//#include "mini\_blas.h"

#ifdef \_\_cplusplus

#define PUT\_IN\_REGISTER

#else

#define PUT\_IN\_REGISTER register

#endif

void cpu\_gemm\_nn(int TA, int TB, int M, int N, int K, float ALPHA,

float \*A, int lda,

float \*B, int ldb,

float BETA,

float \*C, int ldc)

{

int i,j,k;

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

for(k = 0; k < K; ++k){

PUT\_IN\_REGISTER float A\_PART = ALPHA \* A[i \* lda + k];

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

C[i\*ldc+j] += A\_PART\*B[k\*ldb+j];

}

}

}

}

void cpu\_gemm\_nt(int TA, int TB, int M, int N, int K, float ALPHA,

float \*A, int lda,

float \*B, int ldb,

float BETA,

float \*C, int ldc)

{

int i,j,k;

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

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

PUT\_IN\_REGISTER float sum = 0;

for(k = 0; k < K; ++k){

sum += ALPHA\*A[i\*lda+k]\*B[k+j\*ldb];

}

C[i\*ldc+j] += sum;

}

}

}

void cpu\_gemm\_tn(int TA, int TB, int M, int N, int K, float ALPHA,

float \*A, int lda,

float \*B, int ldb,

float BETA,

float \*C, int ldc)

{

int i,j,k;

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

for(k = 0; k < K; ++k){

PUT\_IN\_REGISTER float A\_PART = ALPHA \* A[k \* lda + i];

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

C[i\*ldc+j] += A\_PART\*B[k\*ldb+j];

}

}

}

}

void cpu\_gemm\_tt(int TA, int TB, int M, int N, int K, float ALPHA,

float \*A, int lda,

float \*B, int ldb,

float BETA,

float \*C, int ldc)

{

int i,j,k;

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

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

for(k = 0; k < K; ++k){

C[i\*ldc+j] += ALPHA\*A[i+k\*lda]\*B[k+j\*ldb];

}

}

}

}

void cpu\_gemm(int TA, int TB, int M, int N, int K, float ALPHA,

float \*A, int lda,

float \*B, int ldb,

float BETA,

float \*C, int ldc)

{

int i, j;

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

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

C[i\*ldc + j] \*= BETA;

}

}

if(!TA && !TB)

cpu\_gemm\_nn( TA, TB, M, N, K, ALPHA,A,lda, B, ldb,BETA,C,ldc);

else if(TA && !TB)

cpu\_gemm\_tn( TA, TB, M, N, K, ALPHA,A,lda, B, ldb,BETA,C,ldc);

else if(!TA && TB)

cpu\_gemm\_nt( TA, TB, M, N, K, ALPHA,A,lda, B, ldb,BETA,C,ldc);

else

cpu\_gemm\_tt( TA, TB, M, N, K, ALPHA,A,lda, B, ldb,BETA,C,ldc);

}