Minipaca II 用户手册

Xu Xin

sxuxin@protonmail.com

目　录

1 介绍

Minipaca 是一个用于软件开发的工具。它使用 Chez Scheme 做为编程语言，GTK4 做为图形介面库来开发软件。

Minipaca 内置了 Chez Scheme 和 GTK4 之间的接口绑定，所以无论是开发命令行软件还是图形介面软件，都是使用 Chez Scheme 这一种编程语言，不需要使用 C 语言。

Minipaca 先将源文件编译成一个可执行文件，再将可执行文件和其它文件打包成各发行版的安装包。目前，Minipaca 支持 Linux、BSD 和 Windows 三大平台下的众多发行版。

Minipaca 就是使用自身技术开发的。

Minipaca:	`https://minipaca.github.io`
Chez Scheme:	`https://github.com/cisco/ChezScheme`
GTK4:	`https://docs.gtk.org/gtk4`

2 安装

- AMD64 平台

AMD64	发行版	简写	安装包格式
Linux	Arch Linux	arch	pkg.tar.zst
Linux	EndeavourOS	endeavour	pkg.tar.zst
Linux	Debian Testing	debiant	deb
Linux	Debian 12	debian12	deb
Linux	Ubuntu 21.10	ub2110	deb
Linux	Ubuntu 22.04	ub2204	deb
Linux	Ubuntu 23.04	ub2304	deb
Linux	OpenSUSE Tumbleweed	opensusetw	rpm
Linux	Fedora Linux 36	fc36	rpm
Linux	Fedora Linux 37	fc37	rpm
Linux	Fedora Linux 38	fc38	rpm
Linux	Void Linux	void	xbps
BSD	FreeBSD 13	fb13	pkg
Windows	Windows 10	win10	exe
Windows	Windows 11	win11	exe

- AArch64 平台

AArch64	发行版	简写	安装包格式
Linux	Arch Linux for AArch64	arch-aarch64	pkg.tar.zst

请到 Minipaca 网站选择最接近你的系统的安装包下载，并按下面的说明来安装：

- Arch Linux，EndeavourOS：

```
yay -S chez-scheme
sudo pacman -U ./xx.pkg.tar.zst
```

- Arch Linux (aarch64)：

```
yay -S chez-scheme-racket-git
sudo pacman -U ./xx.pkg.tar.zst
```

- Ubuntu 22.04+，Debian 12+，Debian Testing：

```
sudo apt install ./xx.deb
```

- openSUSE Tumbleweed：

```
sudo zypper addreop https://download.opensuse.org/\
repositories/devel:languages:misc/\
openSUSE_Tumbleweed/devel:languages:misc.repo
sudo zypper refresh
sudo zypper install ./xx.rpm
```

- Fedora 36+：

```
sudo dnf copr enable superboum/chez-scheme
sudo dnf install ./xx.rmp
```

- Void Linux：

```
xbps-rindex -a xx.x86_64.xbps
sudo xbps-install -R ./ xx
```

- FreeBSD 13+：

```
pkg add -f xx.pkg
```

- Windows 10+：

双击打开安装

3 初次使用

- 打开一个终端软件，输入 minipaca，按下 Enter 键：

```
$ minipaca

    用ChezScheme和GTK4构建跨平台软件

使用方法:

    minipaca -x [命令]
    minipaca <快捷方式>

可用的[命令]有:

    app         新建一个命令行软件
    app-gtk4    新建一个图形界面软件

    generate    生成可执行文件
    pack        为发行版打包

    version     显示版本号
    usage       查看用法

    functree    显示函数调用树
    translate   自动翻译

    user        显示用户信息
    reference   查看编程参考

可用的<快捷方式>有:

    -g          '-x generate'的快捷方式
    -p          '-x pack'的快捷方式
```

4 开发你的第一个命令行软件

为了方便，推荐在一个专门的文件夹中来开发软件，比如说 dev 文件夹。

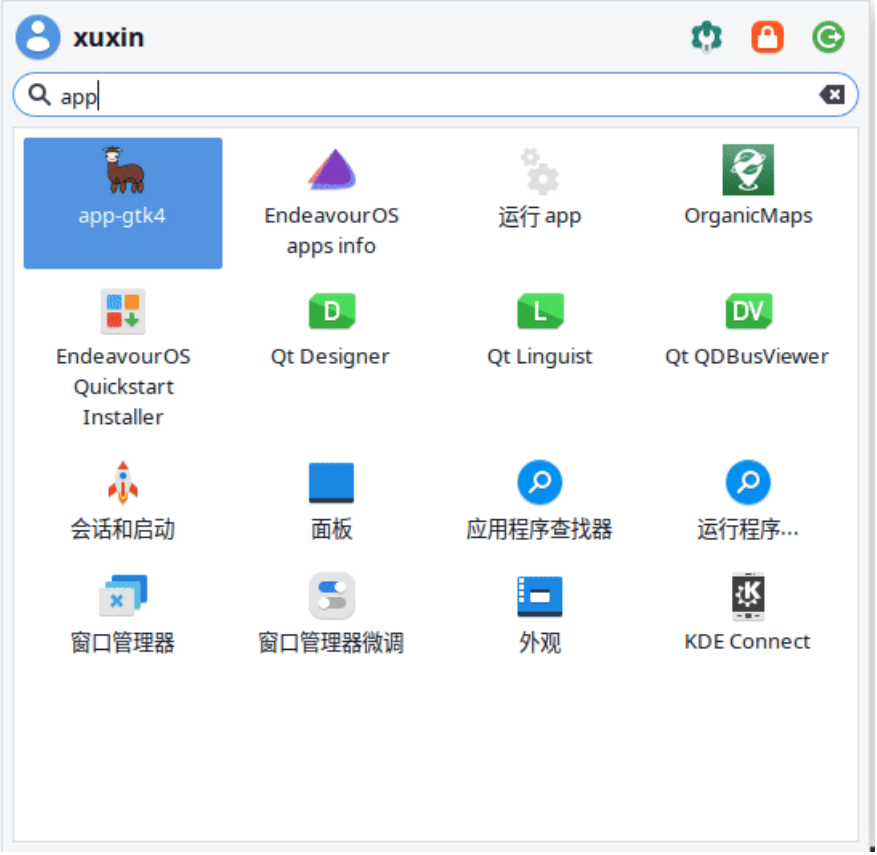
- 打开一个终端软件，输入 mkdir dev，按下 Enter 键，来新建 dev 文件夹；
- 输入 cd dev，按下 Enter 键，来进入 dev 文件夹；
- 输入 minipaca -x app，按下 Enter 键，来新建软件 app；
- 输入 cd app，按下 Enter 键，来进入 app 文件夹；
- 输入 minipaca -g，按下 Enter 键，来编译 app；
- 输入 minipaca -p，按下 Enter 键，来打包 app；
- 打开文件管理器，导航到 ~ → dev → app → p，来查看打包后的安装包；
- 根据前面提到的和 minipaca 相同的安装方式来安装 app 即可；
- 输入 app，按下 Enter 键，来运行 app；

恭喜你，你已经成功开发了一个命令行软件。

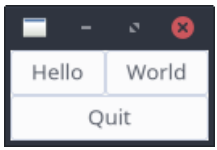
5 开发你的第一个图形界面软件

由于前面你学会了开发命令行软件，开发图形界面软件的步骤与此相同。有两点差别，一是请把 app 换成 app-gtk4，二是图形界面软件的启动方式要通过程序启动器来启动。

- 打开程序启动器，找到 app-gtk4：



- 点击打开 app-gtk4：



恭喜你，你又成功开发了一个图形界面软件。

6 学习 Chez Scheme

- 学习 Scheme 编程可以参考 SICP:

计算机程序的构造和解释

- 学习 Chez Scheme 编程接口可以参考官方文档:

The Scheme Programming Language (4th Edition)
The Chez Scheme User's Guide for Version 9.5.8

- 快速查询函数用法可以参考函数列表:

函数列表

7 学习 Minipaca 内置函数

7.1 *f-e*

此函数用于存放全局变量。

- (**f-e** *pair*) → 返回值未定

把 *pair* 存入 *f-e* 。

- (**f-e** *key*) → 返回一个 pair 或者 #f

从 *f-e* 获取 *pair* 。如果未发现，返回 #f 。

7.2 *f-t*

翻译字符串。

- (**f-t** *lang*) → 返回值未定

设定当前语言为 *lang* 。*lang* 必需为一个 symbol。

- (**f-t** *pair*) → 返回值未定

把翻译放入 *f-t* 。*pair* 中的 *key* 为翻译前字符串，*value* 为翻译后字符串。

- (**f-t** *string*) → 返回一个 pair

从 *f-t* 获取翻译。

7.3 *f-copy-file*

- (**f-copy-file** *a b*) → 返回值未定

复制文件，从 *a* 到 *b* 。

7.4 *f-cpdir-rec*

- (**f-cpdir-rec** *m n*) → 返回值未定

递归地把文件夹 *m* 复制到文件夹 *n* 里面。

7.5 *f-mkdir-rec*

- (**f-mkdir-rec** m) → 返回值未定

递归地创建文件夹 m 。

7.6 *f-rmdir-rec*

- (**f-rmdir-rec** m) → 返回值未定

递归地删除文件夹 m 。

7.7 *f-disk-usage*

- (**f-disk-usage** path) → 返回一个 number

计算一个文件或文件夹的大小，单位为 Byte 。

7.8 *f-which-executable-file*

- (**f-which-executable** str) → 返回一个 path 或者 #f

一个命令的全路径。如果未发现，返回 #f 。

7.9 *f-string-split*

- (**f-string-split** *string delim*) → 返回一个 list

用 *delim* 把字符串分割开。*delim* 是一个字符。

7.10 *f-datum-file-read*

- (**f-datum-file-read** *file*) → 返回一个 alist

从文件 *file* 中读取 alist 。

7.11 *f-datum-file-write*

- (**f-datum-file-write** *alist file*) → 返回值未定

把 *alist* 写入文件 *file* 。

8 扩展你的第一个命令行软件

- 了解软件的源代码结构

您已经熟悉了 Minipaca 的基本使用方法，也已经熟悉了 Chez Scheme 编程语言，那么您可以试着扩展您的第一个命令行软件了。在真正开始之前，您有必要了解下默认的 app 软件的源代码结构。

```
$ minipaca -x app
$ ls app/
distro doc source
```

distro → 存放发行版信息的文件夹
doc → 存放文档的文件夹
source → 存放源代码的文件夹

```
$ ls app/distro
all.datum.ss            debiant.datum.ss    fc37.datum.ss
fc38.datum.ss           ub2304.datum.ss     arch.datum.ss
debian12.datum.ss       fc36.datum.ss       ub2204.datum.ss
...省略一部分
```

all.datum.ss → 软件名称、版本号等信息
其它.datum.ss → 与发行版相关的信息、软件依赖等

```
$ ls app/doc
log
```

log → 记录软件更改的文件

```
$ ls app/source
f i x ss.ss
```

ss.ss → 主脚本源文件
x → 存放 x- 开头的函数的文件夹
i → 存放 i- 开头的函数的文件夹
f → 存放 f- 开头的函数的文件夹

在编译前，软件会依次把 f 函数、i 函数、x 函数复制到 ss.ss 文件的前面。所以，各种函数的的层级从高到低为：

ss.ss	最上层
x	次上层
i	中间层
f	次下层
Chez Scheme	最下层

强制的规则为：① 函数仅可以调用比它低层次的函数；② 一个文件仅定义一个函数；③ x 函数接受任意数量的参数，并且返回值类型一定为字符串，如：

```
(define x-name
  (lambda parameters
    ...
    "abcdef"))
```

建议的规则为：为了后续的升级，不要增加或修改 f 函数，您的重点应放在编写 x 和 i 函数上。

- 更改软件名和版本号

打开 all.datum.ss 文件，更改相应的项，重新编译和打包即可。

- 为软件增加一个功能

比如说您想要读取一个文本文件并显示出来：

新建文件 **source/x/x-cat.ss**:

```
(define x-cat
  (lambda parameters
    (let* ((file (car parameters))
           (port (open-input-file file))
           (str (get-string-all port)))
      (close-port port)
      str)))
```

修改文件 **source/ss.ss**:

```
...
  (if (and (>= (length (cdr (command-line))) 2)
           (string=? (cadr (command-line)) "-x"))
      (let ((s (caddr (command-line))))
        (case s
         ("version" (set! x "x-version"))
         ("usage" (set! x "x-usage"))
         ("cat" (set! x "x-cat"))       ;; 增加这一行
         (else (set! x "x-usage")))
        (set! parameters (cdddr (command-line)))))
...
```

重新编译后试一下效果:

```
$ minipaca -g
$ g/XX/usr/bin/app -x cat ./text.txt
## XX为你的发行版
```

多次修改后，达到了您的目的后，您可以编译、打包、安装后再试下效果。

更多的例子您可以在 https://github.com/minipaca 找到。

9 学习 GTK4

学习 GTK4 主要是浏览官方文档：GTK Documentation。最重要的部分是 GTK、GDK 和 GSK。

10 扩展你的第一个图形界面软件

观察下 app 和 app-gtk4 的源代码，我们发现多了两个函数文件：

f-gtk4.ss	→	Chez Scheme 对 GTK4 的绑定
x-gtk4.ss	→	定义图形界面的代码

我们不要修改 f-gtk4.ss 文件，仅修改 x-gtk4.ss 即可。通过前面的技巧，不断修改、试效果，直到满足您的要求为止。

11 移植到其它平台的发行版

使用 Minipaca 开发的软件本身就是跨平台的。只需在想支持的平台中编译和打包您的软件即可。