Gtest-Gcovr工程构建说明

一.准备阶段

1.gcovr\_html.sh

路径：./mycode（工作目录）

作用：1.运行单元测试相关的可执行程序;

2.使用gcovr工具生成覆盖率报告（默认html格式）;

2.gcovr

路径：./mycode（工作目录）

作用：生成覆盖率报告

3.主Makefile

路径：./mycode（工作目录）

作用：1.设置编译选项并设置为环境变量;

2.指定目标路径，执行目标路径下的Makefile;

3.链接.o文件，生成可执行文件;

4.执行该目录下的gcovr脚本;

5.目标工程Makefile（目标工程自带内容需改动）

路径：./mycode/target

作用：根据目标主Makefile设置的编译选项编译生成.o/.gcno文件;

6.output目录

路径：./mycode

作用：存储覆盖率报告相关的html文件和gtest生成的单元测试报告;

二.编译阶段

1.C/C++混编

1.1若cpp文件需要调用c文件的api，则需要在该api的声明处添加“extern “C””标识，例：

#ifdef \_\_cplusplus

extern “C” {

#endif

extern int get\_messageid\_priority(unsigned int messageid);

#ifdef \_\_cplusplus

}

#endif

1.2 在涉及到同时编译c和cpp程序时，Makefile生成.o文件需按如下方式编写：

%.o : %.c

$(CC) $(CPPFLAGS) $(CFLAGS) $(INCLUDES) $(LIBS) -c $< -o $@

%.o : %.c

$(CC) $(CPPFLAGS) $(CXXFLAGS) $(INCLUDES) $(LIBS) -c $< -o $@

2.GCC/G++选项

2.1 CPPFLAGS ：预处理器选项

CPPFLAGS := -fprofile-arcs -fgest-coverage -fPIC -O2

2.2 CFLAGS/CXXFLAGS：C/C++编译器选项

CFLAGS/CXXFLAGS := -g -Wextra -pthread

2.3 LIBS：链接选项

LIBS := -ldbus-glib-1 -ldbus-1 -lgobject-2.0 -lglib-2.0 -lpthread -lrt

LIBS += ../../lib/gtest\_main.a

LIBS += -lgcov