Contiguous Memory Allocation

Phan Nhật Tân - 23521405
Nguyễn Huy Phước - 23521234
Nguyễn Văn Minh - 23520945
Đỗ Quang Lực - 23520902

Project Overview

This project will involve managing a contiguous region of memory of size MAX where addresses may range from 0...MAX−1. Program must respond to four different requests:

1. Request for a contiguous block of memory
2. Release of a contiguous block of memory
3. Compact unused holes of memory into one single block
4. Report the regions of free and allocated memory

01

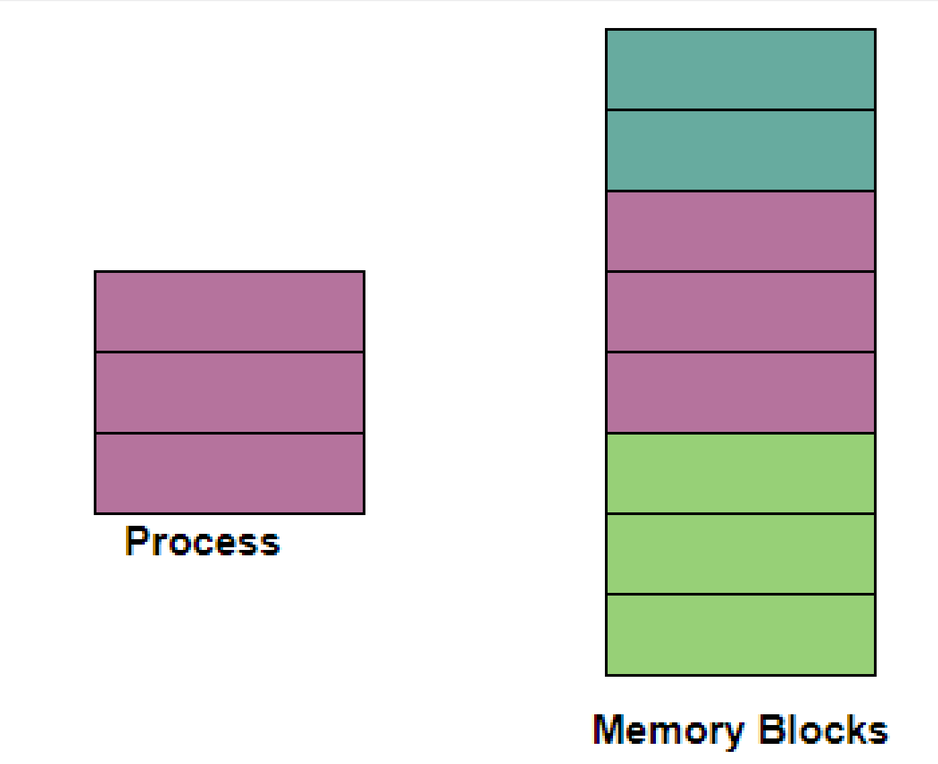
Theoretical Foundation

Project

Advantages and Disadvantages

Definition:

- Contiguous Memory Allocation is a technique where the operating system allocates a single, continuous block of memory to a process.
- Advantages:
  - Simplifies memory management and access.
  - Ideal for systems with limited memory size where fast memory access is critical.

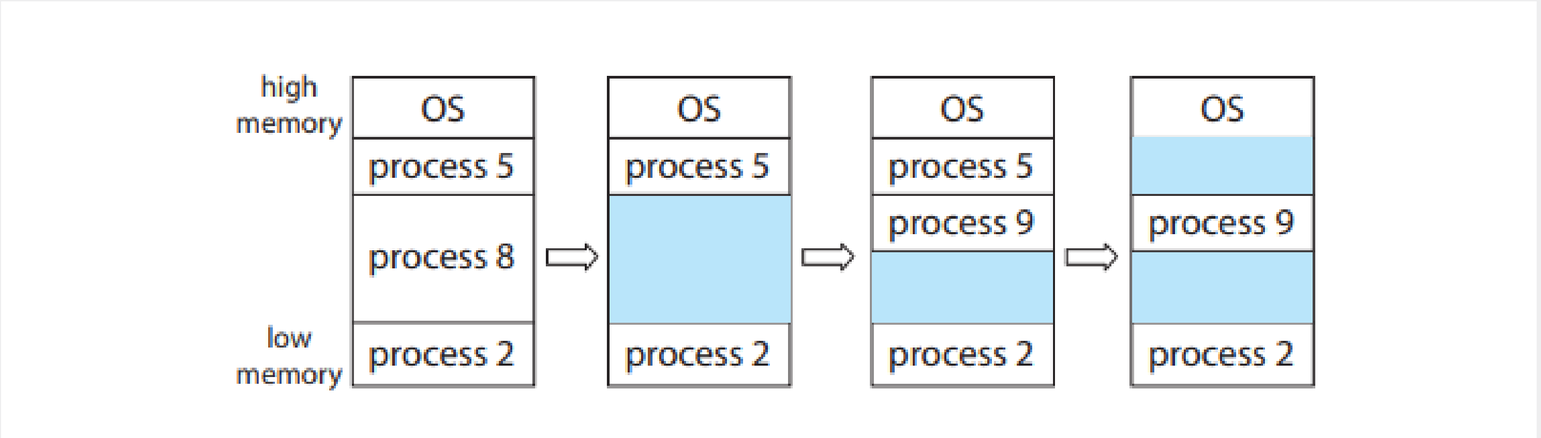


Memory Allocation

- A method of assigning processes to variably sized memory partitions, where each partition holds exactly one process.
- **Management:**
  - The operating system maintains a table to track available and occupied memory.
  - Memory starts as one large block ("hole") and becomes fragmented as processes are added and removed.

Partition Management

- **Initially:** Memory is a single large block ("hole").
- As processes arrive and leave: Holes of varying sizes are created.



Memory Allocation and Release

- **Allocation:**
  - The OS considers process requirements and available memory to allocate space.
- **Release:**
  - Upon completion, memory is freed and reused.
  - Insufficient memory:
    - Reject the process with an error message.
    - Place the process in a wait queue for later allocation.

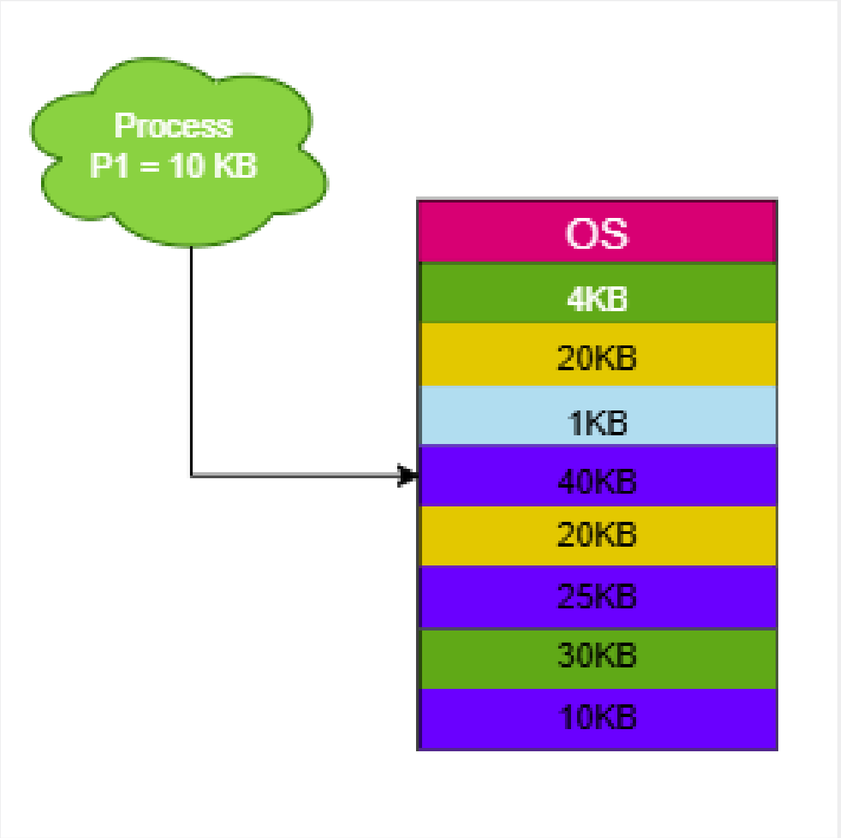
Free Space Management

- **Fragmentation:**
  - Memory holes of varying sizes are scattered throughout memory.
- **Handling Large Holes:**
  - If a hole is too large:
    - Split into allocated and free space.
  - Adjacent free holes:
    - Merged into a larger block.

Allocation Strategies:

- **Strategies to select memory holes:**
  - **First-fit:** Choose the first hole large enough.
  - **Best-fit:** Choose the smallest hole large enough.
  - **Worst-fit:** Choose the largest hole.
- **Comparison:**
  - First-fit and Best-fit are generally faster and more efficient than Worst-fit.

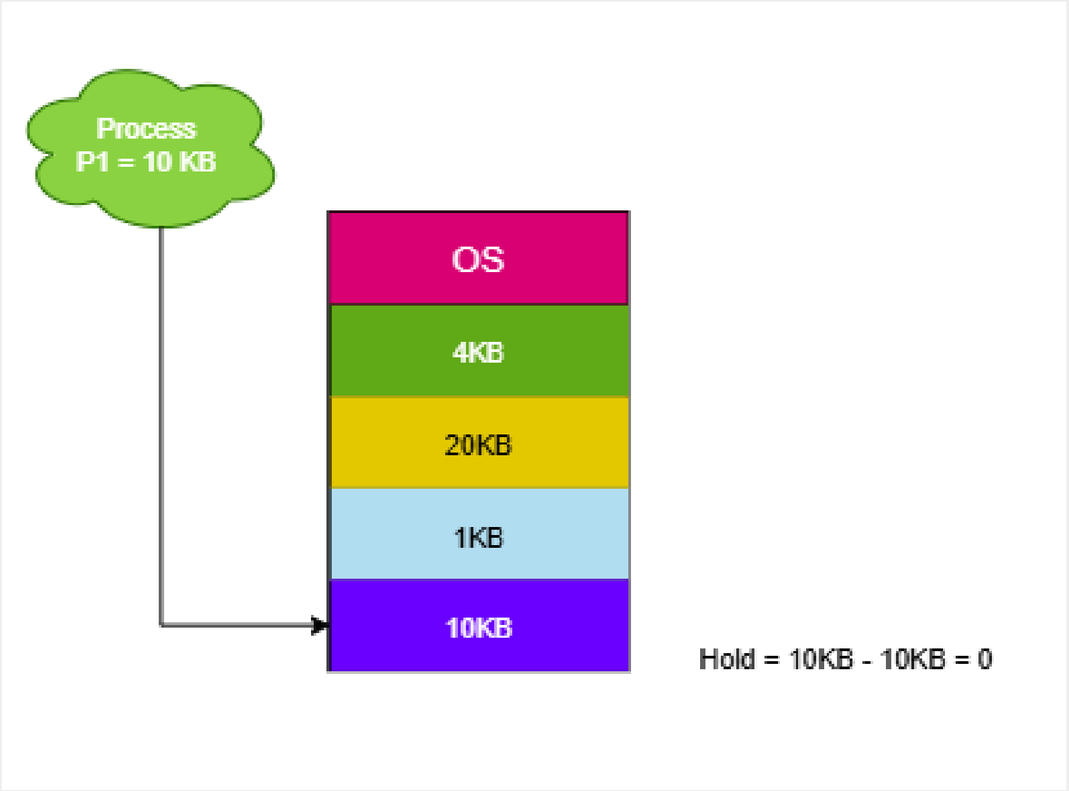
FIRST-FIT



Allocation Strategies:

- **Strategies to select memory holes:**
  - **First-fit:** Choose the first hole large enough.
  - **Best-fit:** Choose the smallest hole large enough.
  - **Worst-fit:** Choose the largest hole.
- **Comparison:**
  - First-fit and Best-fit are generally faster and more efficient than Worst-fit.

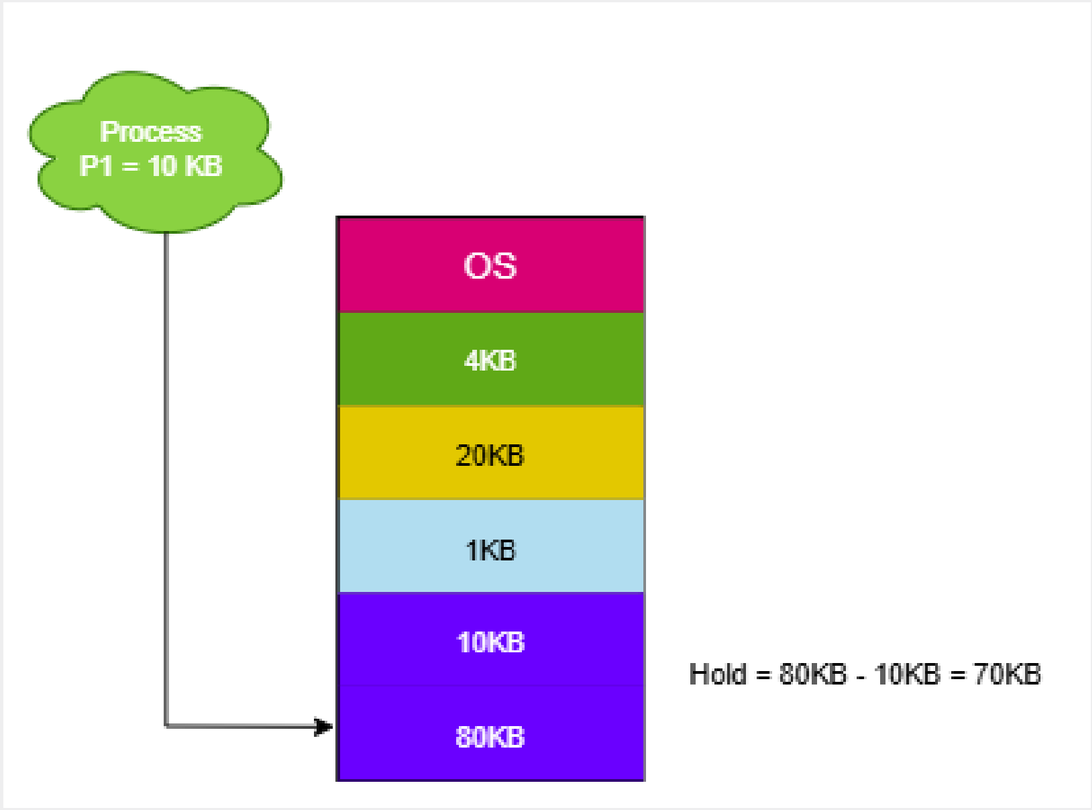
BEST-FIT



Allocation Strategies:

- **Strategies to select memory holes:**
  - **First-fit:** Choose the first hole large enough.
  - **Best-fit:** Choose the smallest hole large enough.
  - **Worst-fit:** Choose the largest hole.
- **Comparison:**
  - First-fit and Best-fit are generally faster and more efficient than Worst-fit.

WORST-FIT



Theoretical Foundation

02

Project

Advantages and Disadvantages

Contiguous Memory Allocation

- **Purpose:**
  - Simulate memory allocation and management processes.
  - Handle requests, releases, and compaction of memory dynamically.
- **Initialization Command:**

  `/allocator 1048576`

  -> Allocates 1 MB (1,048,576 bytes) of memory.
- **User Interaction Prompt:**

  `allocator>`

Commands

- **RQ (Request):**
  - Request memory for a process.
  - Syntax: *RQ <Process> <Bytes> <Strategy>*
    - Example: *RQ P0 40000 W* (Worst-fit)
- **RL (Release):**
  - Release memory allocated to a process.
  - Syntax: *RL <Process>*
    - Example: *RL P0*
- **STAT (Status Report):**
  - Display allocated and unused memory regions.
  - Syntax: *STAT*
- **X (Exit):**
  - Terminate the program.

```
allocator>RQ P0 20 F
allocator>RQ P1 45 W
allocator>STAT
Addresses [0:19] Process P0
Addresses [20:64] Process P1
Addresses [65:199] Unused
allocator>RL P0
allocator>STAT
Addresses [0:19] Unused
Addresses [20:64] Process P1
Addresses [65:199] Unused
allocator>C
allocator>STAT
Addresses [0:44] Process P1
Addresses [45:199] Unused
allocator>X
Bye
```

Theoretical Foundation

Project

03

Advantages and Disadvantages

Advantages:

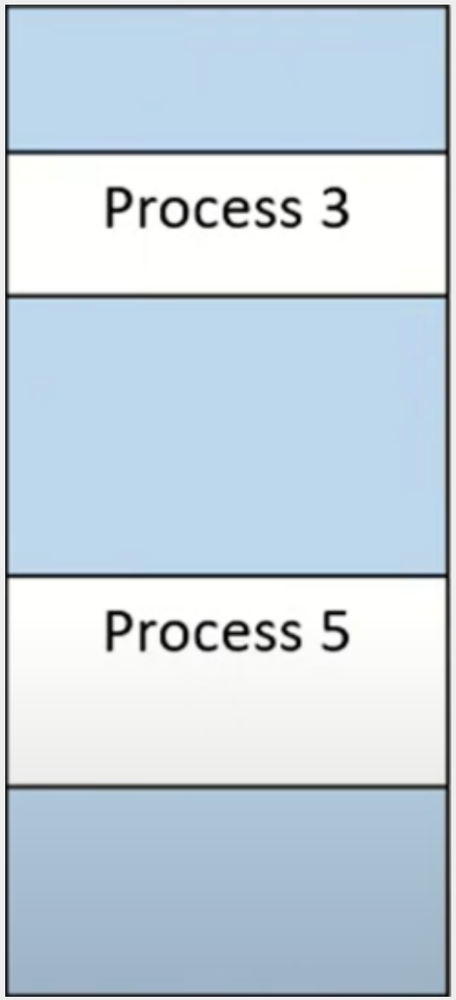
- **Simplicity**
- **Efficiency**
- **Low Fragmentation**

Disadvantages:

- **Limited Flexibility**
- **Memory Wastage**
- **Difficulty Managing Larger Memory Sizes**

External Fragmentation:

- Occurs with both first-fit and best-fit strategies.
- Leads to fragmented, non-contiguous free regions.
- The "50-percent rule": 1/3 of memory can remain unused.



Q&A

Thank You