

# SPEC CPU 2006

2018年12月21日 9:21

## 工作环境

centos7-1810版本

gcc版本: 4.8.5

## 预先准备

---

# 更新软件包

`sudo yum update`

# centos7默认编译环境(4.x)就可以用于编译SPEC, 如果是在最新的Ubuntu中, 需要手动安装旧版本编译工具

# 安装所需要的静态库, 包括libstdc++和libgfortran以及glibc相关包

`sudo yum install libstdc++-static libgfortran-static glibc-static glibc-utils`

---

## 安装步骤

---

# 解压安装包

`tar -xzf cpu2006.tar.gz`

# 进入解压后目录执行安装脚本, 此时大概率弹出没有对应二进制工具集的提示, 要求自己编译工具集

# 自己编译工具集有很多问题, 可以指定一个工具集

`./install`

# 引入环境变量

`source cshrc`

---

在toos/src目录下为工具集源代码, 提供了buildtools自动化脚本来编译工具集, 但是会出一些问题:

- md5sum: md5sum中自定义getline函数和std中getline函数冲突, 注释掉md5sum.c文件中对getline.h头文件的包含。
- perl: 在perl源码包的ext/IPC/SysV目录下, 找到SysV.xs文件, 将asm/page.h的包含注释掉, 因为高版本内核中已经没有该文件了。

解决以上两点之后, 虽然不能完全解决所有的问题, 但是工具集的编译基本就可以走完流程了。

神奇的地方在于：在执行完一次编译之后，再次执行install.sh脚本，此时大概率会找到一个已有的、可用的二进制工具集。注意，该工具集一开始就存在，不是手动编译出来的结果。

修改配置文件：从config目录中选择一个配置文件，比如Example-linux64-amd64-gcc43.cfg，复制该文件重命名为gcc43.cfg。修改配置文件，在编译选项中加上-static，表示使用静态编译。

---

```
## Base is low opt
default=base=default=default:
COPTIMIZE  = -O2 -fno-strict-aliasing -static
CXXOPTIMIZE = -O2 -fno-strict-aliasing -static
FOPTIMIZE  = -O2 -fno-strict-aliasing -static
```

---

执行编译任务：

---

```
# 编译所有整型benchmark
runspec --config=gcc43.cfg --action=build --tune=base int
# 编译所有浮点型benchmark
runspec --config=gcc43.cfg --action=build --tune=base fp
```

---

执行所有任务：

---

```
# size指定使用的输入数据，n用于控制迭代次数I表示忽略错误
# 执行所有的任务一次是为了能够在所有的benchmark中生成run文件夹
# run文件夹中包含该benchmark执行的需要的文件以及完整的执行命令
# 借助run文件夹，可以很容易了解如何执行某个特定的benchmark
runspec --config=gcc43.cfg --size=ref --noreportable -n=1 -I --tune=base int
runspec --config=gcc43.cfg --size=ref --noreportable -n=1 -I --tune=base fp
```

---