Memory Allocation
&
Fragmentation Analyzer

By using Best, Worst, First Fit Algorithms

Team:
Shaik Rehan 24WU0102094
Danish 24WU0102099
Anshul 24WU0102088
Devansh 24WU0102072

Under Guidance of:
Prof. Veeresh Biradar

Problem Context: Memory Challenges

⚠ The Core Problem

OS memory allocation strategies directly determine how efficiently RAM is utilized. **Poor allocation decisions lead to both external and internal fragmentation**, resulting in significant wasted memory space and degraded system performance. Understanding these allocation mechanisms is fundamental to optimizing operating system behavior.

30-60%	O(n)	34.86%	2 Types
Execution time spent in memory allocation	Time complexity for allocation algorithms	Speed improvement with head-first Best Fit	Internal & External fragmentation

Project Objectives

1

Simulate Memory Layout

Model the relationship between logical (virtual) addresses generated by CPU and physical addresses in RAM, demonstrating address translation via MMU.

2

Implement Allocation Algorithms

Develop contiguous memory allocation using three strategies: First Fit, Best Fit, and Worst Fit with complete algorithmic implementations.

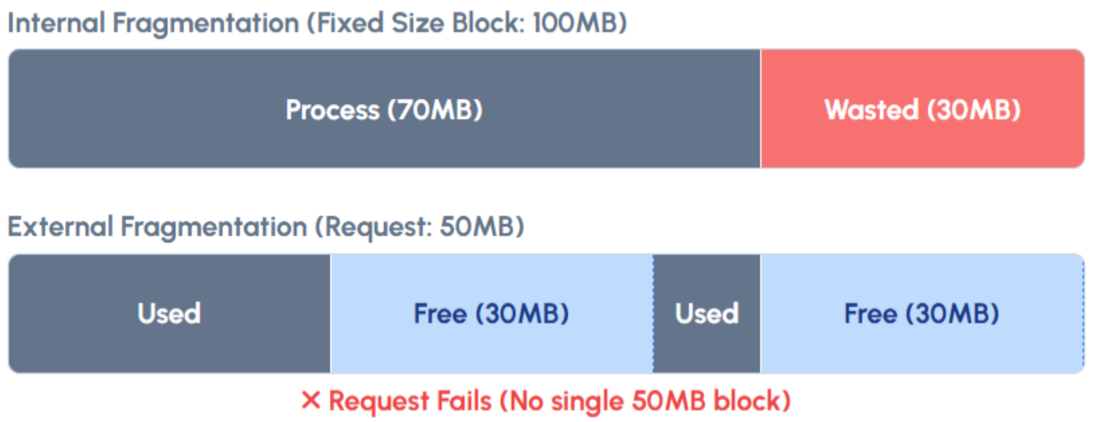
3

Analyze Fragmentation

Generate allocation results with fragmentation statistics: memory block status tables, internal/external fragmentation values, and efficiency comparisons.

Fragmentation Types

- **Internal Fragmentation:** Wasted space *inside* an allocated block. Occurs when the process is smaller than the partition.
- **External Fragmentation:** Wasted space *outside* allocated partitions. Total free space exists but is not contiguous.



Internal vs. External Fragmentation

FEATURE	INTERNAL FRAGMENTATION	EXTERNAL FRAGMENTATION
Definition	Wasted space *within* an allocated block.	Wasted space *between* allocated blocks.
Cause	Process size < Partition size.	Non-contiguous free memory.
Occurrence	Fixed Partitioning techniques.	Dynamic Partitioning techniques.
Solution	Best Fit Algorithm / Variable Partitioning.	Compaction / Paging / Segmentation.

Internal: Paying for a whole pizza but only eating one slice.

External: Having enough empty seats in a theater, but they aren't together.

Memory Block Status & Fragmentation

LOCATION	BLOCK SIZE	JOB #	JOB SIZE	STATUS	INTERNAL FRAG.
10567	30 K	J1	20 K	Busy	10 K
30457	50 K	J4	50 K	Busy	None
300875	200 K	J2	200 K	Busy	None
809567	700 K	J3	500 K	Busy	200 K

TOTAL AVAILABLE
980 K

TOTAL USED
770 K

TOTAL WASTED (INTERNAL)
210 K

1. First Fit Allocation

Method: Allocate the *first* hole that is big enough.

- Scanning starts from the beginning of memory.
- **Advantage:** Very fast. Simple to implement.
- **Disadvantage:** Creates fragmentation at the start of memory.

2. Best Fit Allocation

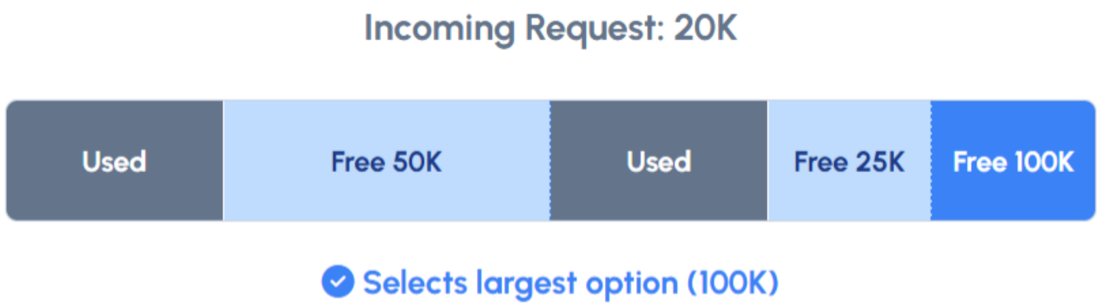
Method: Allocate the *smallest* hole that is big enough.

- Must search the entire list.
- **Advantage:** Minimizes wasted space for the current request.
- **Disadvantage:** Slow. Creates very small, useless holes.

3. Worst Fit Allocation

Method: Allocate the *largest* available hole.

- Must search the entire list.
- **Advantage:** Produces the largest leftover hole, which is more likely to be useful.
- **Disadvantage:** Slow. Breaks up large contiguous blocks quickly.



Allocation Efficiency Comparison

ALGORITHM	SPEED	STORAGE UTILIZATION	FRAGMENTATION ISSUE
First Fit	**Fastest**	Good (Usually)	Concentrates at start
Best Fit	Slow	**Best (Initially)**	Creates tiny holes
Worst Fit	Slow	**Poor**	Breaks large blocks

Performance Metrics

Key Takeaway: First fit is generally the best general-purpose algorithm due to its speed. Best fit is efficient for space but computationally expensive to search the whole list every time.

Methodology: Analysis Approach

Evaluation Framework:

- **Workload Simulation:** We simulate a sequence of job requests (allocation) and job completions (deallocation) to observe memory behavior.
- **Key Metrics:**
  - ⏱ **Search Speed:** Blocks traversed.
  - 🧩 **Fragmentation:** Space wasted.
  - ✅ **Success Rate:** Allocation count.