

Experiment:21-Develop a C program to implement the worst fit algorithm of memory management.

Aim:

The aim of this program is to implement the Worst Fit memory allocation algorithm in C. In this algorithm, the process is allocated to the largest available memory block, aiming to leave the smallest leftover space in the memory.

Procedure:

1. The system has a set of memory blocks of varying sizes and a set of processes that need to be allocated to the memory blocks.
2. The Worst Fit algorithm selects the largest available block for each process.
3. If the process can fit into the memory block, the block is allocated, and the remaining space in that block is updated.
4. If no suitable block is available, the process is marked as unallocated.

Code Implementation:

```
#include <stdio.h>
```

```
void worstFit(int blockSize[], int m, int processSize[], int n) {
```

```
    int allocation[n];
```

```
    for (int i = 0; i < n; i++) {
```

```
        allocation[i] = -1;
```

```
    }
```

```
    for (int i = 0; i < n; i++) {
```

```
        int maxIndex = -1;
```

```
        for (int j = 0; j < m; j++) {
```

```
            if (blockSize[j] >= processSize[i]) {
```

```
                if (maxIndex == -1 || blockSize[j] > blockSize[maxIndex]) {
```

```
                    maxIndex = j;
```

```
                }
```

```
            }
```

```
}
```

```
if (maxIndex != -1) {
```

```
    allocation[i] = maxIndex;
```

```
    blockSize[maxIndex] -= processSize[i];
```

```
}
```

```
}
```

```
printf("Process No\tProcess Size\tBlock no\n");
```

```
for (int i = 0; i < n; i++) {
```

```
    if (allocation[i] != -1) {
```

```
        printf("%d\t%d\t%d\n", i + 1, processSize[i], allocation[i] + 1);
```

```
    } else {
```

```
        printf("%d\t%d\tNot Allocated\n", i + 1, processSize[i]);
```

```
    }
```

```
}
```

```
}
```

```
int main() {
```

```
    int m, n;
```

```
    printf("Enter number of memory blocks: ");
```

```
    scanf("%d", &m);
```

```
    int blockSize[m];
```

```
    printf("Enter the sizes of memory blocks:\n");
```

```
    for (int i = 0; i < m; i++) {
```

```
        scanf("%d", &blockSize[i]);
```

```
}

printf("Enter number of processes: ");

scanf("%d", &n);

int processSize[n];

printf("Enter the sizes of processes:\n");

for (int i = 0; i < n; i++) {

    scanf("%d", &processSize[i]);

}

worstFit(blockSize, m, processSize, n);

return 0;

}
```

Output:

Output

Enter number of memory blocks: 3

Enter the sizes of memory blocks:

6

4

9

Enter number of processes: 2

Enter the sizes of processes:

4

8

Process No	Process Size	Block no
------------	--------------	----------

1	4	3
---	---	---

2	8	Not Allocated
---	---	---------------

192324085|

=== Code Execution Successful ===