

# ArchLinux 编程环境

西安理工大学 Linux 小组指导教师 张 翔

<https://www.facebook.com/xautofzx/>

本教程主要面向计算机类专业大三及以上的大学生，最好学过《操作系统》课程。

读者应该有 Ubuntu/Fedora 等流行的 Linux 发行版的安装配置使用经验，或在教师指导下使用本教程。

## 1. 启动并测试网络

打开 VMware Workstation，启动虚拟机中已经安装好的最精简 ArchLinux，输入用户名 root 和密码 xaut，进入 Shell 命令行状态。

在命令行输入 `pacman -Sy reflector`，安装测试镜像服务器速率的工具。

输入：`reflector --verbose --country China --sort rate --save /etc/pacman.d/mirrorlist`  
找到最快的中国区镜像服务器，加快后续软件包的下载速度。

## 2. 安装编程开发环境

在命令行输入：`pacman -S base-devel`，在 (default=all) 处直接回车，就会安装 26 中最常用的编译软件。该软件包的作用与 Debian 和 Ubuntu 中的 build-essential 包一样。

```
[root@xautLinux ~]# pacman -S base-devel
:: There are 26 members in group base-devel:
:: Repository core
  1) autoconf  2) automake  3) binutils  4) bison  5) fakeroot  6) file  7) findutils  8) flex
  9) gawk  10) gcc  11) gettext  12) grep  13) groff  14) gzip  15) libtool  16) m4  17) make
  18) pacman  19) patch  20) pkgconf  21) sed  22) sudo  23) systemd  24) texinfo  25) util-linux
  26) which
Enter a selection (default=all):
```

至此，就能用 Vi、GCC、make 等工具进行 C/C++ 的编程开发，也能进行 Linux Kernel 的内核编译了。

为了能够调试程序，使用命令：`pacman -S gdb`，安装调试工具包。

使用命令：`gcc --version`，`gdb --version`，`make --version`，来检查是否正确安装了。

参考文档：<https://wiki.archlinux.org/index.php/GNU#Toolchain>

## 3. 使用 GCC 编译 C 语言程序

用 vi 编写一段 C 语言程序，在命令行执行：`vi ctest.c`，输入下列程序，保存并退出：

```
#include <stdio.h>
int main () {
    printf ( "Hello C\n" );
    return 0;
}
```

在命令行执行：`gcc ctest.c -o ctest`，参数 -o，必须是小写字母，表示 output，输出的文件名。

在命令行执行：`./ctest`，小数点表示当前目录。Linux 默认只执行 PATH 环境变量中的可执行文件，可用命令：`echo $PATH`，查看默认的搜索路径。要想执行其他位置的文件，就必须写明路径，相对路径或绝对路径均可。

提示：`gcc` 会自动给生成的文件，授予可执行的权限。如果不能执行，就用命令：`chmod +x ctest`。

GCC 支持的语言标准：<https://gcc.gnu.org/onlinedocs/gcc/Standards.html>

## 4. 使用 GCC 编译 C++ 程序

用 vi 编写一段 C++ 程序，在命令行执行：`vi cpptest.cpp`，输入下列程序，保存并退出。

```
#include <iostream>
using namespace std;
int main () {
    cout << "Hello Cpp" << endl;
    return 0;
}
```

在命令行执行: `g++ cpptest.cpp -o cpptest` 。

在命令行执行: `./cpptest` 。

g++除了能编译C++程序以外, 也作为 linker 链接器, 将 C / C++ 的目标文件装配到一起。

## 5. 使用 GCC 编译 Fortran / Go 等

GCC 还支持 Ada、Fortran、Go、Objective-C 等语言。

参考文档: [https://wiki.archlinux.org/index.php/Programming\\_languages](https://wiki.archlinux.org/index.php/Programming_languages)

GCC 还支持内联 ASM 汇编语言, 在 binutils 软件包中, 就有 as 汇编器和 ld 链接器。在 Unix 和 Linux 系统中, 采用的还是 AT&T 格式。参考文档: <https://www.ibm.com/developerworks/cn/linux/1-assembly/>

如果要编译 DOS/Windows 下的 Intel 格式, 需要安装 NASM 软件包, 或者安装 dosbox 虚拟机, 在 DOS 中使用 MASM 软件。参考文档: <https://blog.csdn.net/odaynot/article/details/7952201>

## 6. 使用 make 自动编译项目

GNU make 是自动化构建软件, 使用 makefile 文件来确定一个 target 文件的依赖关系, 然后把生成这个 target 的相关命令传给 shell 去执行。根据依赖文件的修改时间进行判断, 决定是否要重新生成该文件, 避免重复编译。

以 cpptest.cpp 为例, 编写一个 makefile 文件, 在命令行执行: `vi makefile`, 输入下列代码保存并退出。一定要注意: 缩进必须使用 Tab 键, 不能用空格。

```
OBJS = ctest.o
CC = gcc
CFLAGS = -Wall -O -g

maketest : $(OBJS)
    $(CC) $(OBJS) -o maketest
ctest.o : ctest.c
    $(CC) $(CFLAGS) -c ctest.c
clean :
    rm -rf *.o maketest
```

在命令行执行: `make clean`, 删除以前编译生成的旧文件。通常只在对项目全新编译时才使用。

在命令行执行: `make`, 进行自动化编译。

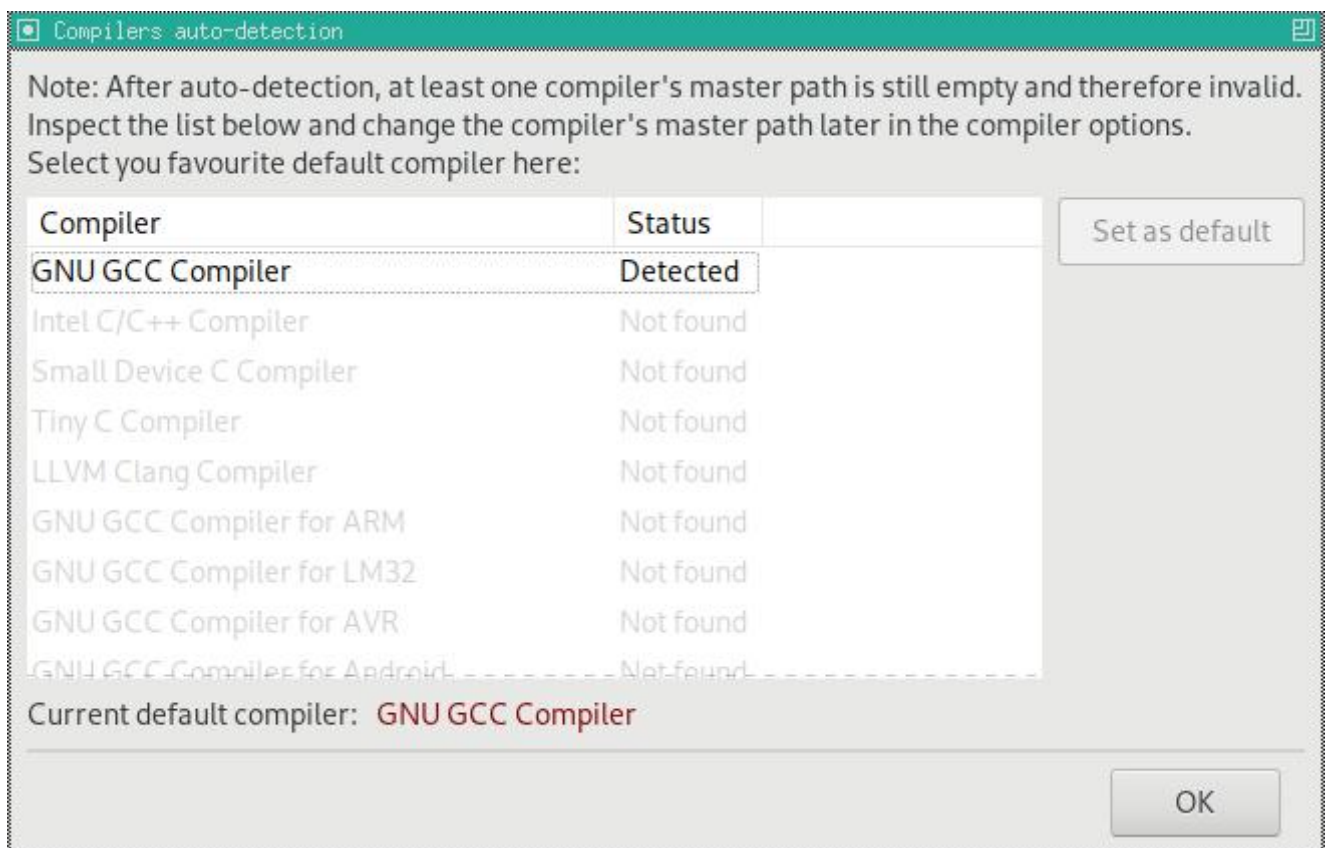
在命令行执行: `./maketest`, 可以运行生成的可执行文件。

## 7. 安装 IDE 集成开发环境

Code::Blocks 是很好的 IDE 集成开发环境, 支持 C / C++ 和 Fortran 语言, 会自动集成 GCC 和 GDB 工具。

输入命令: `pacman -S codeblocks` 即可安装。需要 startx 启动图形界面后才能运行。

在命令行输入: `codeblocks &`, 第一次启动运行后, 会检测到 GNU GCC, 要设置为 default 编译器。参考文档: [https://wiki.archlinux.org/index.php/List\\_of\\_applications/Utilities#Integrated\\_development\\_environments](https://wiki.archlinux.org/index.php/List_of_applications/Utilities#Integrated_development_environments)



因为会启动多个窗体，所以当鼠标出现虚框时，就移动到屏幕左上角，然后点击左键，新的窗体就能显示出来。

## 8. 安装文件传输工具 FileZilla

在命令行输入 `pacman -S filezilla`，该软件支持 FTP/SFTP 的文件传送，可以把文件备份到服务器。

```
[root@xautLinux ~]# pacman -S filezilla
resolving dependencies...
looking for conflicting packages...

Packages (5) libfilezilla-0.15.0-1 libidn-1.35-1 xdg-utils-1.1.3-3 xorg-xset-1.2.4-1
filezilla-3.38.1-1

Total Download Size: 4.34 MiB
Total Installed Size: 17.91 MiB

:: Proceed with installation? [Y/n] y_
```

首先在命令行中输入 `startx`，启动 X.org 图形界面，然后才能运行其他带有 GUI 图形界面的应用程序。

同样的，在终端窗口中输入 `filezilla &`，等待几秒钟即可看到一个虚框，移动鼠标将虚框移动到屏幕正中，点击鼠标就能看到 FileZilla 的窗口。

如果服务器端是 Windows，推荐安装 FileZilla Server，免费且简单好用。

<https://filezilla-project.org/download.php?type=server>

如果服务器端是 Linux，推荐安装 SSH，然后通过 SFTP 协议传输文件。

## 9. 配置 UTF-8 语言编码

在命令行输入：`vi /etc/locale.gen`，就会看到 Linux 所支持的语言列表。

输入 176，再按 Shift+G 跳转，找到 “`# en_US.UTF-8 UTF-8`”，按 x 键删除第一列的 # 号。

输入 498，再按 Shift+G 跳转，找到 “`# zh_CN.UTF-8 UTF-8`”，按 x 键删除第一列的 # 号。

然后 Shift+冒号，输入 wq 再回车，即可存盘退出。

在命令行输入：`locale-gen`，等十几秒后，就生成本地语言信息。

在命令行输入：`locale -a`，就能看到系统所支持的语言。

在命令行输入：`echo LANG="en_US.UTF-8" > /etc/locale.conf`

然后输入：**exit**，退出命令行，再重新登录，执行：**locale**，就能看到当前的语言环境变量信息了。  
参考文档：<https://wiki.archlinux.org/index.php/Locale>

提示：当前系统并没有安装 Gnome/KDE/LXDE 等桌面环境，所以并没有将全局配置文件/etc/locale.conf 设置为中文。因为这会造成 TTY 命令行下，汉字会呈现方块状的乱码。其次，UTF-8 编码也会造成 twm 窗口管理器无法启动。只能在命令行输入：**export LANG=C**，才能临时切换到英文 ANSI 编码状态。

推荐做法是，把切换语句存入脚本文件中：

```
echo LANG="zh_CN.UTF-8" > ~/chinese.sh
```

只在需要中文环境的 Shell 窗口中，运行：**source chinese.sh**，就能切换到中文。

## 10. 在 Xterm 中显示汉字

图形终端窗口 xterm 历史悠久、用途广泛。但是在中文环境下却无法正常使用汉字，网上有改变其全局配置的方法，但不一定能起作用，请自行尝试：**vi /usr/share/X11/app-defaults/XTerm**

手工处理的方法如下：

首先用 startx 启动图形界面，运行：**source ~/chinese.sh**，切换到中文环境。然后执行：**xterm**，就会打开新的命令行窗口。在执行命令：**df -h**，可以看到中文是方块的乱码。

此时按住 Ctrl 键，再按住鼠标右键不放，即可打开 VT Fonts 菜单。选中“TrueType Fonts”再松开鼠标键，汉字就能显示出来了。再次打开菜单选中“Small”，令字体变小。再次打开菜单点击“Packed Fonts”，汉字显示就正常了。

这样要按着 Ctrl 点击三次鼠标，很麻烦。可在命令行带参数启动：**xterm -fa mono -fs 12**，指定新窗口使用 TrueType 字体，Small 大小，这样就只需 Ctrl+鼠标右键，打开菜单点击“Packed Fonts”，汉字显示就正常了。

为了持久的起作用，可以用 **vi /etc/X11/xinit/xinitrc** 打开配置文件。用 Shift+g 跳转到最后一行，改为：

```
export LANG="zh_CN.UTF-8"
```

```
exec xterm -fa mono -fs 12 -geometry 80x40+0+0 -name login
```

这样每次用 startx 启动图形界面，只需 Ctrl+鼠标右键，打开菜单点击“Packed Fonts”，汉字显示就正常了。

参考 <http://www.x.org/archive/X11R6.8.1/doc/xterm.1.html>

参考 <https://wiki.archlinux.org/index.php/Xterm>

## 11. 配置时区和时间

输入命令：**timedatectl**，即可查看本地时间、统一时间、主板时钟芯片 RTC 时间、时区等信息。

Linux 默认的时区是 UTC（英语：Coordinated Universal Time，协调世界时），基于国际原子时，并通过不规则的加入闰秒来抵消地球自转变慢的影响。

命令：**ls -l /etc/localtime**，可以看到这是软链接文件，指向了/usr/share/zoneinfo/UTC

命令：**hwclock -v**，查看主板上的硬件时钟。如果不加参数则等价于命令：**date**，查看系统软件时钟，这是在启动时，由硬件时钟，按时间标准、时区、夏令时进行了修正的本地时间。

在命令行执行：**timedatectl set-timezone Asia/Shanghai**，则系统软件时钟是东八区时间。该命令等价于：**ln -s /usr/share/zoneinfo/Asia/Shanghai /etc/localtime**，改变时区配置文件，即可更改时区为中国上海，这是 1912 年划分的，1949 年后改为北京时间，属于东八区。Ubuntu 下使用 tzselect 工具。

参考文档：[https://wiki.archlinux.org/index.php/System\\_time](https://wiki.archlinux.org/index.php/System_time)

通过 systemd-timesyncd 后台服务，可以自动进行网络校时

在命令行执行：**timedatectl set-ntp true**

在命令行执行：**timedatectl show-timesync**

参考文档：<https://wiki.archlinux.org/index.php/systemd-timesyncd>

提示：在 Windows 中把硬件时钟设置为本地时间了，而 Linux 默认硬件时钟为 UTC，然后根据时区进行修正。在安装双系统的电脑中就会发现从 Linux 重启进入 Windows，时间就相差 8 小时。因为 PC 机中的 BIOS 和主板上的时钟芯片 RTC 都是本地时区的时间，VMware 虚拟机内的时间也跟 Windows 时间保持一致。如果 Windows 要改为 UTC，需要修改注册表并禁用服务，很麻烦。

如果要修改 Linux 虚拟机的 RTC 为本地时间，执行：**timedatectl set-local-rtc true**，它会自动生成/etc/adjtime 文件。这种情况下 timesyncd 服务不会修改 RTC 时钟，需要手工执行 **hwclock -w** 修改 RTC 硬件时钟。

提示：LiveCD 中提供了 ntpdate 命令，执行：**ntpdate pool.ntp.org**，即可校准系统软件时钟的 UTC 时间，如果加参数 -q 只查询不校准。有时候会因为 GFW 的缘故，造成连接错误，可以用命令：**traceroute -U -p ntp**

pool.ntp.org 探测。但是装到硬盘上的系统中没有该命令，官方推荐安装 ntpd 软件包或 sntp 第三方工具。

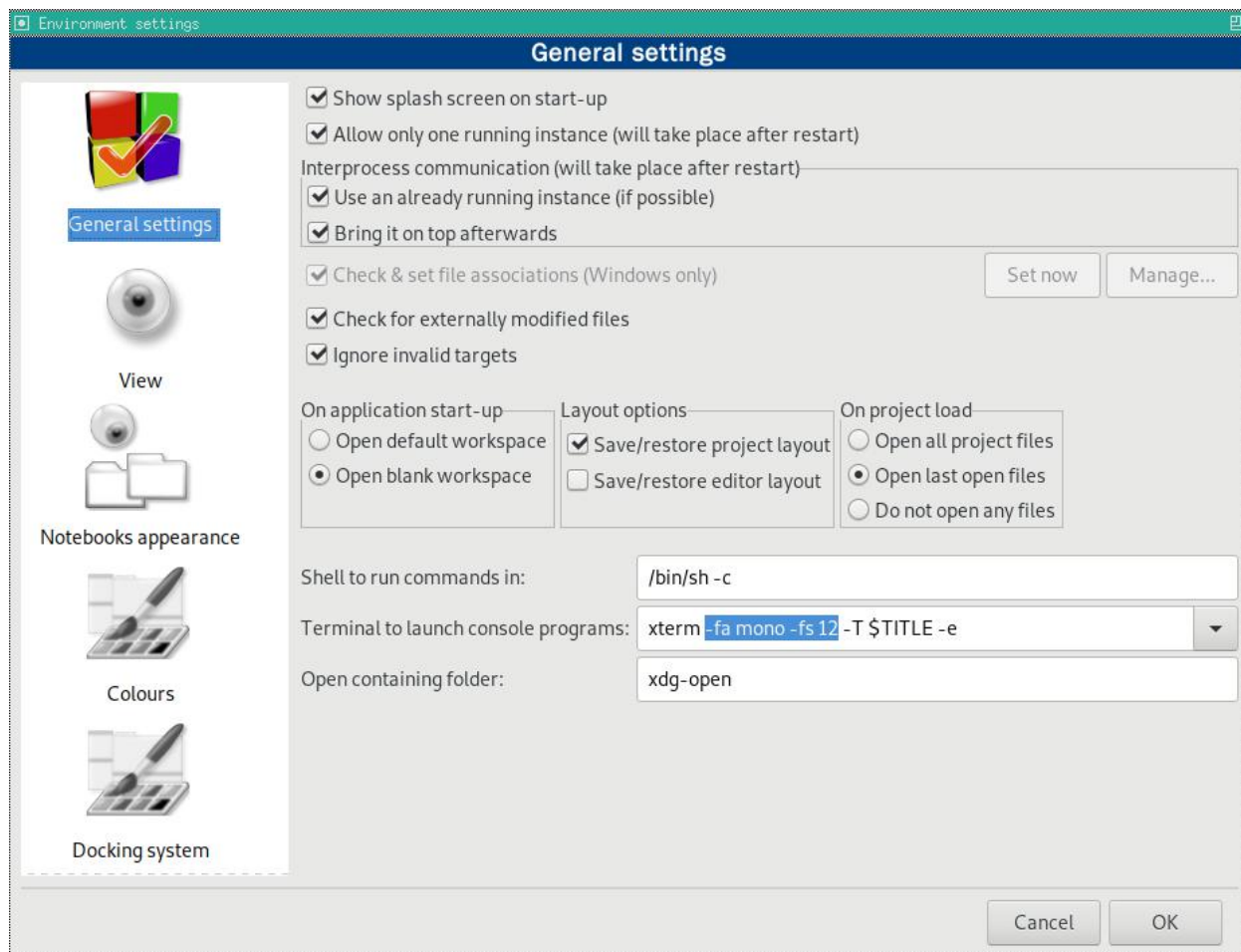
## 12. 在 codeblocks 中显示汉字

当前的图形界面没有桌面环境，无法使用汉字输入法。可以用 FileZilla 从其他服务器下载源程序文件，一定要确保文件使用 UTF-8 格式存储的，否则在编辑器中打开，汉字就是乱码。通常 Windows 下以 GBK 编码存储中文。

可以使用命令：`iconv -f gbk -t utf8 文件名`，即可自动转换中文编码。

如果程序运行时要使用 `printf` 等函数显示汉字，则 Code::blocks 的程序运行窗口，也就是 xterm 中，可能会出现汉字乱码。解决方法就是 `Ctrl+鼠标右键`，依次点击菜单“TrueType Fonts”、“Small”、“Packed Fonts”就好了。要想一劳永逸的话，就在“Settings”菜单-“Environment”中，修改 xterm 的参数。

注意：必须加在前面，因为“-e”参数后面会附加可执行文件的完整路径名。



## 13. 安装其他应用软件

参考文档：[https://wiki.archlinux.org/index.php/List\\_of\\_applications](https://wiki.archlinux.org/index.php/List_of_applications)

## 14. 汉字输入法

汉字输入法需要完整的桌面环境，如 GNOME / KDE / Xfce / LXDE 等，才能出现输入法状态栏、汉字输入框等。在当前简易的 Xorg 图形界面下，无法输入汉字。