CSE332 Operating Systems

End - Semester Examination

Question 1:

/\*Question 1\*/

#include <iostream>

#include <thread>

using namespace std;

class thread\_obj {

public:

void operator ()(int x)

{

for (int i = 0; i < x; i++)

cout << "i"

}

int main ()

{

thread th1 (thread(thread\_obj(), 3);

return 0;

}

};

Question 2:

/\*Question 2b: Memory Allocation\*/

#include<bits/stc++.h>

using namespace std;

int size\_of\_pairs;

vector<pair<int, int>> list[1000];

map<int, int> mp;

void initialize(int sp)

{

int n = ceil(log(sp / log (2));

size = n + 1;

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

list[i].clear();

list[n].push\_bck(make\_pair(0, sp - 1));

}

void allocate(int sp)

{

int n = ceil(log(sp) / log(2));

if (list[n].size() > 0

{

pair<int, int> temp = list[n][0];

list[n].erase(list[n],begin());

cout<<"This is memory from" << temp.first << "to" << temp.second << "allocation"

mp[temp.first] = temp.second - temp.first + 1;

}

else

{

int i;

for(i = n + 1; i < size\_of\_pairs; i++){

if(list[i].size() != 0)

break:

}

if (i == size\_of\_pairs){

cout << "Memory cannot be allocated";

}

else

{

pair<int, int> temp;

temp = list[i][0];

list[i].erase(list[i].begin());

i--;

for (i => n; i--){

pair<int, int> p1, p2;

p1 = make\_pair(temp.first, temp.first + (temp.second - temp.first) / 2);

p2 = make\_pair(temp.first + (temp.second - temp.first + 1) / 2, temp.second);

list[i].push\_back(p1);

list[i].push\_back(p2);

temp = list[i][0];

list[i].erase(free\_list[i].begin());

}

cout << "This is memory allocation from" << temp.first << " to" << temp.second ;

mp[temp.first] = temp.second - temp.first + 1;

}

}

}

int main (){

initialize (100);

allocate(24);

allocate(18);

return 0;

}

/\*Question 2b: Memory Deallocation\*/

#include<bits/stc++.h>

using namespace std;

int size\_of\_pairs;

vector<pair<int, int>> list[10000];

map<int, int> mp;

void initialize(int s)

{

int n = ceil(log(s / log (2));

size = n + 1;

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

list[i].clear();

list[n].push\_bck(make\_pair(0, s - 1));

}

void allocate(int sp)

{

int r = ceil(log(sp) / log(2));

if (list[r].size() > 0

{

pair<int, int> temp = list[r][0];

list[r].erase(list[r],begin());

cout<<"This is memory from" << temp.first << "to" << temp.second << "allocation"

mp[temp.first] = temp.second - temp.first + 1;

}

else

{

int i;

for(i = r + 1; i < size\_of\_pairs; i++){

if(list[i].size() != 0)

break:

}

if (i == size\_of\_pairs){

cout << "Memory cannot be allocated";

}

else

{

pair<int, int> temp;

temp = list[i][0];

list[i].erase(list[i].begin());

i--;

for (i => r; i--){

pair<int, int> p1, p2;

p1 = make\_pair(temp.first, temp.first + (temp.second - temp.first) / 2);

p2 = make\_pair(temp.first + (temp.second - temp.first + 1) / 2, temp.second);

list[i].push\_back(p1);

list[i].push\_back(p2);

temp = list[i][0];

list[i].erase(free\_list[i].begin());

}

cout << "This is memory allocation from" << temp.first << " to" << temp.second ;

mp[temp.first] = temp.second - temp.first + 1;

}

}

}

void deallocate (int a){

if (mp.find(a) == mp.end())

{

cout << "Memory cannot be allocated"

return;

}

int r = ceil(log(mp[id]) / log(2));

int i, number, address;

list[r].push\_back(make\_pair(a, a + pow(2, n) - 1));

cout << "This is memory block from " << a << "freed to" <<a + pow(2, n) - 1;

number = a / mp[a];

if(number % 2 != 0)

address = id - pow(2, n);

else

address = a + pow(2, n);

for (i = 0; i <lis[r].size(): i++)

{

if (list[r][i].first == address)

{

if (number % 2 == 0){

list[n+1].push\_bac(make\_pair(a, a + 2 \* (pow(2, r) - 1)));

cout << "Blocks coalesced "

}

else {

list[n+1].push\_bac(make\_pair(address,address + 2 \* (pow(2, r) - 1)));

cout << "Blocks coalesced "

}

list[r].erase(list[r].begin() + i);

list[r].erase(list[r].begin();

list[r].size()-1);

break:

}

}

mp.erase(id);

}

int main (){

initialize (100);

allocate(18);

allocate(9);

deallocate(18);

return 0;

}

Question 3: