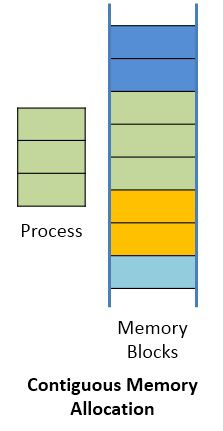
**Operating Systems**

**Project Report**

**On**

**Contiguous Memory Allocation**

****

Project By -

Ayush Vashistha – 169108037

Himanshu Dahiya - 169108060

**Introduction**

Contiguous memory allocation is a classical memory allocation model that assigns a process consecutive memory blocks (that is, memory blocks having consecutive addresses).Contiguous memory allocation is one of the oldest memory allocation schemes. When a process needs to execute, memory is requested by the process. The size of the process is compared with the amount of contiguous main memory available to execute the process. If sufficient contiguous memory is found, the process is allocated memory to start its execution. Otherwise, it is added to a queue of waiting processes until sufficient free contiguous memory is available.

The contiguous memory allocation scheme can be implemented in operating systems with the help of two registers, known as the base and limit registers. When a process is executing in main memory, its base register contains the starting address of the memory location where the process is executing, while the amount of bytes consumed by the process is stored in the limit register. A process does not directly refer to the actual address for a corresponding memory location. Instead, it uses a relative address with respect to its base register. All addresses referred by a program are considered as virtual addresses. The CPU generates the logical or virtual address, which is converted into an actual address with the help of the memory management unit (MMU). The base address register is used for address translation by the MMU. Thus, a physical address is calculated as follows:

Physical Address = Base register address + Logical address/Virtual address

One disadvantage of contiguous memory allocation is that the degree of multiprogramming is reduced due to processes waiting for free memory.

**Overview**

Implementing contiguous memory allocation with a case of multiple threads. Considering a large array shared among the threads considered as a hole by main thread, multiple threads are invoked. Each thread requests for memory in the array, main thread allocates memory using best fit, first fit and worst fit. The comparison for efficiency between the different algorithms is performed.

**Main Thread Algorithm**

1. Create 3 arrays, 1 for first fit, 1 for worst fit and 1 for best fit with some memory allocated initially.

2. Create n number of threads for n processes and pass pipe to each thread for communication.

3. Wait for request from a process.

4. When request comes, apply first fit, worst fit and best fit algorithm, to array1, 2, and 3 respectively.

5. If all requests done, done print summary and exit

6. Otherwise, go to step 3.

**Process Thread Algorithm**

1. Generate random number of bytes from 50-200.

2. Send request to main thread using pipe.

3. Exit.

**Best Fit Algorithm**

1. Find the minimum block size that can be allocated.

2. If found, allocate the block size.

3. Otherwise, print message and continue.

**Worst Fit Algorithm**

1. Find the maximum amount of free block size that can be allocated.

2. If found, allocate the block size.

3. Otherwise, print message and continue.

**First Fit Algorithm**

1. Find the first block of free space that can be allocated.

2. If found, allocate the block size.

3. Otherwise, print message and continue.

