| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [Common/include](http://docs.google.com/index.html) - MemMgr.h (source / [functions](http://docs.google.com/MemMgr.h.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 41 | 49 | 83.7 % | | Date: | 2012-03-23 |  | Functions: | 1 | 1 | 100.0 % | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 29 | 58 | 50.0 % | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : #ifndef NFT\_MEMMGR\_H  2 : : #define NFT\_MEMMGR\_H  3 : :   4 : : #include <cstdlib>  5 : : #include <boost/thread/mutex.hpp>  6 : : #include "ErrorHandler.h"  7 : : #include "Logger.h"  8 : : namespace nft {  9 : :   10 : : namespace \_impl {  11 : : template <typename \_Tp, ssize\_t cQSize = 1024 >  12 : : class Queue {  13 : : public:  14 : : Queue()  15 : 1121280 : :mFront( -1 ),  16 : 1121280 : mRear( -1 ) {  17 : : }  18 : :   19 : : bool  20 [ + + ][ + - ]: 1144556 : empty() const {  [ + + ][ + + ]  21 : : return mFront == mRear;  22 : : }  23 : :   24 : : bool  25 [ - + ][ + - ]: 12368 : full() const {  [ - + ][ + - ]  26 : : return (mRear + 1) % cQSize == mFront;  27 : : }  28 : :   29 : : bool  30 : : push( \_Tp inData )  31 [ + - ][ + - ]: 12368 : {  32 : 12368 : if( !full() )  33 : : {  34 : 12368 : mRear = ( mRear + 1 ) % cQSize;  35 : 12368 : mQueue[ mRear ] = inData;  36 : 0 : return true;  37 : : }  38 : : return false;  39 : : }  40 : :   41 : : bool  42 : : pop( \_Tp &outValue )  43 [ + - ]: 10908 : {  44 : 12368 : if( !empty() )  45 : : {  46 [ - + ][ - + ]: 12368 : mFront = ( mFront + 1 ) % cQSize;  47 : 10908 : outValue = mQueue[ mFront ];   48 : 0 : return true;  49 : : }  50 : : return false;  51 : : }  52 : : private:  53 : : ssize\_t mFront;  54 : : ssize\_t mRear;  55 : : \_Tp mQueue[cQSize];  56 : : };  57 : :   58 : : } //namespace \_impl  59 : :   60 : : class MemMgr  61 : : {  62 : : public:  63 : : MemMgr()  64 : 1095 : :mMutex(),  65 [ + + ][ # # ]: 1121280 : mOffset() {  [ # # ]  66 : : }  67 : :   68 : 1095 : ~MemMgr() {  69 [ + + ]: 1122375 : for( size\_t i = 0; i < 1024; i++ )  70 : : {  71 : 1132188 : while( !mMemory[ i ].empty() )  72 : : {  73 : 10908 : void \*ptr;  74 : 10908 : if( mMemory[ i ].pop( ptr ) )  75 : : {  76 : 10908 : nftFree( ptr );  77 : : }  78 : : }  79 : : }  80 : : }  81 : :   82 : : static MemMgr \*  83 : 51662 : instance() {  84 [ + - ]: 25831 : static MemMgr obj;  [ + - - + ]  [ # # ][ - + ]  [ # # ][ - + ]  [ # # ]  85 : : return &obj;  86 : : }  87 : :   88 : : void \*  89 : 37104 : getMemory( size\_t inSize ) throw(DPL::Error \*) {  90 : 12368 : boost::mutex::scoped\_lock l(mMutex);  91 : 12368 : size\_t index = (inSize >> 1) - mOffset;  92 [ + - ]: 12368 : if( index < 1024 )  93 : : {  94 : 12368 : if( !mMemory[ index ].empty() )  95 : : {  96 : 1460 : void \*ptr;  97 : 1460 : if( mMemory[ index ].pop( ptr ) )  98 : 1460 : {  99 : : return ptr;  100 : : }  101 : : }  102 : : }  103 : 10908 : void \*outMemory = nftMalloc( inSize );  104 : : //#ifdef ERROR\_SEED //<<Test\_Id:outMemory\_Set\_To\_NULL>>  105 : : // outMemory=NULL;  106 : : //#endif  107 [ - + ]: 10908 : if(NULL == outMemory){  108 : 0 : Logger::instance()->log(LogMessageTable::eMEMORY\_ALLOC);  109 : 0 : int iErrNo = errno; //System error no  110 : 0 : char systemErrorMsg[BUFSIZ];  111 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  112 : 0 : pErr->errorAdd(NULL, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  113 : : ErrorHandler::eErrMemoryAllocation,  114 : : pErr->getErrorMsg(  115 : : ErrorHandler::eErrMemoryAllocation,  116 : : strerror\_r( iErrNo, systemErrorMsg, BUFSIZ )));  117 : : }  118 : 10908 : return outMemory;  119 : : }  120 : :   121 : : void  122 : 24736 : releaseMemory( void \*pMem, size\_t inSize ) {  123 : 12368 : boost::mutex::scoped\_lock l(mMutex);  124 : 12368 : size\_t index = (inSize >> 1) - mOffset;  125 [ + - ][ + - ]: 12368 : if( index < 1024 )  126 : : {  127 : 12368 : if( mMemory[ index ].push( pMem ) )  128 : 12368 : {  129 : : return;  130 : : }  131 : : }  132 : 0 : nftFree( pMem );  133 : : }  134 : :   135 : : void  136 : 2190 : setOffset( size\_t inOffset ) {  137 : 1095 : mOffset = inOffset;  138 : : }  139 : :   140 : : private:  141 : : typedef void \* Data;  142 : : typedef \_impl::Queue<Data> MemList;  143 : :   144 : : boost::mutex mMutex;  145 : : size\_t mOffset;  146 : :   147 : : MemList mMemory[ 1024 ];  148 : : };  149 : :   150 : : } //namespace nft  151 : :   152 : : #endif |

|  |
| --- |
| Generated by: [LCOV version 1.9](http://ltp.sourceforge.net/coverage/lcov.php) |