Rodger Lee

CS33

Prof. Reinman

HW4

**6.45** ◆◆◆◆  
In this assignment, you will apply the concepts you learned in Chapters 5 and 6 to the problem of optimizing code for a memory-intensive application. Consider a procedure to copy and transpose the elements of an N × N matrix of type int. That is, for source matrix S and destination matrix D, we want to copy each element si,j to dj,i. This code can be written with a simple loop, where the arguments to the procedure are pointers to the destination (dst) and source (src) matrices, as well as the matrix size N (dim). Your job is to devise a transpose routine that runs as fast as possible.

void transpose(int \*dst, int \*src, int dim) {

int i, j, row, col;

int blocksize = 8;

for (i = 0; i < dim-8; i += blocksize) {

for (j = 0; j < dim-8; j += blocksize) {

for (row = i; row < i + blocksize && row < dim; row++) {

int rowdim = row\*dim;

for (col = j; col < j + blocksize && col < dim; col++)

dst[col\*dim + row] = src[rowdim + col];

}

}

}

}

while (i < dim) {

while (j < dim) {

dst[j\*dim + i] = src[i\*dim + j];

j++;

}

i++;

}

}

- break up 2D array into multiple smaller blocks

- change multiplication operations into addition and placed them outside of the loop