# First Fit

#include <iostream>

using namespace std;

int main() {

int memBlocks[] = {300, 600, 350, 200, 750, 125};

int numBlocks = 6;

int processes[] = {115, 500, 358, 200, 375};

int numProcesses = 5;

int allocation[numProcesses];

int i = 0;

while (i < numProcesses) {

allocation[i] = -1;

int j = 0;

while (j < numBlocks) {

if (memBlocks[j] >= processes[i]) {

allocation[i] = j;

memBlocks[j] -= processes[i];

break;

}

j++;

}

i++;

}

cout << "Process No.\tProcess Size\tPartition No.\n";

i = 0;

while (i < numProcesses) {

cout << i+1 << "\t\t" << processes[i] << "\t\t";

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

cout << allocation[i]+1;

} else {

cout << "Not Allocated";

}

cout << endl;

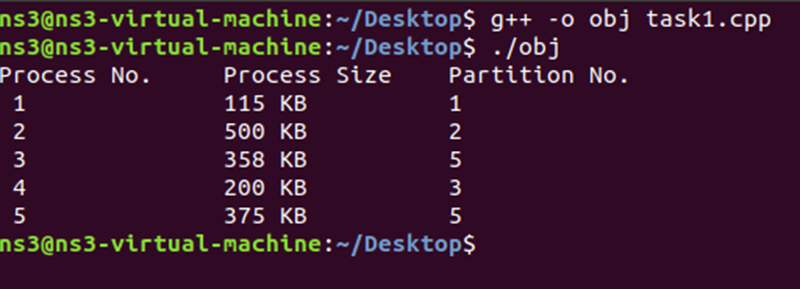
i++;

}

return 0;

}

out put



BEST FIT  
  
#include <iostream>

using namespace std;

int main() {

int memory\_blocks[] = {300, 600, 350, 200, 750, 125};

int num\_blocks = 6;

int processes[] = {115, 500, 358, 200, 375};

int num\_processes = 5;

int allocation[num\_processes];

int i = 0;

while (i < num\_processes) {

allocation[i] = -1;

int best\_index = -1;

int j = 0;

while (j < num\_blocks) {

if (memory\_blocks[j] >= processes[i]) {

if (best\_index == -1) {

best\_index = j;

} else {

if (memory\_blocks[j] < memory\_blocks[best\_index]) {

best\_index = j;

}

}

}

j++;

}

if (best\_index != -1) {

allocation[i] = best\_index;

memory\_blocks[best\_index] -= processes[i];

}

i++;

}

cout << "Process No.\tProcess Size\tPartition No.\n";

i = 0;

while (i < num\_processes) {

cout << i+1 << "\t\t" << processes[i] << "\t\t";

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

cout << allocation[i]+1;

} else {

cout << "Not Allocated";

}

cout << endl;

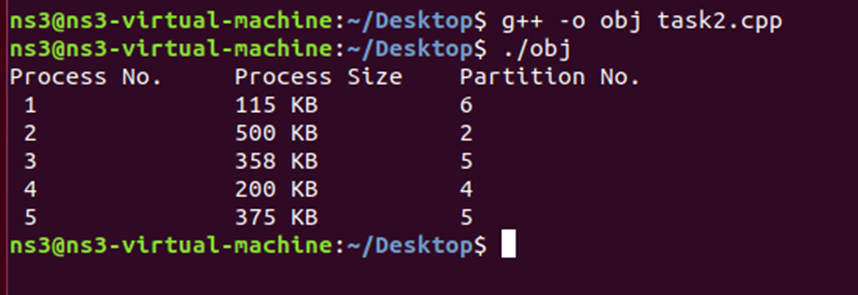
i++;

}

return 0;

}

# OUT PUT



# WORST FIT

#include <iostream>

using namespace std;

int main() {

int memory\_blocks[] = {300, 600, 350, 200, 750, 125};

int num\_blocks = 6;

int processes[] = {115, 500, 358, 200, 375};

int num\_processes = 5;

int allocation[num\_processes];

int i = 0;

while (i < num\_processes) {

allocation[i] = -1;

int worst\_index = -1;

int j = 0;

while (j < num\_blocks) {

if (memory\_blocks[j] >= processes[i]) {

if (worst\_index == -1 || memory\_blocks[j] > memory\_blocks[worst\_index])

worst\_index = j;

}

j++;

}

if (worst\_index != -1) {

allocation[i] = worst\_index;

memory\_blocks[worst\_index] -= processes[i];

}

i++;

}

cout << "Process No.\tProcess Size\tPartition No.\n";

int k = 0;

while (k < num\_processes) {

cout << k+1 << "\t\t" << processes[k] << "\t\t";

if (allocation[k] != -1)

cout << allocation[k]+1;

else

cout << "Not Allocated";

cout << endl;

k++;

}

return 0;

}

# OUTPUT

