2021270660 이지원 - 운영체제 실습 과제 5

Matrix Multiplication

Table of contents

System Description : read_matrix

```
#ifndef READ_MATRIX_H
#define READ_MATRIX_H

#include <vector>
#include <string>

#define MAX_ROW 100
#define MAX_COL 100

void read_matrix(std::vector<std::vector<int>>& A, int& Ar, int& Ac, const char*
    filename);

#endif // READ_MATRIX_H

```

➔ 헤더파일 분리 및 vector 사용으로 전환

System Description : read_matrix

```cpp
#define MAX_LINE 1000
#define DELIM " \r\t"  // 공백, 탭 등을 구분자로 설정

void read_matrix(std::vector<std::vector<int>>& A, int& Ar, int& Ac, const char* filename) {
    char line[MAX_LINE], * tok;
    FILE* fp;

    // 파일 열기
    if (!(fp = fopen(filename, "r"))) {
        printf("ERROR: file open\n");
        exit(0);
    }

    Ar = 0;
    A.clear();  // 기존의 데이터 비우기

    // 파일에서 한 줄씩 읽어오기
    while (fgets(line, MAX_LINE, fp)) {
        tok = strtok(line, DELIM);  // 첫 번째 토큰을 구분자 기준으로 잘라서 가져오기
        std::vector<int> row;  // 각 행을 저장할 벡터
        Ac = 0;

        // 행에 있는 모든 숫자를 읽어서 벡터에 저장
        do {
            row.push_back(atoi(tok));  // 문자열을 정수로 변환하여 벡터에 추가
            Ac++;
        } while (tok = strtok(NULL, DELIM));  // 다음 토큰으로 이동

        A.push_back(row);  // 현재 행을 2D 벡터에 추가
        Ar++;  // 행 개수 증가
    }

    fclose(fp);  // 파일 닫기
}
```

➔ 파일 읽어오는 로직은 그대로 사용하되 vector로 저장하고 col, row 수는 따로 누적하여 계산했음. Vector메서드 써도 되지만 함수 오버헤드가 어떨지 모르겠어서 그냥 쉬운 방법 사용했음

System Description : mm

```cpp
// 행렬 곱셈 함수 (멀티스레딩)
vector<vector<int>> matrix_multiply(const vector<vector<int>>& A, const vector<vector<int>>& B, int m, int k, int n) {
    vector<vector<int>> C(m, vector<int>(n, 0));
    vector<thread> threads;

    // 각 C[i][j] 항목을 계산하는 스레드
    auto compute_entry = [&](int i, int j) {
        this_thread::sleep_for(chrono::seconds(1));  // 1초 대기
        for (int l = 0; l < k; ++l) {
            C[i][j] += A[i][l] * B[l][j];
        }
        };

    // 스레드 생성 (각 C[i][j] 항목 계산)
    for (int i = 0; i < m; ++i) {
        for (int j = 0; j < n; ++j) {
            threads.push_back(thread(compute_entry, i, j));  // 각 스레드 생성
        }
    }

    // 모든 스레드 종료 대기
    for (auto& th : threads) {
        th.join();  // 스레드 종료 대기
    }

    return C;
}
```

System Description : mm

```cpp
// 행렬 출력 함수
void print_matrix(const vector<vector<int>>& matrix) {
    for (const auto& row : matrix) {
        for (int val : row) {
            cout << setw(4) << val << " ";
        }
        cout << endl;
    }
}
```

System Description : mm

```
int main(int argc, char** argv) {
    if (argc < 3) {
        cout << "ERROR: input sequence! Usage: mm <file1> <file2> ...\n";
        return 0;
    }

    vector<vector<int>> A, B, C;
    int m, k, n;

    // 시간 측정 시작
    auto start_time = chrono::high_resolution_clock::now();

    // 첫 번째 행렬 파일 읽기
    read_matrix(A, m, k, argv[1]);
```

➔ 기존 read_matrix의 main 함수 부분을 적절히 수정하여 사용

System Description : mm

```
// 각 파일을 읽어와 행렬 곱셈을 수행
for (int arg = 1; arg < argc - 1; ++arg) {
    if (arg > 1) {
        A = C;
    }

    // 다음 행렬 파일 읽기
    read_matrix(B, k, n, argv[arg + 1]);

    cout << "(" << setw(5) << m << "x" << setw(5) << k << ") X ("
        << setw(5) << k << "x" << setw(5) << n << ") = ("
        << setw(5) << m << "x" << setw(5) << n << ")" << endl;

    // 행렬 곱셈 수행
    C = matrix_multiply(A, B, m, k, n);
}
```

➔ 첫번째 행렬 이후로는 앞선 계산 결과를 이용하여 연산

System Description : mm

```cpp
// 시간 측정 끝
auto end_time = chrono::high_resolution_clock::now();

// 결과 행렬 출력
cout << "Result Matrix:" << endl;
print_matrix(C);

// 결과를 파일에 저장
ofstream result_file("result.txt", ios::app);
if (result_file.is_open()) {
    result_file << "Result Matrix (" << m << "x" << n << "):\n";
    for (const auto& row : C) {
        for (int val : row) {
            result_file << setw(4) << val << " ";
        }
        result_file << endl;
    }
    result_file << "\n";
    result_file.close();
}
else {
    cout << "Unable to open file for writing result.\n";
}

// 처리 시간 계산
chrono::duration<double> duration = end_time - start_time;
cout << "Processing Time: " << fixed << setprecision(3) << duration.count() << " sec\n\n";

return 0;
```

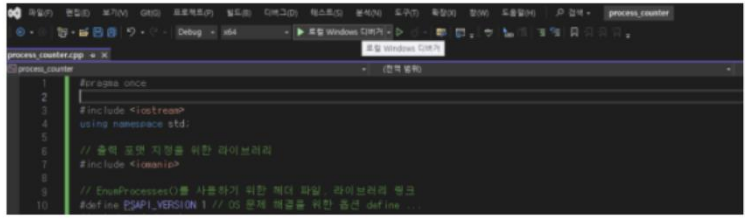
➔ 시간 계산 및 결과 출력

Test description

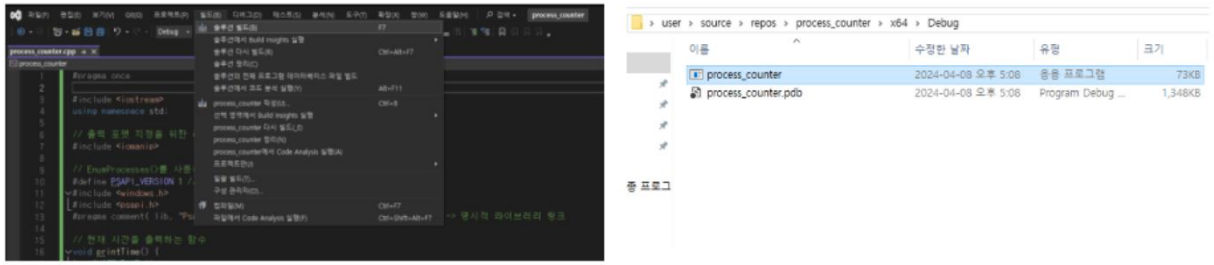
```
C:₩Users₩USER₩source₩repos₩이지원-2021270660-assignment-05₩이지원-2021270660-assignment-05₩x64₩Debug>mm 3x4 4x5 5x100 100
x100
(    3x    4) X (    4x    5) = (    3x    5)
(    3x    5) X (    5x  100) = (    3x  100)
(    3x  100) X (  100x  100) = (    3x  100)
Result Matrix:
-425 -621  240  144  765 -732 -156  -30  561  152  221  -87  963 -1053 -421 -148  456  175    8 -109  114 -182 -357   92
-429  245  453 -734 -746  213  -80  375 -308 1060  224   99  142   74  220  405  155  103  143   90  126  538 -304 -575
531 -639 -312 -221  383  -43 1174 -291 -1099  242  245  -14  219  -26  310 -372 -366  -31 -428  193  790  -61  208  182
836  464  694 -348   61   72  619 -598 -581 -528 -651 -129 -771  151  101 -417   90  107  768  217  178 -1451 1198  165
834 -602  418  320
1932  142  569 -1489 -360  -78 1167 -532 1265 1191  211 -269 -1595  127  200 -181  155  973 -590  423 1197 -222 -281 1007
 1260  303  908  776  208 1010  118  465  949 -1078 -702 -770  116 -119  189  618 -170  475 1762  740  302 1145 -954 -162
 -1596  391  111    3 1008 -809  393   69   20  113  596 -1224 -1329 -829 -394  380 -669 1819 -772 -360 -616 -238 -748 -1
047  197  519 -494 -1147 -705 -206 -291 1088  253 -464  990  402  581 -238  164  359  304  575   92   19  -94 -119 -475
257 1054 -914 -102  527
 239  875 -125 -355 -517  388 1063   -4 -432 -1307 -1032 -510  872  214   -7 1347 -673 -932  160 -284 -207  218 1462  449
  969 -1890 -1379 1456  858 -475 -738 -168  847 -426 -362 1055  872   83 -347  707 -699 1354 -541   24 -130 -451  448  95
9 -199 1438  137 -716 -841 -216 -213  -74  855 -139 -919 -630   50  881  542    6  615  324  728 -921 -436  241 -200 -785
 -1475 -509  310  -41 -116  540 -1486  634  180 1102  -75  -51  840 -1257 -939  488 -554  198 -948  358  222 1090 -1597 -
452 -1800 1142 -142  775
Processing Time: 3.082 sec
```

User documentation

1. .exe 파일 실행 (빌드된 상태)
2. 로컬 디버거로 실행 : 프로젝트 생성 후 소스 파일에 추가 후 로컬 디버거로 실행



3. 빌드 후 .exe 파일 실행 : 마찬가지로 프로젝트 생성 후 빌드 - 솔루션 빌드 후 .exe 파일을 찾아 실행



exe 파일 실행 시 더블클릭 대신에
cmd 창에서 파일경로/파일명 행렬파일명 행렬파일명... 를 입력하여 실행합니다.

ex) C:₩...프로젝트 경로₩mm 3x4 4x5 5x100

Self-evaluation

90/100

모듈화 적절히 사용하였음 (헤더파일이용)
주석 및 가독성 신경써서 작성
라이브러리 및 주어진 코드 적절히 사용

그러나 출력 형태가 약간 마음에 들지 않음 ...