

# Linux C语言开发工具链

---

## 本地开发 vs 交叉平台开发

### 本地开发

开发软件的系统与运行软件的系统是相同的

### 交叉平台开发

嵌入式系统开发属于交叉平台开发，也就是说开发软件的系统与运行软件的系统不同

- 宿主机：开发软件的平台，称为宿主机，往往是通用电脑
- 目标机：运行软件的平台，称为目标机，在这里是嵌入式系统

## (嵌入式) 开发工具链

包括

1. 编辑器
2. 交叉编译器
3. 调试工具
4. 项目管理工具

### GNU tools

GNU tools和其他一些优秀的开源软件可以完全覆盖上述类型的软件开发工具。为了更好的开发嵌入式系统，需要熟悉如下一些软件

#### vim编辑器

#### GCC

很多人认为GCC只是一个C编译器，其实GCC = GNU Compiler Collection（GNU编译器集合）

```
gcc - 符合ISO等标准的C编译器
g++ - 基本符合ISO标准的C++编译器
gcj - GCC的java前端
gnat - GCC的GNU ADA 95前端
```

**预处理器(cpp)、编译器(cc)、汇编器(as)、链接器(ld)等组件**

#### 一般情况下，c程序的编译过程为

1. 预处理（Pre-Processing）
2. 编译（Compiling）
3. 汇编（Assembling）
4. 链接（Linking）

## 预处理

### 一般是以下内容

1. 文件包含
2. 条件编译
3. 布局控制
4. 宏替换
5. 其他的预处理有：`#line`是用于修改预定义宏**LINE**（当前所在的行号）和**FILE**（当前源文件的文件名）；`#error` / `#warning`分别用于输出一个错误/警告信息；

## 编译成汇编代码(汇编)

最简单的程序也用到了系统函数库（C运行时库）

## 编译

优化编译

优化编译选项有：`-O0`缺省情况，不优化-`O1`-`O2`-`O3`

## 链接

主要工作

1. 将分散的数据和机器代码收集并合成一个单一的可加载并可执行的文件
2. 符号解析
3. 地址重定位

预编译函数库：静态库、动态库

静态库（.a后缀）——是一系列的目标文件的归档，静态链接时，静态库中的目标函数体会被复制到程序的可执行文件中

动态库（libname.so[major.minor.release]）——不具有“复制”操作，程序运行时根据需要载入内存，且动态地将函数调用与函数体关联到一起。动态库在内存中只有一份拷贝，因此也称为共享库。易于升级

**静态库与动态库，见代码**