessed data to reduce memory access time and improve performance.
- **Locality of Reference**: Determines what is stored in the cache.
    - **Temporal Locality**: Recently accessed data is likely to be accessed again soon.
    - **Spatial Locality**: Data located near the recently accessed data is also likely to be accessed.
- **Cache Hits and Misses**:
    - **Cache Hit**: Data is found in the cache.
    - **Cache Miss**: Data is not found, requiring access to main memory.