

A Study on Reliable Memory Allocation based on FIRST-FIT & BEST-FIT

Dr. N Chinnaiyan, Vishwanath Y, C N Ravi

Associate Professor, Information Science and Engineering Department, New
Horizon College of Engineering, Bangalore, India,
Chinnaiyanr@newhorizonindia.edu

Asha Rani Borah, Sridevi S

Assistant Professor, Information Science and Engineering Department, New
Horizon College of Engineering, Bangalore, India,
asharb@newhorizonindia.edu

Received 20, December 2015 | Accepted 29, December 2015

Abstract—Memory Management is one of the services provided by operating system which is needed for optimize memory usage of the available memory in a computer system. Best-fit uses the best memory block based on the process memory request. First-fit in this scheme we check the blocks in a sequential manner which means we pick the first process then compare its size with first block size.

I. INTRODUCTION

BEST-FIT memory allocation makes the best use of memory space but slower in making allocation. The BF deals with allocating the smallest free partition which meets the requirement of the requesting process [1]. This algorithm first searches the entire list of free partitions and considers the smallest hole that is adequate. It then tries to find a hole which is closed to actual process size needed. It chooses the block that is closet in size to the request. It allocates the smallest hole(block) that is big enough.

FRIST-FIT approach is to allocate the first free partition or hole large enough which can accommodate the process. It finishes after finding the first suitable free partition. The blocks is in sequential manner which means we pick the first process then compare its size with first block size[2]. If it is less than size of block it is allocated otherwise we move to second block and so on. Allocate first hole that is big enough. FF begins to scan memory from the beginning and chooses the first available block that is large enough.

First-Fit and Best-Fit are among the most popular algorithms for dynamic memory allocation. First fit is generally faster. On the average, FF searches half of the free list per allocation. BF searches the entire free list to find the smallest free block large enough. These algorithms suffer from external fragmentation.

II. PROPOSED SYSTEM

- Memory Management is the functionality of an OS which handles or manages primary memory and moves processes back and forth between main memory and disk during execution[3].
- Memory management keeps track of each and every memory location, regardless of either it is allocated to some process or it is free.
- It checks how much memory is to be allocated to processes, it decides which process will get memory at what time, it tracks whenever some memory gets freed or unallocated and correspondingly it updates the status[4].
- The OS keeps the record of memory allocation and deallocation in the form of table. Initially all the memory is available for user processes. When the process arrives in the system and needs memory, OS search for large hole for this process[5].

- Hole is one large block of free available memory. If any free hole is found, process is allocated to the free hole of memory as is needed. After allocating number of holes for the processes, a set of various size holes is scattered throughout memory at any given time.
- When a process arrives and searches for (memory) set of holes. But the holes must be large enough to accommodate the process[6]. The set of holes is searched to determine which hole is best to allocate.

III. MODULES

A. FIRST-FIT MODULE

The user can enter how many blocks and size of each blocks accordingly and how many processor with size and can see how the FIRST-FIT memory allotment is done

B. BEST-FIT MODULE

The user can enter the blocks with size and the number of processor with size accordingly and the user will be able to view how the BEST-FIT memory allocation is done[7].

IV. EASE OF USE

User can view details of how the processor are allotted according to their size. It is convenient for users as this system provides accurate allotment and description of the processor[8]. The project is flexible to be used. User can view different allotment of process in each blocks according to its size. Hence it saves time, efforts and cost by achieving quantified objectives with effectiveness, efficiency and satisfaction in a qualified context of use[9].

V. FEATURES AND LIMITATIONS

A. Features:

- FF: Fastest Algorithm because it searches as little as possible[10].
- BF: Memory utilization is much better than first-fit as it searches the smallest free partition first available[11].

B. Limitations:

- FF: The remaining unused memory areas left after allocation become waste if it is too smaller. Thus request for larger memory requirement cannot be accomplished[12].
- BF: It is slower and may even tend to fill up memory with tiny useless holes.

VI. APPENDIX

A. Abbreviations and Acronyms

BF - Best Fit.

FF- First Fit

OS – Operating System

B. Figures and Tables

This table shows the output (result) for the users.

BEST-FIT

PROCESS NO.	PROCESS SIZE	BLOCK NO.	BLOCK SIZE	FRAGMENT
1	150	1	200	50
2	100	2	100	0
3	10	3	200	190

FIRST-FIT

BLOCK NO.	PROCESS SIZE	PROCESS NO.	BLOCK SIZE
1	150	1	200
2	100	2	100

VII. REFERENCES

- [1] OPERATING SYSTEM BOOK WILLIAM STALLING PDF
- [2] "THE C PROGRAMMING LANGUAGE" BY BRIAN W. KERNIGHAN DENNISM.RITCHIE
- [3] "PROGRAMMING IN ANSI C" BY E. BALAGURUSAMY
- [4] "C PROGRAMMING: A MODERN APPROACH" BY K. N. KING
- [5] "DATA STRUCTURE USING C" BY A. K. SHARMA
- [6] "DATA STRUCTURES AND PROGRAM DESIGN IN C" BY KRUSE ROBERT
- [7] "DATA STRUCTURES AND ALGORITHM ANALYSIS IN C" BY MARK ALLEN WEIS
- [8] "ALGORITHMS IN C PARTS 1-4" BY ROBERT SEDGEWICK
- [9] "DATA STRUCTURES AND ALGORITHMS " BY ALFRED V. AHO AND JEFFREY D. ULLMAN
- [10]"DATA STRUCTURES AND ALGORITHMS MADE EASY: DATA STRUCTURE AND ALGORITHMIC PUZZLES " BY NARASIMHAKARUMANCHI
- [11]"DATA STRUCTURES: A PSEUDOCODE APPROACH WITH C " BY RICHARD F. GILBERG AND BEHROUZ A. FOROUZAN
- [12]"OPERATING SYSTEM CONCEPTS" BY AVISILBERSCHATZ, PETER GALVIN, GREG GAGNE