

实时嵌入式系统的图形界面支持系统 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022__E5_AE_9E_

[E6_97_B6_E5_B5_8C_E5_c103_144589.htm](https://www.100test.com/kao_ti2020/144/2021_2022__E5_AE_9E_E6_97_B6_E5_B5_8C_E5_c103_144589.htm) 1 Linux 下的软件维护和建立工具 对开放源码的自由软件来说，程序员得到的通常是源代码。在编译源代码并正确安装和配置的过程中，往往会涉及到许多工具和函数库，因此其过程经常显得有些繁复。MiniGUI也不例外。为了说明 MiniGUI 的正确编译和安装过程，我们有必要首先了解 Linux 系统下用于软件维护和建立的工具。

1.1 make 和 makefile

make 是 Linux 下最常用的二进制程序、函数库的建立生成工具。make 运行时要根据当前目录下的 makefile 文件（一般是 Makefile），确定要生成什么样的二进制文件，以及对应的命令。我们还可以在 makefile 文件中建立要生成的目标与源代码之间的依赖关系，从而可以让 make 工具根据时间自动判断是否需要通过中间过程而生成最终目标。尽管通过 makefile 文件可以组织一个大的项目，但往往手工编写一个 makefile 文件并不是一件轻松的事情，并且在需要维护一个源代码的目录树时，makefile 文件的维护工作就会大大增加。为此，GNU 又开发了 Autoconf/Automake 工具，可以用来自动生成 makefile 文件，并且能够检查系统的配置信息，从而帮助提供源代码的可移植性。

1.2 Autoconf/Automake

GNU 的 Autoconf 及 Automake 这两个软件实际是由若干 Shell 脚本组成的，它可以帮助程序员轻松产生 makefile 文件。现在的各种自由软件，如 Apache、MySQL 等都是利用 Autoconf, Automake 实现自动配置和编译的。MiniGUI 也采用了 Autoconf/Automake 接口。用户只要使

用"./configure"、"make"、"make install" 就可以把程序编译并安装到系统中。为了更好地了解 Autoconf 和 Automake，我们需要对 makefile 作一简单回顾。Makefile 基本上就是"目标"（target）、"依赖性"（dependencies）和"动作"三者所组成的一系列规则。而 make 就是根据 makefile 文件的规则决定如何编译（compile）和连接（link）程序或者其它动作。当然，make 可做的不只是编译和连接程序，例如 FreeBSD 的 port collection 中，Makefile 还可以做到自动下载远程程序，解压缩（extract），打补丁（patch），然后编译并安装到系统中。Makefile 基本结构虽然很简单，但是妥善运用这些规则就可以变换出许多不同的花样。却也因为这样，许多人刚开始学写 makefile 时会觉得没有规范可以遵循，每个人写出来的 makefile 都不大一样，不知道从哪里下手，而且常常会受到开发环境的限制，只要环境参数不同或者路径更改，可能 makefile 就得跟着修改。虽然有 GNU Makefile Conventions（GNU Makefile 惯例）制订出一些在进行 GNU 程序设计时编写 makefile 的一些标准和规范，但是其内容很长而且很复杂，并且经常作一些调整，为了减轻程序开发人员维护 makefile 的负担，就出现了 Automake。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com