FIFO

#include <iostream>

#include <vector>

#include <algorithm>

using namespace std;

int main() {

int m, n, i1 = 0, k = 0;

cout << "FIFO page replacement algorithm..\n";

cout << "Enter the number of frames: ";

cin >> n;

cout << "Enter the reference string (terminate with 0):\n";

vector<int> rs;

while (true) {

int value;

cin >> value;

if (value == 0)

break;

rs.push\_back(value);

}

m = rs.size();

vector<int> p(n, 0);

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

if (find(p.begin(), p.end(), rs[i]) != p.end()) {

cout << "Data already in page...\n";

} else {

p[i1] = rs[i];

i1 = (i1 + 1) % n;

k++;

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

cout << "\nPage" << j + 1 << ":" << p[j];

if (p[j] == rs[i])

cout << "\*";

}

cout << "\n\n";

}

}

cout << "Total number of page faults = " << k << endl;

return 0;

}

MEMORY MANAGEMENT

#include<iostream>

using namespace std;

int main() {

int ms, bs, nob, ef, n, mp[10], tif = 0, i, p = 0;

cout << "Enter the total memory available in Bytes: ";

cin >> ms;

cout << "Enter the block size in bytes: ";

cin >> bs;

nob = ms / bs;

ef = ms - nob \* bs;

cout << "Enter the number of processes: ";

cin >> n;

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

cout << "Enter memory required for process " << i + 1 << ": ";

cin >> mp[i];

}

cout << "\nNo. of Blocks available in memory: " << nob << endl;

cout << "PROCESS\t MEMORYREQUIRED\tALLOCATED\tINTERNALFRAGMENTATION" << endl;

for (i = 0; i < n && p < nob; i++) {

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

if (mp[i] > bs)

cout << "\t\tNO\t\t--";

else {

cout << "\t\tYES\t" << bs - mp[i];

tif += bs - mp[i];

p++;

}

cout << endl;

}

if (i < n)

cout << "\nMemory is full, remaining processes cannot be accommodated" << endl;

cout << "Total Internal Fragmentation is " << tif << endl;

cout << "Total External Fragmentation is " << ef << endl;

return 0;

}

VARIABLE SIZED PARTITION

#include <iostream>

using namespace std;

int main() {

int ms, mp[10], i, temp, n = 0;

char ch = 'y';

cout << "\nEnter the total memory available (in Bytes)-- ";

cin >> ms;

temp = ms;

for (i = 0; ch == 'y'; i++, n++) {

cout << "\nEnter memory required for process " << i + 1 << " (in Bytes) -- ";

cin >> mp[i];

if (mp[i] <= temp) {

cout << "\nMemory is allocated for Process " << i + 1;

temp = temp - mp[i];

} else {

cout << "\nMemory is Full";

break;

}

cout << "\nDo you want to continue(y/n) -- ";

cin >> ch;

}

cout << "\n\nTotal Memory Available -- " << ms;

cout << "\n\n\tPROCESS\t\t MEMORY ALLOCATED ";

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

cout << "\n \t" << i + 1 << "\t\t" << mp[i];

cout << "\n\nTotal Memory Allocated is " << ms - temp;

cout << "\nTotal External Fragmentation is " << temp;

return 0;

}

FIXED SIZED PARTITION

#include <iostream>

using namespace std;

int main() {

int ms, bs, nob, ef, n, mp[10], tif = 0;

int i, p = 0;

cout << "Enter the total memory available (in Bytes) -- ";

cin >> ms;

cout << "Enter the block size (in Bytes) -- ";

cin >> bs;

nob = ms / bs;

ef = ms - nob \* bs;

cout << "\nEnter the number of processes -- ";

cin >> n;

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

cout << "Enter memory required for process " << i + 1 << " (in Bytes)-- ";

cin >> mp[i];

}

cout << "\nNo. of Blocks available in memory--" << nob;

cout << "\n\nPROCESS\tMEMORYREQUIRED\tALLOCATED\tINTERNAL FRAGMENTATION";

for (i = 0; i < n && p < nob; i++) {

cout << "\n " << i + 1 << "\t\t" << mp[i];

if (mp[i] > bs)

cout << "\t\tNO\t\t---";

else {

cout << "\t\tYES\t" << bs - mp[i];

tif = tif + bs - mp[i];

p++;

}

}

if (i < n)

cout << "\nMemory is Full, Remaining Processes cannot be accommodated";

cout << "\n\nTotal Internal Fragmentation is " << tif;

cout << "\nTotal External Fragmentation is " << ef;

return 0;

}

LEAST RECENTLY USED

#include <iostream>

using namespace std;

int main() {

int q[20], p[50], c = 0, c1, d, f, i, j, k = 0, n, r, t, b[20], c2[20];

cout << "Enter no of pages: ";

cin >> n;

cout << "Enter the reference string: ";

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

cin >> p[i];

cout << "Enter no of frames: ";

cin >> f;

q[k] = p[k];

cout << "\n\t" << q[k] << endl;

c++;

k++;

for (i = 1; i < n; i++) {

c1 = 0;

for (j = 0; j < f; j++) {

if (p[i] != q[j])

c1++;

}

if (c1 == f) {

c++;

if (k < f) {

q[k] = p[i];

k++;

for (j = 0; j < k; j++)

cout << "\t" << q[j];

cout << endl;

} else {

for (r = 0; r < f; r++) {

c2[r] = 0;

for (j = i - 1; j >= 0; j--) {

if (q[r] != p[j])

c2[r]++;

else

break;

}

}

for (r = 0; r < f; r++)

b[r] = c2[r];

for (r = 0; r < f; r++) {

for (j = r; j < f; j++) {

if (b[r] < b[j]) {

t = b[r];

b[r] = b[j];

b[j] = t;

}

}

}

for (r = 0; r < f; r++) {

if (c2[r] == b[0])

q[r] = p[i];

cout << "\t" << q[r];

}

cout << endl;

}

}

}

cout << "\nThe no of page faults is " << c << endl;

return 0;

}