

使用 `sudo` 配置超级用户特权

解释

熟悉有关 `sudo` 配置的基础知识，并了解如何使用 `sudo` 委托超级用户特权。

原因

某些命令需要管理员或 `root` 特权。使用 `sudo`，可以将执行这些命令的特权委托给特定的用户或组。

工作量

读完本文最多需要 20 分钟。编写第一个 `sudo` 配置规则只需几分钟时间，但建立一个可在环境中正常运行的 `sudo` 配置需要相当长的时间，具体取决于设置复杂性。

目标

了解 `sudo` 配置的基本方面。应对 `sudo` 配置的常见用例。了解如何在 `sudo` 设置中使用用户、用户组和别名。熟悉 `sudo` 最佳实践和查错。

要求

- 基本了解 sudo。

- root 特权。有关如何以普通用户身份使用 sudo 的信息，请参见 <https://documentation.suse.com/smart/systems-management/html/sudo-run-commands-as-superuser/index.html> 。

- 需要安装 sudo 软件包。

出版日期：2025 年 12 月 11 日

目录

- 1 **sudo** 配置简介 3
- 2 创建自定义 **sudo** 配置 3
- 3 更改 **sudo** 口令提示超时 8
- 4 使用 root 特权启动外壳 9
- 5 **sudo** 最佳实践 11
- 6 查错 12
- 7 **sudo** 配置参考 13
- 8 法律声明 16
- A GNU 自由文档许可证 17

1 **sudo** 配置简介

sudo 提供了一种安全有效的方式将超级用户特权委托给特定用户或组。

Linux 系统上的某些操作需要 root 或管理员特权。管理自己系统的家庭用户不必委托超级用户特权，因为在这种情况下，管理员和普通用户是同一个人。但是，一旦系统成为包含多个用户、组和主机的更大系统环境的一员，控制谁有权在哪个位置执行哪种操作就变得至关重要。同时，为所有用户和组提供必要的特权来执行其任务也很重要。

sudo 旨在帮助解决这些需求。它提供：

增强的系统安全性

sudo 提供对用户、组、主机和命令的精细控制，从而可以通过降低入侵者或系统用户恶意或意外造成损坏的风险，来提高系统安全性。

完整的审计追踪

每当用户切换特权时，这种行为就会出现在系统日志中，并且该用户以提升的特权执行的所有操作都可以追溯到该用户。

委托 root 特定任务的方式

使用 **sudo**，系统管理员可以允许单个用户或组执行特定的任务，而无需输入 root 口令并切换到 root 帐户。

！ 重要：如何阅读本文

本文提供深入的 **sudo** 配置信息。但是，它不提供任何有关如何构建全面且安全的 **sudo** 策略的建议。与安全相关的策略非常复杂，在很大程度上取决于它们是针对什么环境创建的。

2 创建自定义 **sudo** 配置

您需要了解如何构建简单的示例自定义 **sudo** 配置并逐步扩展它。创建组并使用别名可使自定义配置保持精简且高效。



警告：示例配置仅用于演示目的

下面所述的示例规则纯粹用于演示目的。请使用这些示例来了解 **sudo** 配置文件的一般语法。不要在现实设置中使用它们，因为它们不能反映这些环境的复杂性。

2.1 **sudo** 配置最佳实践

在开始之前，请注意以下有关维护 **sudo** 配置的几条基本规则：

始终使用 **visudo** 来编辑 **sudo** 配置文件

应使用 **visudo** 命令对 **sudo** 配置进行任何更改。**visudo** 是一个定制的工具，可用于编辑 **sudo** 配置文件和运行基本语法检查，以确保配置保持完整且能正常运行。错误的 **sudo** 配置可能导致用户被锁定在自己的系统之外。

始终在 `/etc/sudoers.d/` 下创建自定义配置

自定义配置必须位于 `/etc/sudoers.d/` 下才能由 **sudo** 拉入。自定义配置文件中的设置优先于 `/etc/sudoers` 包含的默认配置中的设置。

始终注意配置读取顺序

为确保以正确的顺序读取自定义配置，请在配置前面加上数字作为前缀。使用前导零来建立文件的读取顺序。例如，`01_myfirstconfig` 在 `10_myotherconfig` 之前分析。如果在某个文件中设置了一个指令，而该文件是在包含冲突信息的另一个文件之前读取的，则会应用最后读取的指令。

始终使用描述性文件名

使用能够提示配置文件作用的文件名。这可以帮助您跟踪 **sudo** 设置的预期功能。



提示：**sudo** 配置和不可变文件系统

不可变文件系统是一旦安装即无法更改的文件系统，它是只读的。如果您使用的 SUSE 产品依赖于不可变文件系统，则该产品附带的 **sudo** 默认配置将安装在 `/usr/etc/sudoers` 下，并且任何预先配置的调整都位于 `/usr/etc/sudoers.d/` 下。

您自己的自定义配置位于 `/etc/sudoers.d/` 下，并优先于 `/usr/etc/sudoers.d/` 中提供的任何内容。如果不存在现有的 `sudoers` 文件，**visudo** 命令可打开 `/usr/etc/sudoers` 并将更改的文件保存到 `/etc/sudoers`。如果已有该文件，**visudo** 将打开该文件并将数据写入其中。位于 `/etc/` 下的实例优先于位于 `/usr/etc/` 下的实例。这样，用户所做的配置调整就不会因更新而被破坏。

2.2 创建用户特定的配置文件

创建一个 **sudo** 配置文件，以允许普通用户 (`tux`) 使用自己的口令（而不是 `root` 口令）运行 **useradd** 命令。

例 1：创建用户特定的配置文件

1. 以系统管理员 (`root`) 身份启动 **visudo**，创建一个用于保存新的用户特定指令的自定义配置文件。使用带编号的描述性名称：

```
# visudo -f /etc/sudoers.d/02_usermanagement
```

2. 创建一条规则，以允许 `tux` 在此 **sudo** 配置应用到的整个环境中执行 `/usr/sbin/useradd` 二进制文件：

```
tux ❶ ALL ❷ = /usr/sbin/useradd ❸
```

- ❶ 指定用户或组。按名称或 `#UID` 列出用户，按 `%GROUPNAME` 列出组。如果此处有多个项目，请用逗号分隔。要否定项，请使用 `!`。
- ❷ 指定一个或多个主机（用逗号分隔）。使用完全限定的主机名或 IP 地址。添加 `ALL` 以在所有主机上全局强制实施此设置。使用 `!` 进行否定。
- ❸ 指定一个或多个可执行文件（用逗号分隔）。指定可执行文件时，请务必注意以下规则：

`/usr/sbin/useradd`

在不添加任何其他选项的情况下，此操作允许执行每个可能的 **useradd** 命令。

/usr/sbin/useradd -c

如果您明确指定某个选项，则该选项是唯一允许的选项。在上面指定的用户无法使用其他任何选项。

/usr/sbin/useradd ""

这样，用户只能单纯调用 **useradd**，而根本不能使用任何选项。

在上面的示例中，您可能希望允许使用所有选项和子命令，或者出于安全原因将允许的选项和子命令限制为少数几个，但在这种情况下，完全禁止用户指定任何选项是毫无意义的。

3. 要让用户使用自己的口令而不是 root 口令，请添加以下一行内容：

```
Defaults:tux !targetpw
```

激活后，此标志要求用户输入目标用户（即 root）的口令。此标志在任何 SLE Micro 系统上默认都处于启用状态。使用 ! 可以对其否定，要求用户只输入自己的口令而不是 root 口令。

4. 保存配置，退出编辑器，并打开另一个外壳来测试 **sudo** 是否遵循您的新配置。

2.3 通过对项目进行分组来创建自定义配置

修改例 1 “创建用户特定的配置文件”中的配置，使一组命名用户无需输入 root 口令即可运行 **useradd** 命令。另外，将 **usermod** 和 **userdel** 添加到此组可用的命令列表中。

例 2：通过对项目进行分组来创建自定义配置

1. 要修改示例配置，请以系统管理员身份使用 **visudo** 将其打开：

```
# visudo /etc/sudoers.d/02_usermanagement
```

2. 使用逗号分隔的列表将更多用户添加到规则：

```
tux, wilber ALL = /usr/sbin/useradd
```

3. 要允许列出的用户执行一系列命令，请将这些命令指定为逗号分隔的列表：

```
tux, wilber ALL = /usr/sbin/useradd, /usr/sbin/usermod, /usr/sbin/userdel
```

4. 要让列出的用户使用自己的口令而不是 `root` 口令，请添加以下一行内容：

```
Defaults:tux, wilber !targetpw
```

如果处于启用状态，此标志要求列出的用户输入目标用户（即 `root`）的口令。此标志在任何 SLE Micro 系统上默认都处于启用状态。使用 `!` 可以对其否定，要求列出的用户只输入自己的口令而不是 `root` 口令。

5. 保存配置，退出编辑器，并打开另一个外壳来测试 `sudo` 是否遵循您的新配置。

2.4 通过应用别名简化配置

使用别名可以进一步简化例 2 “通过对项目进行分组来创建自定义配置”中的自定义配置。对项目进行分组在一定程度上有所帮助，但使用用户、命令和主机全局别名是使 `sudo` 配置保持整洁精简的最有效方法。

使用别名和组而不是列表是处理设置更改的更好方法。如果某个用户离职，只需将其从别名声明文件内的全局 `User_Alias` 声明中去除即可，而无需寻源到所有单独的自定义配置文件。该过程同样适用于任何其他类型的别名（`Host_Alias`、`Cmdn_Alias` 和 `Runas_Alias`）。

例 3：通过应用别名简化配置

1. 创建一个新文件来保存全局别名定义：

```
# visudo /etc/sudoers.d/01_aliases
```

2. 添加以下一行内容以创建 `TEAMLEADERS` 别名：

```
User_Alias    TEAMLEADERS = tux, wilber
```

3. 添加以下一行内容以创建 `USERMANAGEMENT` 别名：

```
Cmdn_Alias    USERMANAGEMENT = /usr/sbin/useradd, /usr/sbin/usermod, /usr/
sbin/userdel
```

4. 保存更改并退出 `visudo`。
5. 以系统管理员身份启动 `visudo` 以编辑示例配置文件：

```
# visudo -f /etc/sudoers.d/02_usermanagement
```

6. 删除先前的规则并将其替换为以下规则，该规则使用您刚刚定义的别名：

```
TEAMLEADERS ALL = USERMANAGEMENT
```

7. 要让 `User_Alias` 定义的所有用户都使用自己的口令而不是 `root` 口令，请添加以下一行内容：

```
Defaults:TEAMLEADERS !targetpw
```

8. 保存配置，退出编辑器，并打开另一个外壳来测试 `sudo` 是否遵循您的新配置。



注意：更多信息

在第 7 节“[sudo 配置参考](#)”中查找 `sudo` 配置语法的更详细说明，并参见 `sudo` 的手册页。

3 更改 `sudo` 口令提示超时

了解如何更改超时设置，以便在执行需要 `root` 特权的命令时，不会每执行一个命令都出现输入 `root` 口令的提示。

首次运行以 `sudo` 开头的命令时，系统会提示您输入 `root` 口令。该口令将保持有效一段时间。一旦口令失效，系统会再次提示用户输入口令。要延长或缩短在执行需要 `root` 特权的命令时实施的超时，请对 `sudo` 配置文件进行以下更改。



警告：不要向 `root` 特权授予不受限的无口令访问权限

出于安全原因，请不要向 `root` 特权授予不受限的访问权限。相反，请设置合理的超时，以防任何入侵者滥用 `root` 帐户。

过程 1：更改 `sudo` 口令提示的超时

1. 以系统管理员身份，使用以下命令为时戳配置创建一个新的 `sudo` 配置文件：

```
# visudo --f=/etc/sudoers.d/timestamp_timeout
```

使用 `root` 口令成功完成身份验证后，将打开该文件。

2. 启用编辑并添加 `timestamp_timeout=` 一行。输入时戳值。

例如，要将超时缩短为三分钟，请输入：

```
timestamp_timeout=3
```

如果时戳设置为零，则每次执行 `sudo` 命令时，系统都会提示您输入 `root` 口令。

3. 保存更改并关闭该文件。

您已创建 `sudo` 配置文件，并缩短了执行 `sudo` 命令的超时设置。

4 使用 `root` 特权启动外壳

使用 `sudo -s` 或 `sudo -i` 命令以永久性 `root` 特权启动外壳。使用这两个命令时，系统只会提示您输入 `root` 口令一次。

4.1 `sudo -s` 与 `sudo -i` 的区别

如果每次以 `root` 身份运行命令都必须输入 `sudo`，您可能会觉得非常不便。您可以使用一个内置机制来以永久性 `root` 特权启动外壳。为此可以使用两个命令选项：

- `sudo -s` 使用当前用户的环境启动外壳，并提供几个特权控制措施。要运行此命令，请输入 `root` 口令。
- `sudo -i` 使用干净的环境将外壳作为交互式登录外壳启动。要运行此命令，请输入 `root` 口令。

使用这两个命令时，外壳将通过新环境启动，您将以 `root` 身份登录。在该外壳中执行的任何后续命令都将以提升的特权运行，您无需再次输入口令。当您关闭外壳时，此环境将会终止，在执行另一个 `sudo` 命令时，您必须再次输入口令。

4.2 使用 `sudo -s` 启动外壳

`sudo -s` 命令启动交互式非登录外壳。使用 `root` 口令成功完成身份验证后，所有后续命令都将以提升的特权执行。

`SHELL` 环境变量或用户的默认外壳指定了要打开哪个外壳。如果此变量为空，则会选取 `/etc/passwd` 中定义的外壳。

默认情况下，`sudo -s` 命令从前一用户的目录运行，因为目标用户会继承前一用户的环境。该命令还会记录到您的历史中。

要以永久提升的特权启动外壳，请输入以下命令：

```
tux:~ > sudo -s
[sudo] password for root:
root:/home/tux # exit
tux:~ >
```

提示符从 `>` 更改为 `#`。

您已使用永久提升的特权启动了外壳。执行所有后续命令时，系统不会再次提示输入口令。

4.3 使用 `sudo -i` 启动外壳

`sudo -i` 与 `sudo -s` 命令行选项类似，但它会启动交互式登录外壳。使用 `sudo -s` 命令时，目标用户将继承前一用户的环境。您可以使用 `sudo -i` 命令来防止出现这种情况，运行此命令后，目标用户将获得一个干净的环境，并可以在他们自己的 `$HOME` 目录开始工作。

要使用 `sudo -i` 运行命令，请输入以下命令：

```
tux:~ > sudo -i
[sudo] password for root:
root:~ # exit
```

```
tux:~ >
```

您已使用永久提升的特权启动了外壳，并且命令已记录到您的历史记录中。执行所有后续命令时，系统不会再次提示输入口令。

5 **sudo** 最佳实践

了解 **sudo** 的一些最佳实践，以控制系统访问权限并提高用户的工作效率。

对 **sudo** 配置进行全面测试和审计

要构建真正高效且安全的 **sudo** 配置框架，请制定例行性的定期测试和审计流程。找出可能的漏洞并予以处理。不要出于便利而疏忽安全。

将自定义的 **sudo** 配置保存在单独的文件中

sudo 的主要策略配置文件为 `/etc/sudoers`。此文件通过系统软件包提供，对其进行更改可能会破坏更新。因此，请在 `/etc/sudoers.d/` 目录下创建单独的配置文件用于保存自定义设置。默认情况下，这些设置将由 `/etc/sudoers` 中的某个指令拉入。

限制 **sudo** 超时

出于安全原因，请不要向 `root` 特权授予不受限的访问权限。相反，请设置合理的超时，以防任何入侵者滥用 `root` 帐户。有关详细信息，请参见第 3 节“更改 **sudo** 口令提示超时”。

使用 **visudo** 命令

使用 **visudo** 命令可以安全地编辑 `/etc/sudoers` 文件，因为它会在保存更改之前检查文件的语法。这是一种预防措施，旨在纠正任何可能破坏系统的错误。除了基本的语法检查之外，您还可以运行 **visudo -c** 来检查整个 **sudo** 配置框架是否以正确的顺序进行分析且不包含错误。

按组而不是按个人管理用户

尽可能使 **sudo** 配置保持精简且易于管理。应采用以下方式管理用户：将用户添加到组，然后向这些组而不是个人授予特权。这样可以方便地通过更改组设置来添加或删除用户，而不必在整个配置中查找用户。

允许示例 `%admingrp` 组中的所有用户执行所有命令的示例规则：

```
%admingrp ALL = (ALL) ALL
```

限制二进制文件的路径

使用 `secure_path` 指令限制用户可以在哪些区域执行命令。以下示例是 SLE Micro 随附的默认设置。

```
Defaults secure_path="/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/bin:/usr/local/sbin"
```

保持 `sudo` 日志记录透明

`sudo` 记录到标准日志文件，但其中的日志项很容易被忽略。将以下规则添加到您的配置，以指定专用的 `sudo` 日志文件。

```
Defaults logfile=/var/log/sudo.log
```

6 查错

了解如何对 `sudo` 配置问题进行调试和查错。

6.1 `/etc/sudoers.d/` 下的自定义配置被忽略

`/etc/sudoers` 中的 `#includedir` 指令会忽略以 `~` 字符结尾或包含 `.` 字符的文件。这是为了避免软件包管理器提供的配置文件（包含 `.`）或编辑器的临时文件或备份文件（以 `~` 结尾）出现问题。请确保自定义配置文件的名称既不包含这些字符，也不以这些字符结尾。如果存在这种情况，请重命名这些文件。

6.2 自定义指令冲突

配置文件的读取顺序决定了何时应用 `sudo` 配置指令。`/etc/sudoers.d/` 下的文件中的指令优先于 `/etc/sudoers` 中的相同指令。如果 `/etc/sudoers.d/` 中指定的自定义指令不起作用，请使用 `visudo -c` 检查文件的读取顺序。根据需要调整顺序。

6.3 由于 `sudo` 配置损坏而被锁定

如果您不小心破坏了系统的 `sudo` 配置，因而将自己锁定在 `sudo` 之外，可以使用 `su -` 和 `root` 口令来启动 `root` 外壳。运行 `visudo -c` 以检查错误，然后使用 `visudo` 修复错误。

7 `sudo` 配置参考

本节提供基本的 `sudo` 配置参考，以帮助您了解和维护默认和自定义的 `sudo` 配置。

7.1 `sudoers` 配置语法

`sudoers` 配置文件包含两种类型的选项：字符串和标志。字符串可以包含任何值，而标志则只能在“ON”或“OFF”之间切换。`sudoers` 配置文件的最重要语法构造如下：

```
# Everything on a line after # is ignored ❶
Defaults !insults # Disable the insults flag ❷
Defaults env_keep += "DISPLAY HOME" # Add DISPLAY and HOME to env_keep ❸
tux ALL = NOPASSWD: /usr/bin/frobnicate, PASSWD: /usr/bin/journalctl ❹
```

- ❶ 存在两个例外：`#include` 和 `#includedir` 是普通命令。较新版本不再使用 `#`。取而代之的是，`include` 指令现在前面带有 `@`。为了向后兼容，仍支持 `#` 表示法。
- ❷ 去除 `!` 字符会将所需标志设置为 ON。
- ❸ 指定在启用 `env_reset` 时应保留的环境变量列表。
- ❹ 一个复杂的规则，规定用户 `tux` 必须输入口令才能运行 `/usr/bin/journalctl`，但不需要输入口令即可在所有主机上运行 `/usr/bin/frobnicate`。

有用的标志和选项

`targetpw`

如果设置此项，则 `sudo` 会提示您输入在 `-u` 选项中指定的用户口令；如果未使用 `-u`，则提示您输入 `root` 口令。默认值为“ON”。

```
Defaults targetpw # Turn targetpw flag ON
```

rootpw

如果设置，**sudo** 将提示输入 root 口令。默认值为 OFF。

```
Defaults !rootpw # Turn rootpw flag OFF
```

env_reset

如果设置，**sudo** 会构造一个具有

TERM、PATH、HOME、MAIL、SHELL、LOGNAME、USER、USERNAME 和 SUDO_* 的极简环境。此外，会从调用环境导入 env_keep 中列出的变量。默认值为 “ON”。

```
Defaults env_reset # Turn env_reset flag ON
```

env_keep

当 env_reset 标志为 ON 时要保留的环境变量列表。

```
# Set env_keep to contain EDITOR and PROMPT
Defaults env_keep = "EDITOR PROMPT"
Defaults env_keep += "JRE_HOME" # Add JRE_HOME
Defaults env_keep -= "JRE_HOME" # Remove JRE_HOME
```

env_delete

当 env_reset 标志为 OFF 时要去除的环境变量列表。

```
# Set env_delete to contain EDITOR and PROMPT
Defaults env_delete = "EDITOR PROMPT"
Defaults env_delete += "JRE_HOME" # Add JRE_HOME
Defaults env_delete -= "JRE_HOME" # Remove JRE_HOME
```

7.2 基本 **sudoers** 规则

每条规则都遵循以下模式（[] 标记的是可选部分）：

#Who	Where	As whom	Tag	What
User_List	Host_List	= [(User_List)]	[NOPASSWD: PASSWD:]	Cmnd_List

SUDOERS 规则语法

User_List

一个或多个标识符（用逗号分隔）：用户名、%GROUPNAME 格式的组，或 #UID 格式的用户 ID。可以使用 ! 前缀指定求反。

Host_List

一个或多个标识符（用逗号分隔）：完全限定的主机名或 IP 地址。可以使用 ! 前缀指定求反。Host_List 的常用选项为 ALL。

NOPASSWD: | PASSWD:

如果用户在 NOPASSWD: 后面运行的命令与 Cmd_List 匹配，系统不会提示用户输入口令。

PASSWD: 为默认选项。仅当 PASSWD: 和 NOPASSWD: 位于同一行时，才需要指定此选项：

```
tux ALL = PASSWD: /usr/bin/foo, NOPASSWD: /usr/bin/bar
```

Cmd_List

一个或多个说明符（用逗号分隔）：可执行文件的路径，后接允许的可选参数。

```
/usr/bin/foo      # Anything allowed
/usr/bin/foo bar  # Only "/usr/bin/foo bar" allowed
/usr/bin/foo ""   # No arguments allowed
```

ALL 可以用作 User_List、Host_List 和 Cmd_List。

7.3 使用别名简化 sudoers

管理员可以通过在组项目中引入别名，来避免维护一组重复的规则或逐个地维护规则。别名的语法与规则的语法相同。支持以下类型的别名：

User_Alias

用户名列表

Runas_Alias

按 UID 划分的用户组

Host_Alias

主机名列表

Cmnd_Alias

命令、目录和别名的列表

将别名视为用户、组、命令和主机的命名列表。以下示例展示了别名的强大作用：

```
Host_Alias    WEBSERVERS = www1, www2, www3 ❶
User_Alias    ADMINS = tux, wilber, suzanne ❷
Cmnd_Alias    REBOOT = /sbin/halt, /sbin/reboot, /sbin/poweroff ❸
ADMINS WEBSERVERS = REBOOT ❹
```

- ❶ 这三个服务器已分组到一个 Host_Alias WEBSERVERS。您可以使用完全限定的主机名或 IP 地址。
- ❷ 与上面分组的主机类似，此处列出了组用户甚至用户组（例如 %wheel）。像往常一样，否定是通过 ! 前缀实现的。
- ❸ 指定在同一上下文中使用的一组命令。
- ❹ 所有别名包含在一条规则中，该规则规定，User_Alias 指定的所有用户都可以在 Host_Alias 中命名的所有主机上执行 Cmnd_Alias 下指定的一组命令。

概括而言，别名可帮助管理员保持 `sudoers` 精简且易管理（因此保持其安全性）。例如，如果某个用户离职，您只需从 User_Alias 语句以及此人所属的任何系统组中删除此人的姓名一次，而不必搜索包含此特定用户的所有规则。

8 法律声明

版权所有 © 2006–2025 SUSE LLC 和贡献者。保留所有权利。

根据 GNU 自由文档许可证 (GNU Free Documentation License) 版本 1.2 或（根据您的选择）版本 1.3 中的条款，在此授予您复制、分发和/或修改本文档的权限；本版权声明和许可证附带不可变部分。许可版本 1.2 的副本包含在题为“GNU Free Documentation License”的部分。

有关 SUSE 商标，请参见 <https://www.suse.com/company/legal/> 。所有其他第三方商标分别为相应所有者的财产。商标符号（®、™ 等）代表 SUSE 及其关联公司的商标。星号 (*) 代表第三方商标。

本指南力求涵盖所有细节，但这不能确保本指南准确无误。SUSE LLC 及其关联公司、作者和译者对于可能出现的错误或由此造成的后果皆不承担责任。

A GNU 自由文档许可证

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. 允许任何人复制和分发此许可证文档的逐字副本，但禁止对其进行更改。

0. 导言

此许可证的目的是赋予手册、教科书或其他功能性的和有用的文档以“自由”：即保证每个人都有复制和再分发此类文档的有效自由，无论是否进行修改，也无论将其用于商业或非商业用途。其次，此许可证为作者和出版者保留了因工作获得声誉但不视为对他人所做修改负责的方式。

本许可证是一种“非盈利版权”，这意味着从该文档衍生的作品也必须是以同一方式自由的。它补充了 GNU 通用公共许可证（为自由软件设计的非盈利版权许可证）。

我们设计此许可证旨在将其用于免费软件的手册，因为免费软件需要自由文档：免费程序所附手册应具有与软件本身同样的自由。但是此许可证不限于软件手册；它可以用于任何文本作品，无论主题如何或它是否作为印刷书籍出版。建议本许可证主要用于目的是指导或参考的作品。

1. 适用性和定义

此许可证适用的对象：由版权所有者在其中明确声明可按照此许可证条款以任何媒体分发的任何手册和其他作品。此类声明授予在此处所述的条款和条件下使用该作品的全球无限期无版权许可证。下述“文档”指任何此类手册或作品。任何公众成员都是一个被许可人，以下称为“您”。如果您以需要版权法许可的任何方式复制、修改或分发该作品，则表示您接受该许可证。

该文档的“修改版本”表示包含该文档或其一部分（或者逐字复制或者有修改和/或翻译为另一语言）的任何作品。

“次要章节”是该文档的命名附录或扉页章节，专门讲述该文档的出版者或作者与该文档整个主题（或相关问题）的关系，不包含与整个主题相关的内容。（因此，如果该文档是数学课本的一部分，则辅助部分可能不说明任何数学问题。）这种关系可以是与主题或相关问题的历史联系，或与它们相关的法律、商业、哲学、伦理或政治地位。

在该文档基于此许可证项发布的声明中，“固定章节”是将其标题指定为固定章节标题的一些辅助章节。如果一个章节不适用上述辅助章节的定义，则不允许将其指定为固定章节。该文档可能不包含固定章节。如果该文档不标识任何固定章节，则表示没有固定章节。

在该文档基于此许可证项发布的声明中，“封页文本”是作为封面文本或封底文本列出的简短文本段落。封面文本最多 5 个单词，封底文本最多 25 个单词。

文档的“透明”副本是一个机器可读的副本，使用公众可以得到其规范的格式表达，这样的副本适合于使用通用文本编辑器、（对于像素构成的图像）通用绘图程序、（对于绘制的图形）广泛使用的绘画编辑器直接修改文档，也适用于输入到文本格式处理程序或自动翻译成各种适用于输入到文本格式处理程序的格式。一个用其他透明文件格式表示的副本，如果该格式的标记（或缺少标记）已经构成了对读者的后续修改的障碍，那么就是不透明的。表示实质性数量的文本的图像格式都是不透明的。不“透明”的副本称为“不透明”。

适于作为透明副本的格式的示例有：没有标记的纯 ASCII 文本、Texinfo 输入格式、LaTeX 输入格式、使用公共可用 DTD 的 SGML 或 XML，符合标准的简单 HTML、可以人为修改的 PostScript 或 PDF。透明图像格式的示例有 PNG、XCF 和 JPG。不透明的格式包括：仅可以被私有版权的字处理软件使用的私有版权格式、所用的 DTD 和/或处理工具不是广泛可用的 SGML 或 XML、机器生成的 HTML、一些字处理器生成的只用于输出目的的 PostScript 或 PDF。

对于印刷书籍，“扉页”就是扉页本身以及随后的一些用于补充的页，显然本许可资料需要出现在扉页上。对于那些没有扉页的作品形式，“扉页”代表接近作品最突出标题的、在文本正文之前的文本。

“命名为 XYZ”的章节表示文档的一个特定的子单元，其标题就是 XYZ 或在括号中包含 XYZ 且后跟 XYZ 的其他语言翻译文本。（这里 XYZ 代表下面提及的特定章节名称，比如“致谢”、“题献”、“签名”或“历史”。）要在修改文档时这类章节“保留标题”就是依据此定义保持这样一个“命名为 XYZ”的章节。

文档可能在文档遵照此许可证的声明后面包含免责声明。这些免责声明应作为参考信息包含在此许可证中，但是只能将其视作免责声明：这些免责声明暗指的任何其他含义均无效，且对此许可证的含义不产生任何影响。

2. 逐字复制

您可以用任何媒体复制并分发文档，无论是出于商业还是非商业目的，只要保证此许可证、版权声明和声称此许可证应用于文档的声明都完整地、无任何附加条件地存在于所有副本中。不能使用任何技术手段阻碍或控制您制作或发布的副本的阅读或再次复制。不过您可以在副本交易中得到报酬。如果发布足够多的副本，则您必须遵循下面第三节中的条件。

您也可以在这样的条件下出租副本和向公众放映副本。

3. 大量复制

如果您出版的文档印刷版副本（或是有印制封页的其他媒体副本）多于 100 份，而文档的许可证声明中要求有封页文本，则您必须将它清晰地置于封页之上，封面文本在封面上，封底文本在封底上。封面和封底上还必须标明您是这些副本的出版者。封面必须同等显著地完整展现标题的所有文字。您可以在封页上加入其他资料。改动仅限于封页的复制，只要保持文档的标题不变并满足这些条件，可以在其他方面被视为逐字复制。

如果需要加上的文本对于封面或封底过多，无法明显地表示，您应该在封页上列出前面的（在合理的前提下尽量多），把其他的放在邻近的页面上。

如果您出版或分发了超过 100 份文档的不透明副本，则必须在每个不透明副本中包含一份计算机可读的透明副本，或是在每个不透明副本中给出一个计算机网络地址，通过这个地址，网络公共用户可以使用标准网络协议下载文档的无任何附加资料的完整透明副本。如果您选择后者，则必须在开始大量分发非透明副本的时候采用相当谨慎的步骤，保证透明副本在其所给出的位置在（直接或通过代理和零售商）分发最后一次该版本的非透明副本的时间之后一年之内始终是有效的。

在重新大量发布副本之前，请您（但不是必须）与文档的作者联系，以便他们可以有机会向您提供文档的更新版本。

4. 修改

在上述第 2、3 节的条件下，您可以复制和分发文档的修改版本，前提是严格按照此许可证发布修改后的文档，将修改版本用作文档，从而允许任何拥有此修改版副本的人执行分发或修改。

另外，在修改版中，您需要做到如下几点：

- A.** 用于与文档以及以前各个版本（如果有，应该列在文档的“历史”章节中）显著不同的扉页（和封页，如果有）。如果那个版本的原始发行者允许的话，您可以使用和以前版本相同的标题。
- B.** 与作者一样，在扉页上列出承担修改版本中的修改的作者责任的一个或多个人或实体和至少五个文档的原作者（如果原作者不足五个就全部列出），除非他们免除了您的这个责任。
- C.** 与原来的发行者一样，在扉页上列出修改版的发行者的姓名。
- D.** 保持该文档的全部版权声明不变。
- E.** 在与其他版权声明邻近的位置加入恰当的针对您的修改的版权声明。
- F.** 在紧接着版权声明的位置加入许可声明，按照下面附录中给出的形式，以本许可证给公众授于是用修订版本的权利。
- G.** 保持原文档的许可声明中的全部不可变章节、封面文字和封底文字的声明不变。
- H.** 包含一份未作任何修改的本协议的副本。
 - I.** 保持命名为“历史”的章节不变，保持它的标题不变，并在其中加入一项，至少声明扉页上的已修改版本的标题、年份、新作者和出版者。如果文档中没有命名为“历史”章节，则请新建它，并加入一项以声明原文档扉页上所列的标题、年份、作者与出版者，再在其后加入如上所说的描述修改版本的项。
 - J.** 如果文档中有用于公共用户访问的文档透明副本的网址，则保持网址不变，并同样提供它所基于的以前文档版本的网址。这些网址可以放在“历史”章节。您可以不给出那些在原文档发行之前已经发行至少四年的版本给出的网址，或者该版本的发行者授权不列出网址。
 - K.** 对于任何命名为“致谢”或“题献”的章节，保持其标题不变，并保持其全部内容以及对每位贡献者致谢和/或题献的语气不变。
 - L.** 保持文档的所有固定章节不变，不改变它们的标题和内容。章节的编号或相当的内容不被认为是章节标题的一部分。
 - M.** 删除命名为“签名”的章节。这样的章节不可以被包含在修改后的版本中。
 - N.** 不要把任何现有章节重命名为“签名”或与任何不可变章节相冲突的标题。
 - O.** 保持任何免责声明不变。

如果修改版本加入了新的符合次要章节定义的引言或附录章节，并且不含有从原文档中复制的内容，您可以选择将其标记为固定。如果需要这样做，则将它们标题加入修改版本许可声明的不可变章节列表之中。这些标题必须和其他章节的标题相区分。

您可以加入一个命名为“签名”的章节，只要它只包含对您的修改版本由不同的各方给出的签名，例如书评或是声明文本已经被一个组织认定为一个标准的权威定义。

您可以加入一个最多 5 个字的段落作为封面文本和一个最多 25 个字的段落作为封底文本，将它们加入修改版本的封页文本列表末端。一个实体只可以添加（或编排）一段封面和一段封底文本。如果原文档已经为该封页（封面或封底）包含了封页文本，由您或您所代表的实体先加入或排列的文本，不能再新加入一个，但您可以在原来的发行者的明确许可下替换掉原来的那个。

作者和发行者不能通过本许可证授权公众使用他们的名字推荐或暗示认可任何一个修改版本。

5. 组合文档

遵照第 4 节所说的修改版本的规定，您以将文档和其他文档合并并以本许可证发布，只要您在合并结果中包含原文档的所有不可变章节，对它们不加以任何改动，并在合并结果的许可声明中将它们全部列为不可变章节，而且维持原作者的免责声明不变。

合并作品仅需要包含一份此许可证，多个相同的固定章节可以由一个副本取代。如果有多个名称相同但内容不同的固定章节，通过在章节名称后面的括号中加上原作者或出版者的姓名（如果已知）来加以区别，或者使用唯一编号加以区别。并对合并作品许可声明中的固定章节列表中的章节标题做相同的调整。

在合并过程中，必须合并不同原始文档中任何命名为“历史”的章节，从而形成新的命名为“历史”的章节；类似地，还要合并命名为“致谢”和“题献”的章节。必须删除所有命名为“签名”的章节。

6. 文档的合集

您可以制作一个文档和其他文档的合集，在本许可证下发布，并在合集中将不同文档中的多个本许可证的副本以一个单独的副本来代替，只要您在文档的其他方面遵循本许可证的逐字复制的条款即可。

您可以从一个这样的合集中提取一个单独的文档，并将它在本许可证下单独发布，只要您想这个提取出的文档中加入一份本许可证的副本，并在文档的其他方面遵循本许可证的逐字复制的原则。

7. 独立作品的聚合体

将文档或其派生品以及其他独立和无关文档或作品编撰在一个储存卷中或分发媒体上，这称为文档的“聚合体”，前提是编撰成品的著作权对其使用者的法律权限的限制未超出各个独立作品的许可范围。当基于此许可证发布的文档包含在一个聚合体中时，此许可证不适用于聚合体中的本非该文档派生作品的其他作品。

如果第 3 节中的封页文本要求适用于这些文档的副本，则若文档在聚合体中所占的比重小于全作品的一半，文档的封页文本可以放置在聚合体内包含文档部分的封页上，或是电子文档中的等效部分。否则，它必须位于整个聚合体的印刷的封页上。

8. 翻译

翻译被视为一种修改，因此您可以根据第 4 节的条款分发文档的翻译。将固定章节替换为翻译内容需要经得其版权所有者的特别许可，但除了这些固定章节的原始版本之外，您还可以包含一部分或所有固定章节的翻译。您可以包含一个此许可证以及所有许可证声明和免责声明的翻译版本，前提是同时包含它们的原始英文版本。当翻译版本和英文版发生冲突的时候，原始版本有效。

如果在文档中有命名为“致谢”、“题献”或“历史”的章节，保持标题（第 1 节）的要求（第 4 节）恰恰需要更换实际的标题。

9. 终止

除非此许可证中有明确规定，否则您不能对该文档进行复制、修改、分授许可或分发。在此许可证规定外对该文档所进行的任何复制、修改、分授许可或分发都是无效的，并且将自动终止您在此许可证下所拥有的权利。但是，对于在此许可证的规定下从您这里获得副本或权利的各方，只要其完全遵守此许可证的规定，其许可证将不会被终止。

10. 本许可的未来修订版本

自由软件基金会有时会发布 GNU 自由文档许可证的新的修订版版本。这些新版本的主旨和精神与当前版本是一致的，但在解决新问题的具体细节方面可能有所不同。请参见 <https://www.gnu.org/copyleft/>。

许可证的每个版本都有一个不同的版本号。如果文档指定了适用于它的此许可证“或任何后续版本”的特定带编号版本，则您可以选择遵从指定版本或自由软件基金会发布的任何随后版本（非草稿）的条款和条件。如果文档没有指定此许可证的版本号，您可以选择自由软件基金会发布的任何许可证版本（非草稿）。

附录：如何针对您的文档使用此许可证

```
Copyright (c) YEAR YOUR NAME.  
Permission is granted to copy, distribute and/or modify this document  
under the terms of the GNU Free Documentation License, Version 1.2  
or any later version published by the Free Software Foundation;  
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.  
A copy of the license is included in the section entitled "GNU  
Free Documentation License".
```

如果您有固定章节、封面文本和封底文本，请将“with...Texts”部分替换为：

```
with the Invariant Sections being LIST THEIR TITLES, with the  
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

如果有不可变章节而没有封页文本，或这三种内容（不可变章节、封面文本、封底文本）的任何其他组合，请合并这两个备选项以适应您的情况。

如果您的文档包含不一般的程序代码示例，建议同时选择自由软件许可证（如 GNU 通用公共许可证）发布这些示例，以允许它们可以用于自由软件。