# 34. 构建你自己的应用程序

## 34.1. 在DPDK中编译一个示例程序

当编译示例应用程序(如 hello world)时，需要导出变量：RTE\_SDK 和 RTE\_TARGET。

~/DPDK$ cd examples/helloworld/

~/DPDK/examples/helloworld$ export RTE\_SDK=/home/user/DPDK

~/DPDK/examples/helloworld$ export RTE\_TARGET=x86\_64-native-linuxapp-gcc

~/DPDK/examples/helloworld$ make

CC main.o

LD helloworld

INSTALL-APP helloworld

INSTALL-MAP helloworld.map

生成的二进制文件默认放在build目录下：

~/DPDK/examples/helloworld$ ls build/app

helloworld helloworld.map

## 34.2. 在DPDK外构建自己的应用程序

示例应用程序(Hello World)可以复制到一个新的目录中作为开发目录：

~$ cp -r DPDK/examples/helloworld my\_rte\_app

~$ cd my\_rte\_app/

~/my\_rte\_app$ export RTE\_SDK=/home/user/DPDK

~/my\_rte\_app$ export RTE\_TARGET=x86\_64-native-linuxapp-gcc

~/my\_rte\_app$ make

CC main.o

LD helloworld

INSTALL-APP helloworld

INSTALL-MAP helloworld.map

## 34.3. 定制 Makefiles

### 34.3.1. 应用程序 Makefile

示例应用程序默认的makefile可以作为一个很好的起点，我们可以直接修订使用。它包括：

* 起始处包含 $(RTE\_SDK)/mk/rte.vars.mk
* 终止处包含 $(RTE\_SDK)/mk/rte.extapp.mk

用户必须配置几个变量：

* APP: 应用程序的名称
* SRCS-y: 源文件列表(\*.c, \*.S)。

### 34.3.2. 库 Makefile

同样的方法也可以用于构建库：

* 起始处包含 $(RTE\_SDK)/mk/rte.vars.mk
* 终止处包含 $(RTE\_SDK)/mk/rte.extlib.mk

唯一的不同之处就是用LIB名称替换APP的名称，例如：libfoo.a。

### 34.3.3. 定制 Makefile 动作

可以通过定制一些变量来制定 Makefile 动作。常用的动作列表可以参考文档 [Development Kit Build System](https://link.jianshu.com?t=dev_kit_build_system.html" \l "development-kit-build-system" \t "_blank) 章节 [Makefile Description](https://link.jianshu.com?t=dev_kit_build_system.html" \l "makefile-description" \t "_blank) 。

* VPATH: 构建系统将搜索的源文件目录，默认情况下 RTE\_SRCDIR 将被包含在 VPATH 中。
* CFLAGS\_my\_file.o: 编译c文件时指定的编译flag标志。
* CFLAGS: C编译标志。
* LDFLAGS: 链接标志。
* CPPFLAGS: 预处理器标志（只是用于汇编.s文件）。
* LDLIBS: 链接库列表(如 -L /path/to/libfoo - lfoo)。

作者：半天妖  
链接：https://www.jianshu.com/p/31d8e4401b06  
来源：简书  
简书著作权归作者所有，任何形式的转载都请联系作者获得授权并注明出处。