**NAME: Saad Ashraf**

**ROLL NO: 21F9167**

**SECTION: 4B**

**TASK 1:**

1. Given six memory partitions of 300 KB, 600 KB, 350 KB, 200 KB, 750 KB,

and 125 KB (in order), how would the first-fit, best-fit, and worst-fit

algorithms place processes of size 115 KB, 500 KB, 358 KB, 200 KB, and

375 KB (in order)?

**SOLUTION:**

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

struct Block {

int size;

bool available;

};

void allocateFirstFit(vector<Block>& memory, int processSize) {

for (int i = 0; i < memory.size(); i++) {

if (memory[i].available && memory[i].size >= processSize) {

memory[i].available = false;

cout << "FIRST FIT : Allocated " << processSize << " KB to BLOCK: " << i << endl;

return;

}

}

cout << "FIRST FIT: space is not available for process of size " << processSize << " KB" << endl;

}

void allocateBestFit(vector<Block>& memory, int processSize) {

vector<int> availableBlocks;

for (int i = 0; i < memory.size(); i++) {

if (memory[i].available && memory[i].size >= processSize) {

availableBlocks.push\_back(i);

}

}

if (availableBlocks.empty()) {

cout << "BEST FIT: space is not available for process of size " << processSize << " KB" << endl;

return;

}

sort(availableBlocks.begin(), availableBlocks.end(), [&](int a, int b){

return memory[a].size < memory[b].size;

});

memory[availableBlocks[0]].available = false;

cout << "BEST FIT: Allocated " << processSize << " KB to BLOCK: " << availableBlocks[0] << endl;

}

void allocateWorstFit(vector<Block>& memory, int processSize) {

vector<int> availableBlocks;

for (int i = 0; i < memory.size(); i++) {

if (memory[i].available && memory[i].size >= processSize) {

availableBlocks.push\_back(i);

}

}

if (availableBlocks.empty()) {

cout << "WORST FIT: space is not available for process of size " << processSize << " KB" << endl;

return;

}

sort(availableBlocks.begin(), availableBlocks.end(), [&](int a, int b){

return memory[a].size > memory[b].size;

});

memory[availableBlocks[0]].available = false;

cout << "WORST FIT: Allocated " << processSize << " KB to BLOCK: " << availableBlocks[0] << endl;

}

int main() {

vector<Block> memory = {{300, true}, {600, true}, {350, true}, {200, true}, {750, true}, {125, true}};

allocateFirstFit(memory, 115);

allocateFirstFit(memory, 500);

allocateFirstFit(memory, 358);

allocateFirstFit(memory, 200);

allocateFirstFit(memory, 375);

allocateBestFit(memory, 115);

allocateBestFit(memory, 500);

allocateBestFit(memory, 358);

allocateBestFit(memory, 200);

allocateBestFit(memory, 375);

allocateWorstFit(memory, 115);

allocateWorstFit(memory, 500);

allocateWorstFit(memory, 358);

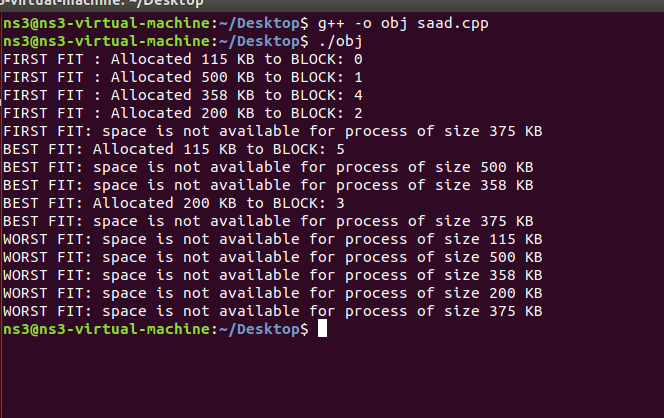
allocateWorstFit(memory, 200);

allocateWorstFit(memory, 375);

return 0;

}

**OUTPUT:**

****