| LCOV - code coverage report |
| --- |
|  |
| | Current view: | [top level](http://docs.google.com/index.html) - [Common/src](http://docs.google.com/index.html) - FileMap.cpp (source / [functions](http://docs.google.com/FileMap.cpp.func.html)) |  |  | Hit | Total | Coverage | | --- | --- | --- | --- | --- | --- | --- | | Test: | EBM9K Converter Coverage Report |  | Lines: | 32 | 41 | 78.0 % | | Date: | 2012-03-23 |  | Functions: | 7 | 8 | 87.5 % | | Legend: | Lines: hit not hit | Branches: + taken - not taken # not executed |  | Branches: | 5 | 16 | 31.2 % | |  | |
|  |

|  |
| --- |
| Branch data Line data Source code  1 : : #include <fcntl.h>  2 : : #include <sys/types.h>  3 : : #include <sys/stat.h>  4 : : #include <sys/mman.h>  5 : : #include "FileMap.h"  6 : : #include <iostream>  7 : : #include <cstring>  8 : : #include <errno.h>  9 : : #include "RetryFunc.h"  10 : : #include "print.h"  11 : :   12 : : namespace nft {  13 : : //Constructor  14 : 36450 : FileMap::FileMap() {  15 : 18225 : mMmapStore.mFd = -1;  16 : 18225 : mMmapStore.mMemAddr = NULL;  17 : 18225 : mMmapStore.mFileSize = 0;  18 : : }  19 : :   20 : : //Destructor  21 : 36450 : FileMap::~FileMap() {  22 : 18225 : try{  23 : 18225 : unmapFileFromMemory();  24 [ # # ][ # # ]: 0 : }catch (DPL::Error \*e)  25 : : {  26 : 0 : e9k\_error("Could not unmap file from Memory\n");  27 : : }  28 : : }  29 : :   30 : : //Map file to memory  31 : : void \* FileMap::mapFileToMemory(const char \*fileName)  32 : 12583 : {  33 : 12583 : char \*addr = NULL;  34 : 12583 : void \*memAddr = NULL;  35 : 12583 : struct stat stat\_buf ;  36 : :   37 : 12583 : mFileName = fileName;  38 : :   39 : 12583 : if(-1 == stat(mFileName.c\_str(), &stat\_buf))  40 : : {  41 : 24 : return NULL;  42 : : }  43 : :   44 [ - + ]: 12559 : if( -1 == (mMmapStore.mFd = nftOpenWithRetry(fileName, O\_RDONLY)))  45 : : {  46 : 0 : return NULL;  47 : : }  48 : :   49 : 12559 : mMmapStore.mFileSize = stat\_buf.st\_size;  50 : :   51 : 12559 : addr = (char\*)0;  52 : 12559 : memAddr = nftMmapWithRetry(addr, mMmapStore.mFileSize, PROT\_READ, MAP\_SHARED, mMmapStore.mFd, 0);  53 : :   54 : : //Exception::Test   55 : : //memAddr = NULL;  56 [ - + ]: 12559 : if(MAP\_FAILED == memAddr || NULL == memAddr)  57 : : {  58 : 0 : return NULL;  59 : : }  60 : :   61 : 12559 : mMmapStore.mMemAddr = memAddr;  62 : 12583 : return memAddr;  63 : : }  64 : :   65 : : //Unmap file from memory  66 : 18225 : void FileMap::unmapFileFromMemory()throw (DPL::Error \*) {  67 [ + + ]: 18225 : if( 0 >= mMmapStore.mFileSize)  68 : : {  69 : 12559 : return;  70 : : }  71 : : else  72 : : {  73 : : //Unmap the file  74 : 12559 : try{  75 : 12559 : nftMunmapWithRetry(mMmapStore.mMemAddr, mMmapStore.mFileSize);  76 : : }  77 [ # # ]: 0 : catch(DPL::Error \*pChild )  78 : : {  79 : 0 : ErrorHandler \*pErr = ErrorHandler::instance();  80 : 0 : pErr->errorAdd(pChild, \_\_FILE\_\_, \_\_FUNCTION\_\_, \_\_LINE\_\_,  81 : : ErrorHandler::eErrFunction,  82 : : pErr->getErrorMsg(  83 : : ErrorHandler::eErrFunction,  84 : : "FileMap::unmapFileFromMemory"));  85 : : }  86 : :   87 [ + - ]: 12559 : if(mMmapStore.mFd)  88 : : {  89 : 12559 : nftCloseWithRetry(mMmapStore.mFd);  90 : : }  91 : 12559 : mMmapStore.mFileSize = 0;  92 [ # # ]: 18225 : mMmapStore.mFd = -1;  93 : : }  94 : : }  95 : :   96 : 0 : void FileMap::setFileName(std::string value) {  97 : 0 : mFileName = value;  98 : : }  99 : :   100 : 1215 : } //namespace nft |

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