Click to verify

KDD_15 2019-03-05 12:25:21 root@ubuntu:/usr/src/linux-5.0# make modules_install INSTALL arch/x86/crypto/aegis128-aesni.ko At main.c:160: - SSL error:02001002:system library:fopen:No such file or directory: ../crypto/bio/bss_file.c:74 - SSL error:2006D080:BIO routines:BIO_new_file:no such file: ../crypto/bio/bss_file.c:81 sign-file: certs/signing_key.pem: No such file or directory scripts/Makefile.modinst:36: recipe for target 'arch/x86/crypto/aegis128-aesni.ko' failed make[1]: *** [arch/x86/crypto/aegis128-aesni.ko] Error 1 Makefile:1281: recipe for target '_modinst_' failed make: *** [_modinst_] Error 2 求救！！！真的做到自闭了！！！！！！ ...全文 比【道友老李】来帮你解答，本回答参考gpt编写，并整理提供，如果还有疑问可以点击头像关注私信或评论。如果答案让您满意，请采纳、关注，非常感谢！在Visual Studio 2022中创建 C++ 项目并编写代码后，若发现未生成 `.exe` 文件，通常是由于以下原因造成的。下面将详细描述可能的问题及其解决方案。 首先，要确保项目被正确配置为生成可执行文件。请按照以下步骤检查配置： - 步骤1: 打开你的项目，右键点击项目名称（通常位于解决方案资源管理器中），选择“属性”。 步骤2: 在“配置属性”中，查看“输出目录”设置，确保输出路径正确。在默认情况下，它应该指向 $(SolutionDir)$(Configuration)\。步骤3: 在“配置属性” > “常规”中，检查“配置类型”。确保其设置为“应用程序(.exe)”而不是“静态库(.lib)”或其他类型。 2. 生成项目 确保你已经生成了项目。可以通过以下方式尝试生成项目： - 步骤1: 点击菜单栏中的“生成” > “生成解决方案”，或者直接按 Ctrl + Shift + B。 - 步骤2: 观察输出窗口，确保没有错误信息。如果出现错误，修复这些错误后再进行构建。 3. 检查代码中的错误 如果代码中存在语法错误或逻辑错误，也可能导致编译失败，因此无法生成 .exe 文件。请确保代码中没有语法错误。 案例： 简单的“Hello World”程序 #include int main() { std::cout “生成解决方案”。 6. 确认Visual Studio的安装有时候，如果安装的工具集不完整或有损坏，可能也会导致生成失败。考虑重新安装 Visual Studio 或尝试修复。 总结 若在Visual Studio 2022中创建 C++ 项目未生成 .exe 文件，首先检查项目配置、生成状态和代码正确性，确保输出路径无误。如果经过以上步骤问题仍未解决，可以考虑重建项目或修复 Visual Studio 的安装。 希望这些信息能够帮助你解决问题！如有其他疑问，请随时询问。

ponydph 2014-07-09 09:07:44 从网络上下载的例子，编译成功后，但是没有生成exe程序。但F5调试的时候 提示没有生成exe程序，很奇怪。 如何查找原因？ ...全文 birdflyw 2020-03-29 11:31:49 BUILD: Doing 'make -j4' parallel build Warning: Kernel ABI header at 'tools/include/uapi/drm/i915_drm.h' differs from latest version at 'include/uapi/drm/i915_drm.h' Warning: Kernel ABI header at 'tools/include/uapi/linux/kvm.h' differs from latest version at 'include/uapi/linux/kvm.h' Warning: Kernel ABI header at 'tools/include/uapi/linux/prctl.h' differs from latest version at 'include/uapi/linux/prctl.h' Warning: Kernel ABI header at 'tools/arch/x86/include/asm/disabled-features.h' differs from latest version at 'arch/x86/include/asm/disabled-features.h' Warning: Kernel ABI header at 'tools/arch/x86/include/asm/cpufeatures.h' differs from latest version at 'arch/x86/include/asm/cpufeatures.h' Warning: Kernel ABI header at 'tools/arch/x86/include/uapi/asm/kvm.h' differs from latest version at 'arch/x86/include/uapi/asm/kvm.h' Warning: Kernel ABI header at 'tools/arch/powerpc/include/uapi/asm/kvm.h' differs from latest version at 'arch/powerpc/include/uapi/asm/kvm.h' Warning: Kernel ABI header at 'tools/arch/s390/include/uapi/asm/kvm.h' differs from latest version at 'arch/s390/include/uapi/asm/kvm.h' Makefile.config:137: *** Error: flex is missing on this system, please install it. 停止。 Makefile.perf:203: recipe for target 'sub-make' failed make[1]: *** [sub-make] Error 2 Makefile:69: recipe for target 'all' failed make: *** [all] Error 2 始终没有找到解决方案，大佬们求教55555 ...全文 动态链接库(Dynamic Link Library,简称DLL)是一种特殊的文件格式，它可以被多个程序同时共享和调用。当一个程序需要使用某个功能时，它会请求操作系统加载相应的DLL文件，并将函数地址注入到程序的内存空间中。这样，程序就可以像使用本地函数一样调用DLL中的函数，而不需要知道DLL的具体实现细节。 动态链接库的主要优点是它可以减少程序的大小，因为多个程序可以共享相同的DLL文件。此外，通过动态链接库，程序可以在运行时加载和卸载所需的功能，这使得程序更加灵活和可扩展。 前置条件 安装vs 2022社区版和“使用C++的桌面开发”相关组件，具体可以阅读： [Visual Studio] 基础教程 - Window10下如何安装VS 2022社区版 演练介绍 本文将会创建一个MathLibrary 的C++ 动态连接库和一个MathClient的 C++控制台程序去调用MathLibrary.dll，2个项目都放置在同一个解决方案的“DLLDemo”目录中，最后效果如下图所示： 创建空白解决方案 1) 在菜单栏上，选择“文件”>“新建”>“项目”，打开“创建新项目”对话框 ，选择“空白解决方案”。 创建 DLL 项目 将创建一个 DLL 项目，添加代码，并生成它。 首先，启动 Visual Studio IDE，并在需要时登录。 在 Visual Studio 2022 中创建 DLL 项目 1) 在菜单栏上，选择“文件”>“新建”>“项目”，打开“创建新项目”对话框 。 2) 在对话框顶部，将“语言”设置为“C++”，将“平台”设置为“Windows”，并将“项目类型”设置为“库”。 3) 从经过筛选的项目类型列表中，选择“动态链接库(DLL)”，然后选择“下一步”。 4) 在“配置新项目”页面，在“项目名称”框中输入“MathLibrary”，以指定项目的名称。 把默认“位置”修改为你希望保存的目录。 5) 选择“创建”按钮创建项目。 创建解决方案后，就可以在 Visual Studio 中的“解决方案资源管理器”窗口中看到生成的项目和源文件了。 备注：这里我使用了“解决方案文件夹”来给相关Demo做归类，这也是为什么上面步骤 4) 中修改了项目保存的位置目录。 目前，此 DLL 并没有起到很大的作用。 接下来，创建一个头文件来声明 DLL 导出的函数，然后将函数定义添加到 DLL，使其具备更强大的功能。 将头文件添加到 DLL 1) 若要为函数创建头文件，请在菜单栏上选择“项目”>“添加新项”。 2) 在“添加新项”对话框的左窗格中，选择“Visual C++”。 在中间窗格中，选择 “头文件(.h)”。 指定 MathLibrary.h 作为头文件的名称。 3) 选择“添加”按钮以生成一个空白头文件，该文件显示在新的编辑器窗口中。 4) 将头文件的内容替换为以下代码： extern "C" MATHLIBRARY_API void fibonacci_init(const unsigned long long a, const unsigned long long b); extern "C" MATHLIBRARY_API bool fibonacci_next(); extern "C" MATHLIBRARY_API unsigned long long fibonacci_current(); extern "C" MATHLIBRARY_API unsigned fibonacci_index(); 此头文件声明一些函数以生成通用 Fibonacci 序列，给定了两个初始值。 调用 fibonacci_init(1, 1) 会生成熟悉的 Fibonacci 数字序列。 请注意文件顶部的预处理器语句。 DLL 项目的新项目模板会将 _EXPORTS 添加到定义预处理器宏。 在此示例中，Visual Studio 在生成 MathLibrary DLL 项目时定义 MATHLIBRARY_EXPORTS。 定义 MATHLIBRARY_EXPORTS 宏时，MATHLIBRARY_API 宏会对函数声明设置 __declspec(dllexport) 修饰符。 此修饰符指示编译器和链接器从 DLL 导出函数或变量，以便其他应用程序可以使用它。 如果未定义 MATHLIBRARY_EXPORTS（例如，当客户端应用程序包含头文件时），MATHLIBRARY_API 会将 __declspec(dllimport) 修饰符应用于声明。 此修饰符可优化应用程序中函数或变量的导入。 有关详细信息，请参阅 dllexport、dllimport。 向 DLL 添加实现 1) 在“解决方案资源管理器”中，右键单击“源文件”节点并选择“添加”>“新建项目”。 使用上一步中添加新头文件的相同方式，创建名为 MathLibrary.cpp 的新 .cpp 文件。 2) 在编辑器窗口中，选择 MathLibrary.cpp 的选项卡（如果已打开）。 如果未打开，请在“解决方案资源管理器”中，双击 MathLibrary 项目的“源文件”文件夹中的 MathLibrary.cpp，将其打开。 3) 在编辑器中，将 MathLibrary.cpp 文件的内容替换为以下代码： static unsigned long long previous_; static unsigned long long current_; static unsigned index_; void fibonacci_init(const unsigned long long a, const unsigned long long b) { index_ = 0; current_ = a; previous_ = b; } bool fibonacci_next() { if ((ULLONG_MAX - previous_ < current_) || (UINT_MAX == index_)) { return false; } if (index_ > 0) { previous_ += current_; } std::swap(current_, previous_); ++index_; return true; } unsigned long long fibonacci_current() { return current_; } unsigned fibonacci_index() { return index_; } 若要验证到目前为止是否一切正常，请编译动态链接库。 若要编译 ，请在菜单栏上选择“生成”>“生成解决方案”。 DLL 和相关编译器输出放在解决方案文件夹正下方的“Debug”文件夹中。 如果创建发布版本，该输出会放置在“Release”文件夹中。 输出应类似于： 已启动重新生成... 1>------ 已启动全部重新生成: 项目: MathLibrary, 配置: Debug x64 ------ 1>pch.cpp 1>dllmain.cpp 1>MathLibrary.cpp 1>正在生成代码... 正在创建库 D:\my_project\VCXXTutorials\x64\Debug\MathLibrary.lib 和对象 D:\my_project\VCXXTutorials\x64\Debug\MathLibrary.exp 1>MathLibrary.vcxproj -> D:\my_project\VCXXTutorials\x64\Debug\MathLibrary.dll ========== “全部重新生成”: 1 成功，0 失败，0已跳过 ========== ========== 重新生成 开始于 8:43 AM，并花费了 11.824 秒 ========== 从日志上看，对应的lib和dll文件保存在解决方案目录了，我们需要修改到项目目录： 修改前： 修改后： 重新构建： 创建可使用 DLL 的客户端应用 创建 DLL 时，请考虑客户端应用如何使用它。 若要调用函数或访问由 DLL 导出的数据，客户端源代码必须在编译时具有可用的声明。 在链接时间，链接器需要信息来解析函数调用或数据访问。 而 DLL 在“导入库”中提供此信息，导入库是包含有关如何查找函数和数据的信息的文件，而不是实际代码。 而在运行时，DLL 必须可供客户端使用，位于操作系统可以找到的位置。 无论是你自己的还是来自第三方的信息，客户端应用项目都需要几条信息才能使用 DLL。 它需要查找声明 DLL 导出的标头、链接器的导入库和 DLL 本身。 一种解决方案是将所有这些文件复制到客户端项目中。 对于在客户端处于开发阶段时不太可能更改的第三方 DLL，此方法可能是使用它们的最佳方法。 但是，如果还要同时生成 DLL，最好避免重复。 如果创建了正在开发的 DLL 文件的本地副本，可能会意外更改一个副本而不是另一个中的头文件，或使用过期的库。 为避免不同步的代码，建议在客户端项目中设置包含路径，使其直接包括 DLL 项目中的 DLL 头文件。 此外，在客户端项目中设置库路径以包括 DLL 项目中的 DLL 导入库。 最后，将生成的 DLL 从 DLL 项目复制到客户端生成输出目录中。 此步骤允许客户端应用使用生成的同一 DLL 代码。 在 Visual Studio 2022 中创建客户端应用 1) 在菜单栏上，选择“文件”>“新建”>“项目”，打开“创建新项目”对话框。 2) 在对话框顶部，将“语言”设置为“C++”，将“平台”设置为“Windows”，并将“项目类型”设置为“控制台”。 3) 从筛选的项目类型列表中，选择“控制台应用”，然后选择“下一步”。 4) 在“配置新项目”页面，在“项目名称”框中输入“MathClient”，以指定项目的名称。 把默认“位置”修改为你希望保存的目录。 5) 选择“创建”按钮创建客户端项目。 将为你创建一个最小的控制台应用程序项目。 主源文件的名称与你之前输入的项目名称相同。 在本例中，命名为 MathClient.cpp。 可以生成它，但它还不会使用你的 DLL。 接下来，要在源代码中调用 MathLibrary 函数，你的项目必须包括 MathLibrary.h 文件。 可以将此头文件复制到客户端应用项目中，然后将其作为现有项添加到项目中。 对于第三方库，此方法可能是一个不错的选择。 但是，如果同时处理 DLL 的代码和客户端的代码，则头文件可能会变为不同步。要避免此问题，请设置项目中的“附加包含目录”路径，使其包含指向原始标头的路径。 将 DLL 标头添加到包含路径 1) 右键单击“解决方案资源管理器”中的“MathClient”节点以打开“属性页”对话框。 2) 在“配置”下拉框中，选择“所有配置”（如果尚未选择）。 3) 在左窗格中，选择“配置属性”>“C/C++”>“常规”。 4) 在属性窗格中，选择“附加包含目录”编辑框旁的下拉控件，然后选择“编辑”。 5) 在“附加包含目录”对话框的顶部窗格中双击以启用编辑控件。 或者，选择文件夹图标以创建新条目。 6) 在编辑控件中，指定指向 MathLibrary.h 头文件的位置的路径。 可选择省略号（...）控件浏览到正确的文件夹。 还可将客户端源文件中的相对路径输入到包含 DLL 头文件的文件夹。 如果已按照指示将客户端项目置于 DLL 的相同的解决方案中，则相对路径可能如下所示： ..\MathLibrary 如果 DLL 和客户端项目位于其他文件夹中，请调整相对路径以进行匹配。 或者，使用省略号控件浏览文件夹。 7) 在“附加包含项目”对话框中输入标头文件的路径后，选择“确定”按钮。 在“属性页”对话框中，选择“确定”按钮以保存更改。 现在可以包括 MathLibrary.h 文件，并使用它在客户端应用程序中声明的函数。 使用以下代码替换 MathClient.cpp 的内容： int main() { fibonacci_init(1, 1); do { std::cout “开始执行(不调试)”。 此时，Visual Studio 会打开一个命令窗口，供程序在其中运行。 输出的开始一部分应如下所示： 按任意键关闭命令窗口。 现在，你已创建一个 DLL 和一个客户端应用程序，可以进行试验。 尝试在客户端应用的代码中设置断点，并在调试器中运行该应用。 看看单步执行库调用时会发生什么情况。 将其他函数添加到库中，或编写另一个使用 DLL 的客户端应用。 部署应用时，还必须部署它使用的 DLL。 若要使你生成的或从第三方加入的 DLL 可用于应用，最简单的方法就是将其放在应用所在的同一目录中。 这称为“应用本地部署”。 有关部署的更多信息，请参阅 Deployment in Visual C++。 参考资料 演练：创建和使用自己的动态链接库 (C++) | Microsoft Learn 文章来源：

 ••权声明：

本文为博主原创文章，遵循CC 4.0 BY-SA 知识共享协议，转载请附上原文出处链接和本声明。 正常来说 ，用VS开发WIN32 程序，会在debug路径下得到一个 exe文件 但是我得到的却是一个 XXX.exe.recipe 更诡异的，全网 都没有找到 recipe的解释 ，彷彿是一个全新的问题 另外就是，如果修改文件格式 ，得到exe 文件是无法运行的，提醒如下： *** 关键词：VS2019 ，visual studio 集成电路科普者 2025-05-11 05:05 采纳率: 0% 在使用GCC编译C/C++程序时，遇到“collect2.exe: error: ld returned 1 exit status”错误，通常表明链接阶段出现问题。以下是一些常见原因及解决方法：1) 检查源文件是否全部编译，确保所有相关文件都已包含在编译命令中；2) 确认库文件路径正确，使用-L指定库路径，-l指定库名称；3) 避免函数或变量重复定义，检查头文件是否正确使用#include防护；4) 核实依赖库版本兼容性，更新或替换不兼容的库。例如，尝试将命令改为“gcc -o test test.c -lm”以链接数学库。若问题依旧，可使用“ldd”工具检测动态链接库缺失情况，或查阅编译器具体报错信息定位问题根源。 收起 关注 在使用GCC编译C/C++程序时，如果遇到“collect2.exe: error: ld returned 1 exit status”错误，这通常表明链接阶段出现问题。以下是该错误的基本解释及其常见原因： 链接器（ld ）无法找到某些符号定义，可能是函数或变量。 库文件路径不正确，或者缺少必要的依赖库。 源文件未被完全编译，导致链接阶段缺失目标文件。 为解决此问题，首先需要确认以下几点是否正确设置： 所有相关源文件都已包含在编译命令中。 库路径和名称通过-L和-l参数正确指定。 常见问题分析 接下来，我们深入分析可能导致此错误的几个常见原因，并提供相应的解决方案： 问题描述可能原因解决方案部分源文件未参与编译编译命令中遗漏了某些源文件确保所有源文件均包含在编译命令中，例如：gcc -o test main.c util.c -lm链接器找不到库文件库路径或库名称指定错误使用-L指定库路径，-l指定库名称，例如：gcc -o test test.c -L/path/to/lib -lmylib 3. 高级诊断与解决方案 如果上述方法未能解决问题，可以尝试以下高级诊断步骤： # 检查动态链接库是否完整 ldd your_program # 查阅具体报错信息 gcc -o test test.c -lm 2>&1 | grep "undefined reference" 此外，还可以通过流程图进一步理解问题定位的逻辑： graph TD; A[检查源文件完整性] --> B{源文件齐全？}; B --否--> C[补充遗漏的源文件]; B --是--> D[检查库路径与名称]; D --> E{库路径正确？}; E --否--> F[修正-L和-l参数]; E --是--> G[检查重复定义]; G --> H{存在重复定义？}; H --是--> I[修正头文件防护]; H --否--> J[验证库版本兼容性]; 本回答被题主选为最佳回答 ，对您是否有帮助呢? 本回答被专家选为最佳回答 ，对您是否有帮助呢? 本回答被题主和专家选为最佳回答 ，对您是否有帮助呢? 评论 查看更多回答(0条) 子夜微光 2008-10-01 12:38:28 前几天还用makefile编译了个小工程，可能是装什么软件或是改环境变量了。今天就出现了这个错误！！ process_begin: CreateProcess(NULL,gcc -c test.c, ...)failed. make(e=2): 系统找不到指定的文件 make:*** [test.o] 错误2 这个错误折磨了我两天了！谁来帮帮我啊！！ 谢谢谢谢谢谢谢谢！！！ ...全文 配置latex的时候报错。环境变量已经设置了。网上的大部分回答都试过了没有用报错日志： 其中有一处错误来自texlive中的文件，代码放在最后。好像是字符集的问题？ 下面是配置文件和texlive文件 问题相关代码，请勿粘贴截图 配置文件 { "latex-workshop.latex.recipes": [{ "name": "xelatex", "tools": ["xelatex"] }, { "name": "latexmk", "tools": ["latexmk"] }, { "name": "pdflatex -> bibtex -> pdflatex*2", "tools": ["pdflatex", "bibtex", "pdflatex", "pdflatex"] }, { "name": "xelatex -> bibtex -> xelatex*2", "tools": ["xelatex", "bibtex", "xelatex", "xelatex"] }], "latex-workshop.latex.tools": [{ "name": "latexmk", "command": "latexmk", "args": ["-synctex=1", "-interaction=nonstopmode", "-file-line-error", "-pdf", "%DOCFILE%"] }, { "name": "xelatex", "command": "xelatex", "args": ["-synctex=1", "-interaction=nonstopmode", "-file-line-error", "%DOCFILE%"] }, { "name": "pdflatex", "command": "pdflatex", "args": ["-synctex=1", "-interaction=nonstopmode", "-file-line-error", "%DOCFILE%"] }, { "name": "bibtex", "command": "bibtex", "args": ["%DOCFILE%"] }], "latex-workshop.view.pdf.viewer": "tab", "latex-workshop.latex.clean.fileTypes": ["*.aux", "*.bbl", "*.blg", "*.idx", "*.ind", "*.lof", "*.lot", "*.out", "*.toc", "*.acn", "*.acr", "*.alg", "*.glg", "*.glo", "*.gls", "*.ist", "*.fls", "*.log", "*.fdb_latexmk"] } texlive文件 %% %% This is file `ctex-fontset-fandol.def', %% generated with the docstrip utility. %% %% The original source files were: %% %% ctex.dtx (with options: `fontset,fandol') %% %% Copyright (C) 2003--2021 %% CTEX.ORG and any individual authors listed in the documentation. %% ---------- %% %% This work may be distributed and/or modified under the %% conditions of the LaTeX Project Public License, either %% version 1.3c of this license or (at your option) any later %% version. This version of this license is in %% %% and the latest version of this license is in %% %% and version 1.3 or later is part of all distributions of %% LaTeX version 2005/12/01 or later. %% %% This work has the LPPL maintenance status "maintained". %% %% ---------- %% \GetIdInfo$Id: ctex.dtx 07425fe 2021-03-14 16:43:36 +0800 Xiangdong Zeng $ {Fandol fonts definition (CTEX)} \ProvidesExplFile{ctex-fontset-fandol.def} {\ExplFileDate}{2.5.6}{\ExplFileDescription} \ctex_fontset_case:nnnn { \ctex_fontset_error:n { fandol } } { \ctex_zhmap_case:nnn { \setCJKmainfont { FandolSong-Regular.otf } { cmap = UniGB-UTF16-H, BoldFont = FandolSong-Bold.otf, ItalicFont = FandolKai-Regular.otf } \setCJKsansfont { FandolHei-Regular.otf } { cmap = UniGB-UTF16-H, BoldFont = FandolHei-Bold.otf } \setCJKmonofont { FandolFang-Regular.otf } { cmap = UniGB-UTF16-H } \setCJKfamilyfont { zhsong } { FandolSong-Regular.otf } { cmap = UniGB-UTF16-H, BoldFont = FandolSong-Bold.otf, ItalicFont = FandolKai-Regular.otf } \setCJKsansfont { FandolHei-Regular.otf } { cmap = UniGB-UTF16-H, BoldFont = FandolHei-Bold.otf } \setCJKmonofont { FandolFang-Regular.otf } { cmap = UniGB-UTF16-H } \setCJKfamilyfont { zhsong } { FandolSong-Regular.otf } { FandolSong-Bold.otf } { FandolKai-Regular.otf } { FandolHei-Regular.otf } { FandolHei-Bold.otf } { FandolFang-Regular.otf } \ctex_set_upfamily:nnn { zhsong } { upzhserif } { upzhserifb } \ctex_set_upfamily:nnn { zhhei } { upzhsans } { upzhsansb } \ctex_set_upfamily:nnn { zhfs } { upzhmono } {} \ctex_set_upfamily:nnn { zhkai } { upzhserifit } {} } { \setCJKmainfont { FandolSong-Regular } [Extension = .otf, BoldFont = FandolSong-Bold, ItalicFont = FandolKai-Regular] \setCJKsansfont { FandolHei-Regular } [Extension = .otf, BoldFont = FandolHei-Bold] \setCJKmonofont { FandolFang-Regular } [Extension = .otf] \setCJKfamilyfont { zhsong } { FandolSong-Regular } [Extension = .otf, BoldFont = FandolSong-Bold] \setCJKfamilyfont { zhhei } { FandolHei-Regular } [Extension = .otf, BoldFont = FandolHei-Bold] \setCJKfamilyfont { zhfs } { FandolFang-Regular } [Extension = .otf] \setCJKfamilyfont { zhkai } { FandolKai-Regular } [Extension = .otf] } \NewDocumentCommand \songti { } { \CJKfamily { zhsong } } \NewDocumentCommand \heiti { } { \CJKfamily { zhhei } } \NewDocumentCommand \fangsong { } { \CJKfamily { zhfs } } \NewDocumentCommand \kaishu { } { \CJKfamily { zhkai } } %% %% %% End of file `ctex-fontset-fandol.def'. 小齐大圣 2019-12-17 09:32:10 ...全文