```c
#include <stdio.h>
#include <pthread.h>
#include <time.h>

#define SIZE 3

typedef struct {
    int (*A)[SIZE];
    int (*B)[SIZE];
    int (*C)[SIZE];
    int row;
} ThreadData;

void* multiply_row(void* arg) {
    ThreadData* data = (ThreadData*) arg;
    int row = data->row;
    for (int j = 0; j < SIZE; j++) {
        data->C[row][j] = 0;
        for (int k = 0; k < SIZE; k++) {
            data->C[row][j] += data->A[row][k] * data->B[k][j];
        }
    }
    return NULL;
}

int main() {
    int A[SIZE][SIZE] = {{1, 2, 3}, {4, 5, 6}, {7, 8, 9}};
    int B[SIZE][SIZE] = {{9, 8, 7}, {6, 5, 4}, {3, 2, 1}};
    int C[SIZE][SIZE];
    pthread_t threads[SIZE];
    ThreadData thread_data[SIZE];

    clock_t start = clock();


    for (int i = 0; i < SIZE; i++) {
        thread_data[i].A = A;
        thread_data[i].B = B;
        thread_data[i].C = C;
        thread_data[i].row = i;
        pthread_create(&threads[i], NULL, multiply_row, (void*) &thread_data[i]);
    }

    for (int i = 0; i < SIZE; i++) {
        pthread_join(threads[i], NULL);
    }

    clock_t end = clock();
    double time_spent = (double)(end - start) / CLOCKS_PER_SEC;

    printf("Matriz Resultante C:\n");
    for (int i = 0; i < SIZE; i++) {
        for (int j = 0; j < SIZE; j++) {
            printf("%d ", C[i][j]);
        }
        printf("\n");
    }

    printf("Tempo de execução (multithreaded): %f segundos\n", time_spent);

    return 0;
}
```

Comece a programar ou gere código com IA.