OPERATING SYSTEM LAB

Name: Thorat Amey Arun   Reg No.: 23MCS1004

LAB EXPERIMENT 9
Memory Allocation

Write a C/C++ program for dynamic memory allocation algorithm. Consider six memory partitions of size 200 KB, 400 KB, 600 KB, 500 KB, 300 KB and 250 KB. These partitions need to be allocated to four processes of sizes 357 KB, 210 KB, 468 KB and 491 KB in that order. Calculate the external fragmentation caused if any.
Perform the allocation of processes using-

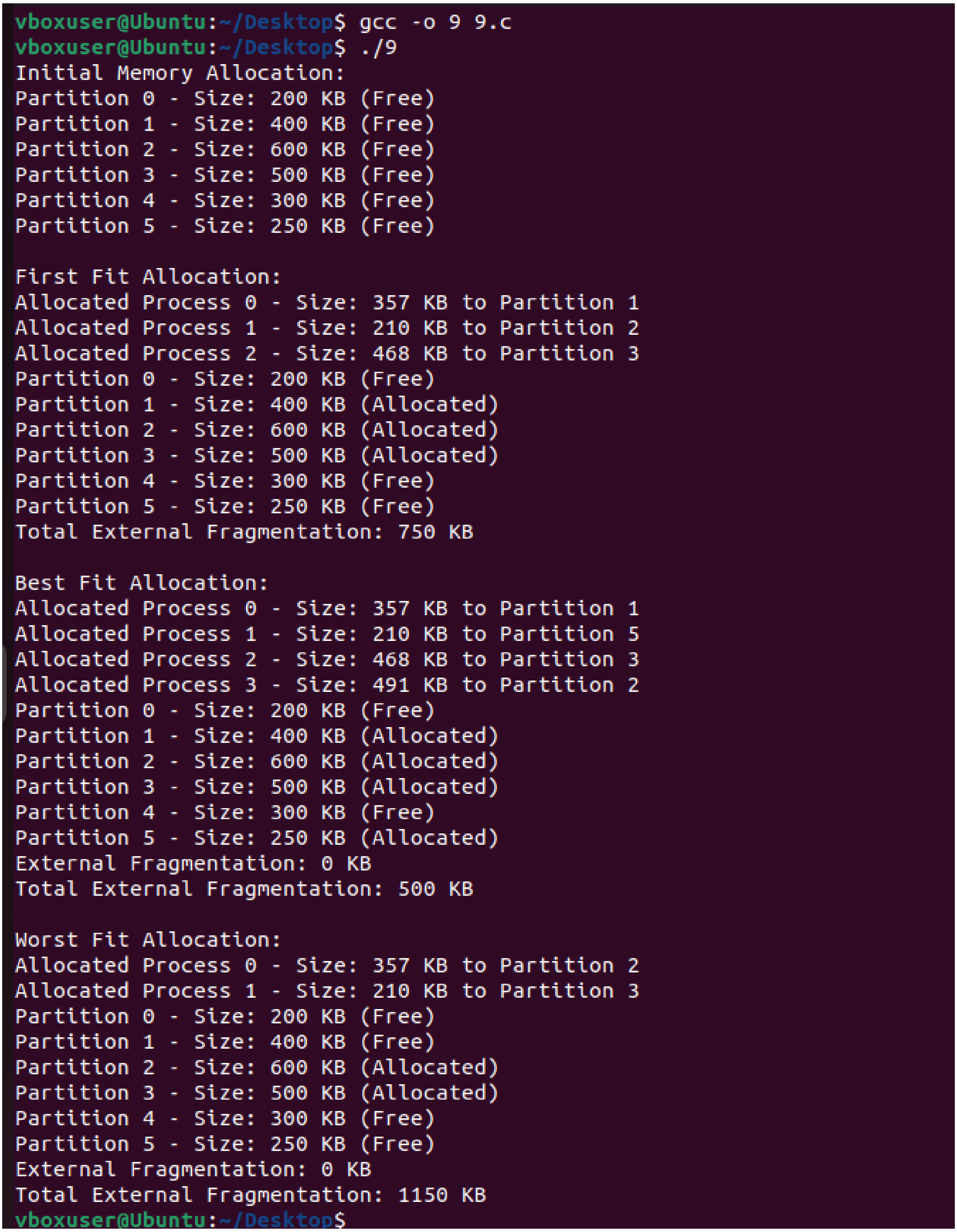
1. First Fit Algorithm
2. Best Fit Algorithm
3. Worst Fit Algorithm

Program:

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

struct MemoryPartition {
   int size;
   bool allocated;
};

struct Process {
   int size;
   bool allocated;
};

int printMemory(const struct MemoryPartition* memory, int n) {
   for (int i = 0; i < n; ++i) {
      printf("Partition %d - Size: %d KB", i, memory[i].size);
      if (memory[i].allocated) {
         printf(" (Allocated)\n");
      } else {
         printf(" (Free)\n");
      }
   }
}
```

```
int firstFit(struct MemoryPartition* memory, int n, struct Process* processes, int
m) {
    for (int i = 0; i < m; ++i) {
        for (int j = 0; j < n; ++j) {

            if (!memory[j].allocated && memory[j].size >= processes[i].size) {
                memory[j].allocated = true;
                processes[i].allocated = true;
                printf("Allocated Process %d - Size: %d KB to Partition %d\n", i,
processes[i].size, j);

                break;
          }
       }
    }
}

int bestFit(struct MemoryPartition* memory, int n, struct Process* processes, int
m) {
    for (int i = 0; i < m; ++i) {
        int bestFitIndex = -1;
        int bestFitSize = INT_MAX;

        for (int j = 0; j < n; ++j) {
            if (!memory[j].allocated && memory[j].size >= processes[i].size) {
                if (memory[j].size - processes[i].size < bestFitSize) {
                    bestFitSize = memory[j].size - processes[i].size;
                    bestFitIndex = j;
                }
            }
        }

        if (bestFitIndex != -1) {
            memory[bestFitIndex].allocated = true;
            processes[i].allocated = true;
            printf("Allocated Process %d - Size: %d KB to Partition %d\n", i,
processes[i].size, bestFitIndex);
        }
    }
}

int worstFit(struct MemoryPartition* memory, int n, struct Process* processes,
int m) {
    for (int i = 0; i < m; ++i) {
        int worstFitIndex = -1;
        int worstFitSize = -1;

        for (int j = 0; j < n; ++j) {
            if (!memory[j].allocated && memory[j].size >= processes[i].size) {
                if (memory[j].size - processes[i].size > worstFitSize) {
                    worstFitSize = memory[j].size - processes[i].size;
                    worstFitIndex = j;
                }
            }
        }

        if (worstFitIndex != -1) {
            memory[worstFitIndex].allocated = true;
            processes[i].allocated = true;
            printf("Allocated Process %d - Size: %d KB to Partition %d\n", i,
processes[i].size, worstFitIndex);
        }
    }
}

int main() {
    struct MemoryPartition memory[] = {
        {200, false},
        {400, false},
        {600, false},
        {500, false},
        {300, false},
        {250, false}
    };

    struct Process processes[] = {
        {357, false},
        {210, false},
        {468, false},
        {491, false}
    };
int k;
    int n = sizeof(memory) / sizeof(memory[0]);
    int m = sizeof(processes) / sizeof(processes[0]);
    printf("Initial Memory Allocation:\n");
    printMemory(memory, n);

    printf("\nFirst Fit Allocation:\n");
    firstFit(memory, n, processes, m);
    printMemory(memory, n);

    int externalFragmentation = 0;
    for (int i = 0; i < n; ++i) {
        if (!memory[i].allocated) {
            externalFragmentation += memory[i].size;
        }
    }

    printf("Total External Fragmentation: %d KB\n", externalFragmentation);

    for (int i = 0; i < n; ++i) {
        memory[i].allocated = false;
    }
    for (int i = 0; i < m; ++i) {
        processes[i].allocated = false;
    }

    printf("\nBest Fit Allocation:\n");
    bestFit(memory, n, processes, m);
    printMemory(memory, n);

    printf("External Fragmentation: %d KB\n", k);

    externalFragmentation = 0;
    for (int i = 0; i < n; ++i) {
        if (!memory[i].allocated) {
            externalFragmentation += memory[i].size;
        }
    }
    printf("Total External Fragmentation: %d KB\n", externalFragmentation);

    for (int i = 0; i < n; ++i) {
        memory[i].allocated = false;
    }
    for (int i = 0; i < m; ++i) {
        processes[i].allocated = false;
    }

    printf("\nWorst Fit Allocation:\n");
    worstFit(memory, n, processes, m);
    printMemory(memory, n);
    printf("External Fragmentation: %d KB\n", k);

    externalFragmentation = 0;
    for (int i = 0; i < n; ++i) {
        if (!memory[i].allocated) {
            externalFragmentation += memory[i].size;
        }
    }

    printf("Total External Fragmentation: %d KB\n", externalFragmentation);

    return 0;
}
```

Output:



```
vboxuser@Ubuntu:~/Desktop$ gcc -o 9 9.c
vboxuser@Ubuntu:~/Desktop$ ./9
Initial Memory Allocation:
Partition 0 - Size: 200 KB (Free)
Partition 1 - Size: 400 KB (Free)
Partition 2 - Size: 600 KB (Free)
Partition 3 - Size: 500 KB (Free)
Partition 4 - Size: 300 KB (Free)
Partition 5 - Size: 250 KB (Free)

First Fit Allocation:
Allocated Process 0 - Size: 357 KB to Partition 1
Allocated Process 1 - Size: 210 KB to Partition 2
Allocated Process 2 - Size: 468 KB to Partition 3
Partition 0 - Size: 200 KB (Free)
Partition 1 - Size: 400 KB (Allocated)
Partition 2 - Size: 600 KB (Allocated)
Partition 3 - Size: 500 KB (Allocated)
Partition 4 - Size: 300 KB (Free)
Partition 5 - Size: 250 KB (Free)
Total External Fragmentation: 750 KB

Best Fit Allocation:
Allocated Process 0 - Size: 357 KB to Partition 1
Allocated Process 1 - Size: 210 KB to Partition 5
Allocated Process 2 - Size: 468 KB to Partition 3
Allocated Process 3 - Size: 491 KB to Partition 2
Partition 0 - Size: 200 KB (Free)
Partition 1 - Size: 400 KB (Allocated)
Partition 2 - Size: 600 KB (Allocated)
Partition 3 - Size: 500 KB (Allocated)
Partition 4 - Size: 300 KB (Free)
Partition 5 - Size: 250 KB (Allocated)
External Fragmentation: 0 KB
Total External Fragmentation: 500 KB

Worst Fit Allocation:
Allocated Process 0 - Size: 357 KB to Partition 2
Allocated Process 1 - Size: 210 KB to Partition 3
Partition 0 - Size: 200 KB (Free)
Partition 1 - Size: 400 KB (Free)
Partition 2 - Size: 600 KB (Allocated)
Partition 3 - Size: 500 KB (Allocated)
Partition 4 - Size: 300 KB (Free)
Partition 5 - Size: 250 KB (Free)
External Fragmentation: 0 KB
Total External Fragmentation: 1150 KB
vboxuser@Ubuntu:~/Desktop$
```