VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CSE2005 OPERATING SYSTEM

STORAGE MANAGEMENT SYSTEM

PROJECT DOCUMENTATION

UNDER THE GUIDENCE OF SHAIK NASEERA

F1-SLOT

TEAM MEMBERS :

SHIVAM KUMAR PANDEY: 17BCB0082

SHIVANSH MISHRA: 17BCB0112

NASAKA RAJA VIVEK: 17BCE0397

KOUSHIK SAHA: 17BCE2417

SAURABH AMBAR: 17BCI0058

CONTENT:

1. ABSTRACT

Memory management is applied on computer memory as a method of resource management. The primary necessity of memory management is to provide ways to dynamically allocate parts of memory to programs at their request and needs, and free it for reuse when no longer needed. Using the concepts of memory management and task scheduling, we aim to find and implement new algorithms for memory and file allocation.

1.First Fit: The first hole that is big enough is allocated to program.

2.Best Fit: The smallest hole that is big enough is allocated to program.

3.Worst Fit: The largest hole that is big enough is allocated to program.

We aim to study these algorithms along with new algorithms such as Buddy fit, Last fit, Indexed fit, Slab fit and Next fit.

2. INTRODUCTION

Memory allocation is a process by which computer programs and services are assigned with physical or virtual memory space.

Important components of it are:

In contiguous memory allocation each process is contained in a single contiguous block of memory. Memory is divided into several fixed size partitions. Each partition contains exactly one process. When a partition is free, a process is selected from the input queue and loaded into it. The free blocks of memory are known as *holes*. The set of holes is searched to determine which hole is best to allocate.

Memory protection is a phenomenon by which we control memory access rights on a computer. The main aim of it is to prevent a process from accessing memory that has not been allocated to it. Hence prevents a bug within a process from affecting other processes, or the operating system itself, and instead results in a segmentation fault or storage violation exception being sent to the disturbing process, generally killing of process.

Segmentation is another memory management scheme that supports the user-view of memory. Segmentation allows breaking of the virtual address space of a single process into segments that may be placed in non-contiguous areas of physical memory.

Memory allocation is the process of reserving a partial or complete portion of computer memory for the execution of programs and processes.

We have already studied First Fit, Worst Fit and Best fit algorithms in class. We aim to find newer algorithms to find newer ways to allocate memory

We identify some new algorithms like **Buddy Fit, Last Fit, Indexed Fit, Slab fit** and **Next-Fit.**

3. LITERATURE SURVEY

1.Efficient and agile storage management in software defined environments
https://pdfs.semanticscholar.org/73ce/1e41c2e0a97a929b9b9999c5daabd7a037b4.pdf

2.Towards Global Storage Management and Data Placement
http://john.e-wilkes.com/papers/HPL-SSP-2001-1.pdf

3.A study of real time memory management: Evaluating Operating system's performance
http://journals.bg.agh.edu.pl/AUTOMAT/2013.17.1/automat.2013.17.1.29.pdf

4.Memory management

https://en.wikipedia.org/wiki/Memory_management

5.Stack-based memory allocation

https://en.wikipedia.org/wiki/Stack-based_memory_allocation

6.Stacks and heaps

https://en.wikibooks.org/wiki/Memory_Management/Stacks_and_Heaps

7.Queue Memory management

https://lavag.org/topic/10400-queue-memory-management/

8.https://users.cs.cf.ac.uk/Dave.Marshall/C/node11.html

9.Storage class memory: Technology, systems and applications
https://ieeexplore.ieee.org/document/7480060/

10.Virtual Memory

http://web.stanford.edu/~ouster/cgi-bin/cs140-spring18/pintos/pintos_4.html

11.Virtual Memory- Operating Systems

https://www.cs.princeton.edu/courses/archive/fall04/cos318/projects/5.html

12.Operating Systems Principles: Virtual Memory and Paging

http://www.cs.princeton.edu/courses/archive/spr03/cs217/lectures/Paging.pdf

13.Memory Management

http://web.cs.ucla.edu/classes/spring16/cs111/slides/06_vmem.pdf

14.Memory Management

http://web.cs.ucla.edu/classes/spring17/cs111/slides/lecture_5.pdf

15.https://inst.eecs.berkeley.edu/~cs162/sp14/

16.Virtual Memory Management

https://www.cc.gatech.edu/~rama/CS2200-External/projects/p3/assignment.html

17.C - Memory Management
https://www.tutorialspoint.com/cprogramming/c_memory_management.htm

18. Memory Management

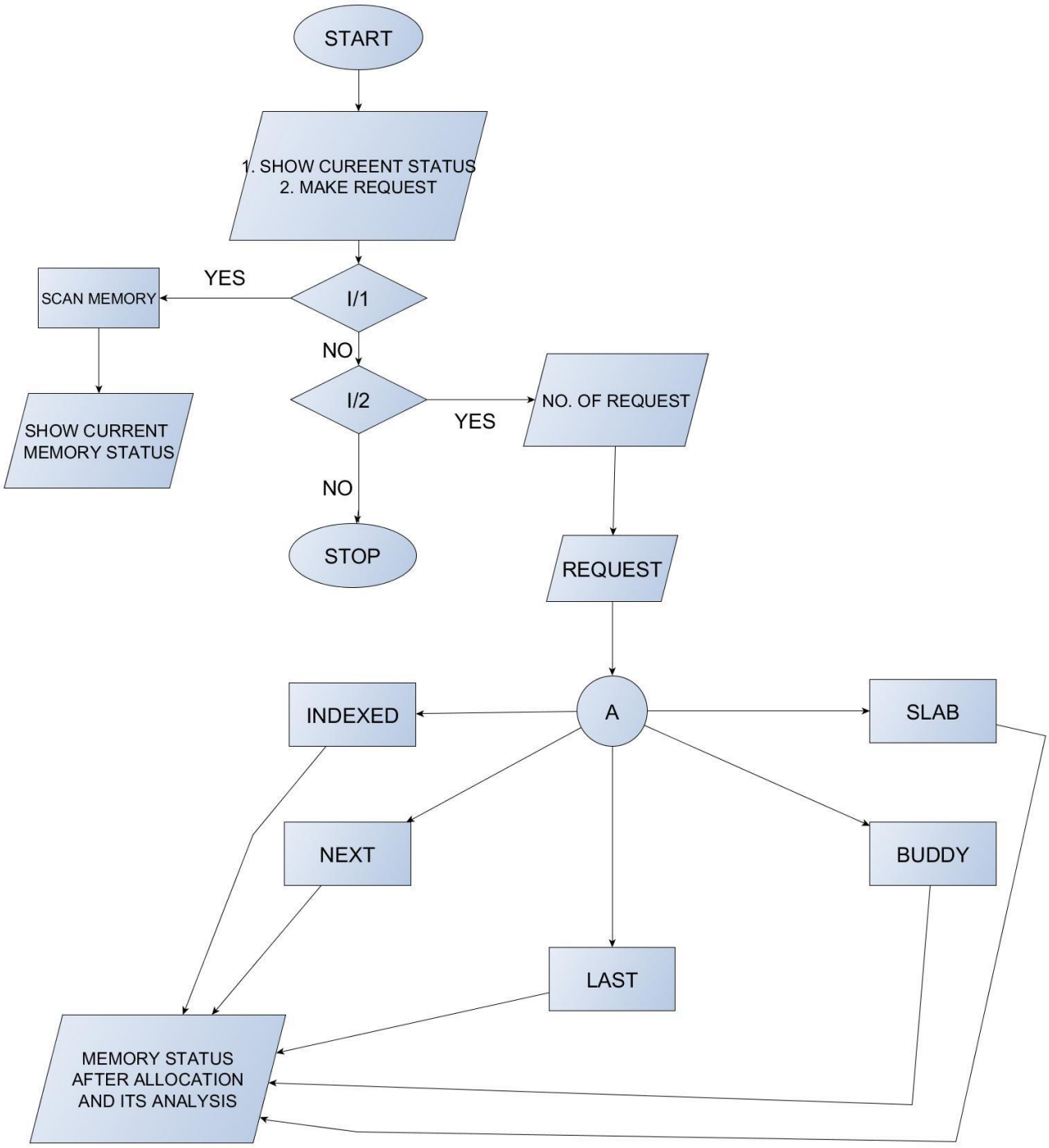
http://ijicse.in/wp-content/uploads/2017/06/30.pdf

19. Memory Management:

http://www.ijirt.org/master/publishedpaper/IJIRT100966_PAPER.pdf

20. Dynamic Memory Allocation
https://www.researchgate.net/publication/265166374_Dynamic_Memory_Allocation_Role_in_Memory_Management

4. SYSTEM DESIGN/ARCHITECTURE



Algorithm:

1. START
2. SHOW CURRENT STATUS
3. MAKE REQUEST
4. IF INPUT 1 IS YES THEN SCAN MEMORY AND THEN SHOW THE CURRENT MEMORY STATUS.
5. IF INPUT 1 IS NO THEN ENTER INPUT 2.
6. IF INPUT 2 IS NO THEN STOP.
7. IF INPUT 2 IS YES THEN INPUT NO. OF REQUEST.
8. THEN PROCESS THE REQUEST TO A.
9. ACCORDING TO REQUEST 'A' WILL PROCESS EITHER INDEX, SLAB, BUDDY, NEXT, LAST.
10. THEN DISPLAY MEMORY STATUS AFTER ALLOCATION AND ITS ANALYSIS.

5. PROPOSED ALGORITHMS

1. Next Fit:

Next fit is a modified version of 'first fit'. It begins as first fit to find a free partition but when called next time it starts searching from where it left off, not from the beginning. This policy makes use of a roving pointer. The pointer roves along the memory chain to search for a next fit.

2. Buddy Fit:

The buddy system is a memory allocation and management algorithm that manages memory in power of two increments. Assume the memory size is 2U, suppose a size of S is required. When the allocator receives a request for memory, it rounds the requested size of S up to a permitted size 2m, and returns the first block from that size's free list. If the free list for that size is empty, the allocator splits a block from a larger size and returns one of the pieces, adding the other to the appropriate free list. When blocks are recycled, there may be some attempt to merge adjacent blocks into ones of a larger permitted size (coalescence). To make this easier, the free lists may be stored in order of address. The main advantage of the buddy system is that coalescence is cheap because the "buddy" of any free block can be calculated from its address.

3. Last fit:

The last fit system is the reverse of the first fit system. Here the memory is allocated to the last available memory slab. This helps to keep more of the unallocated fragments in the beginning addresses of the storage, which helps faster allocation and retrieval for sequential access methods, for which files are stored afterwards.

4. Indexed Fit:

Here each memory/file is stored in random indices. Then a main index is created for each file which contains pointers to various other indices. Index is searched sequentially and its pointer is used to access the file directly. This makes it easier to locate each file.

5. Slab Allocation Fit:

For this, we propose of allocating data with units having sequential address, from the first allocation itself. As the following requests are served with allocation next, immediate addresses, no fragmentation occurs. The essential inspiration for slab allocation is that the introduction and annihilation of bit information items can really exceed the expense of dispensing memory for them. The overhead expenses of instatement can result in huge execution drops.

6. RESULTS (COMPARITIVE STUDY)

After the study and implementation, we can say that slab allocation turns out to be the most preferable algorithm for storage allocation, as the sequential method opted by it prevents fragmentation also with high allocation and retrieval speed. But this method is not ideal for larger storages, as continuous writing can be time consuming on a larger scale, and one cannot ensure availability of such continuous segments in a randomly partitioned disk from user. So the slab allocation is preferable on a small scale storage, like cache or kernel memory.

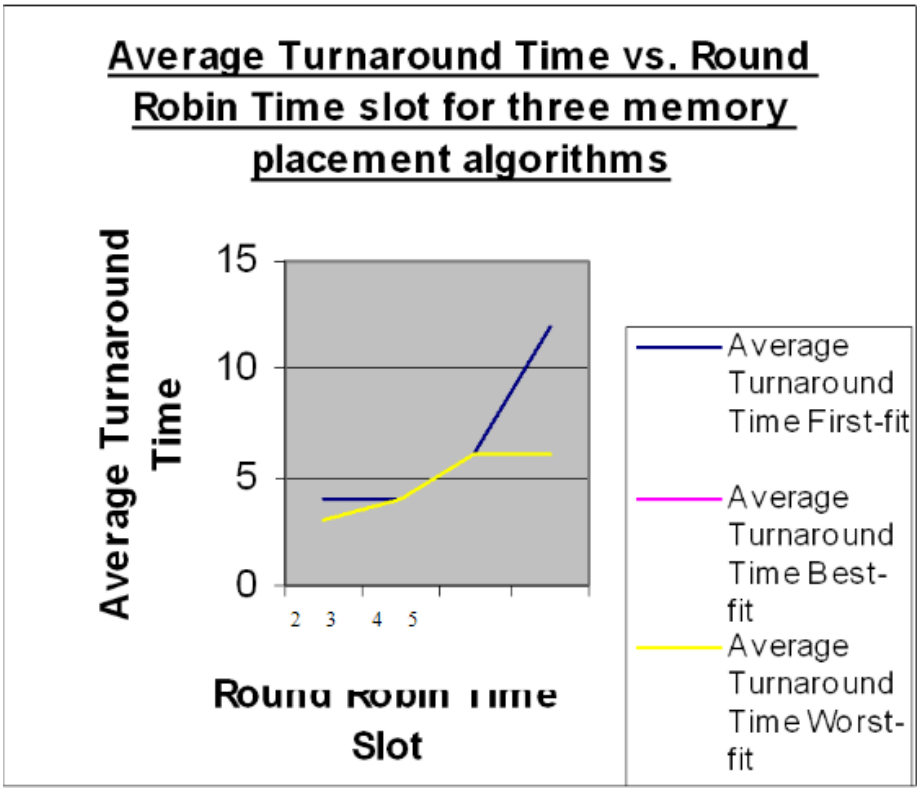
Indexed fit has the fastest allocation speed, thanks to its randomised algorithm. It leads to high amount of fragmentation, although the allocation performance is not affected by it. But it's not ideal for large files as a single index will not be able to hold the pointers to all the addresses.

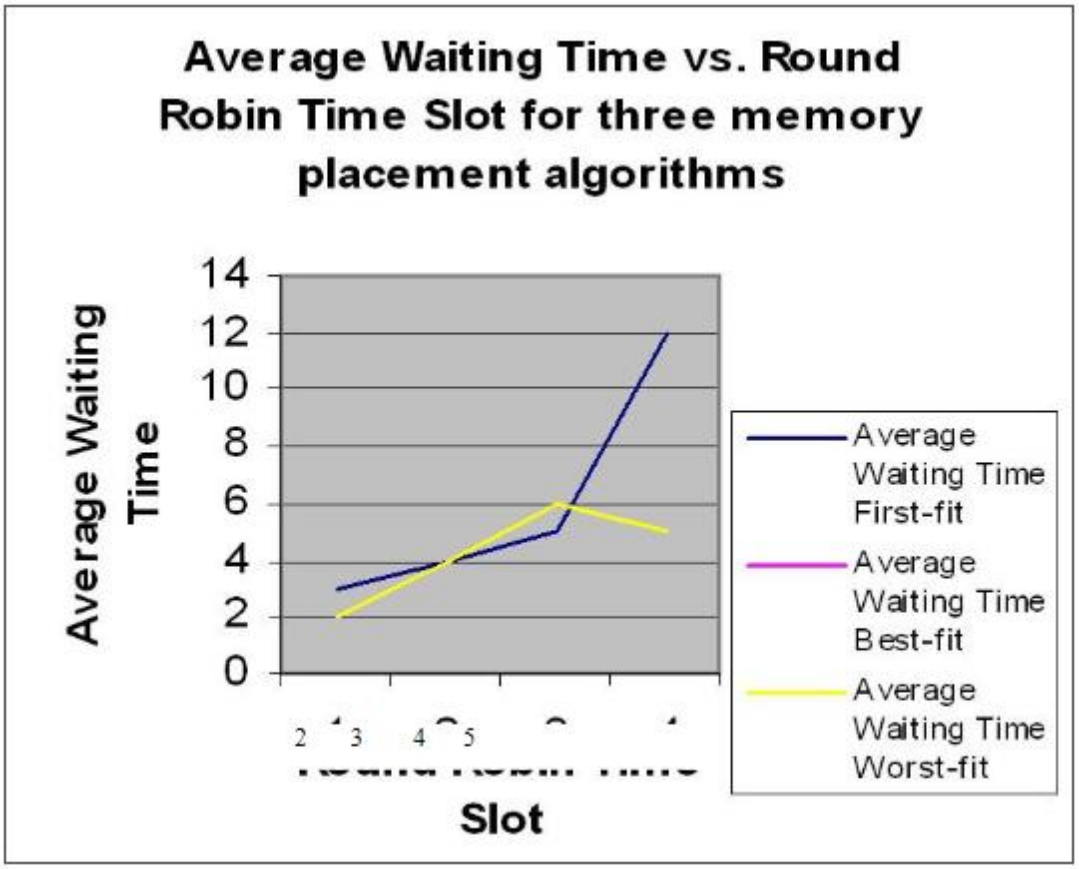
Buddy fit can be said to be a balance between to, with gives a decent control on fragmentation and better allocation than indexed fit, but on the cost of time invested in calculating the optimised size for storage. Also, as we related all the sizes in powers of 2, it's a bit easier to manage the memory and its allocation. Buddy fit is affected when a very small allocation request is made on a very big segment (e.g. 2 kb on a 128 kb space).

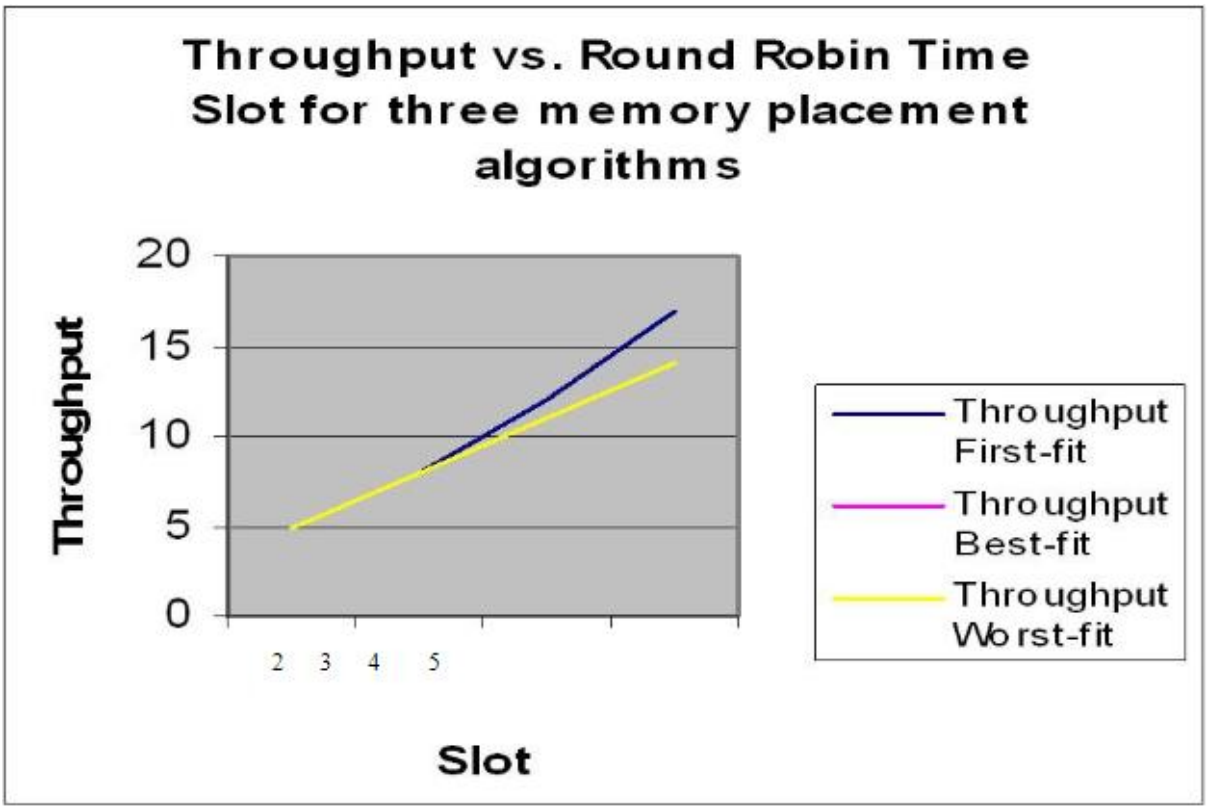
Performance wise, not much difference can be seen between last fit and next fit, although next fit can be seen to be better in terms of uniform allocation, when compared to last fit. The allocation consumes time in storage traversal in both the algorithms. But as we said, last fit leaves more of the unallocated space in initial space of the memory, so the next allocations can tend to shorten their domain for allocation, which can result in performance improvement.

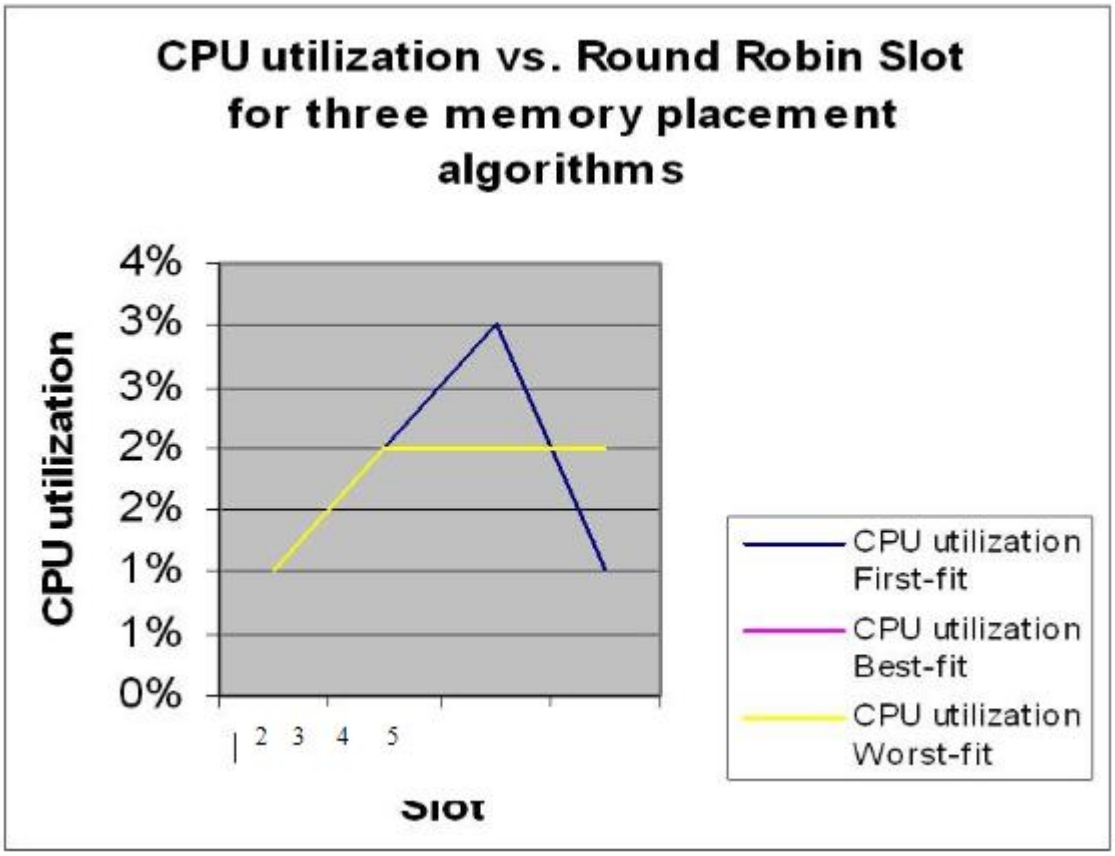
Except for slab allocation, all other algorithms can work over a fragmented storage area efficiently.

Next fit	Buddy fit	Last fit	Indexed fit	Slab fit
1. Not much difference can be seen between last fit and next fit. 2. Better than last fit in terms of memory allocation	1. Buddy fit is balanced 2. Sizes in powers of 2, it's a bit easier to manage the memory	1. Not much difference can be seen between last fit and next fit. 2. Worse than next fit for allocation	1.Indexed fit has the fastest allocation speed, thanks to its randomised algorithm 2. Not ideal for large files	1. Prevents fragmentation also with high allocation and retrieval speed. 2. Not ideal for larger storages




Average Waiting Time vs. Round Robin Time Slot for three memory placement algorithms
Average Waiting Time
14
12
10
8
6
4
2
0
2 3 4 5
Slot
Average Waiting Time First-fit
Average Waiting Time Best-fit
Average Waiting Time Worst-fit


Throughput vs. Round Robin Time Slot for three memory placement algorithms
Throughput
20
15
10
5
0
2 3 4 5
Slot
Throughput First-fit
Throughput Best-fit
Throughput Worst-fit



7. CONCLUSION

After comparing all the algorithms, we can conclude that SLAB allocation turns out to be the most preferable algorithm for storage allocation.

- Buddy fit is the most balanced.
- Indexed fit is the fastest.
- Depending of the situation, we use different algorithms.

8. REFERENCES

- Second, in the UNIX world, we usually over-provision disk space, by carving volumes out of a larger disk, so that there is unallocated disk space to grow the volumes. Disk volume growth is usually a manual system administrator task."
- Shihabur Rahman Chowdhury*, Constantin Adam* , Frederick Wu*, John Rofrano*, and Raouf Boutaba†*: IBM TJ Watson Research Centre.
- http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.708.3485&rep=rep1&type=pdf
- "For years, the storage industry has been facing challenges in managing heterogeneous storage devices which typically have their own management interfaces and protocols."
- Efficient and agile storage management in software defined environments.
- https://pdfs.semanticscholar.org/73ce/1e41c2e0a97a929b9b9999c5daabd7a037b4.pdf
- Yale University Resource http://codex.cs.yale.edu/avi/os-book/OS9/practice-exer-dir/8-web.pdf
- Lecture Notes for May 9, 2005: By David Law, Dennis Nguyen, Young Wook Choi http://read.seas.harvard.edu/~kohler/class/05s-osp/notes/notes10.html
- Stack data management for Limited Local Memory (LLM) multi-core processors https://ieeexplore.ieee.org/document/6043275/
- Computational Study of Static and Dynamic Memory Allocation http://ijarcsse.com/Before_August_2017/docs/papers/Volume_5/8_August2015/V5I80294.pdf