原创 weixin_39809175 于 2020-12-21 05:35:15 发布 246 收藏 版权

文章标签： gcovr 安装

嗷嗷嗷！！！ 好激动，我好蠢，不过最后还是解决了。呜呜呜

有些都是东一块西一块查的，如果有侵权欢迎私信我，我注明出处。

一 gcov&CMake

昨天试了下测试代码和被测代码都是c++的情况，直接编译生成gcovc文件，再一运行，生成gcda文件。脚本统计，blingbling生成了报表，简直漂亮！

不过我们的工程比较大= = 。编译时也需要很多依赖文件。

因此使用场景为：在机器A目录编译，拷贝纯bin文件到B目录上运行。编译方式为CMake

1 修改编译脚本

找到CMakeList.txt文件,添加如下内容：

```
# coverage option

OPTION (ENABLE_COVERAGE "Use gcov" OFF)

MESSAGE(STATUS ENABLE_COVERAGE=${ENABLE_COVERAGE})

IF(ENABLE_COVERAGE)

SET(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -fprofile-arcs -ftest-coverage")

SET(CMAKE_C_FLAGS "${CMAKE_C_FLAGS} -fprofile-arcs -ftest-coverage")

# 其中-fprofile-arcs 用来生成 gcno 文件； -ftest-coverage 用来在统计时生成 gcda 文件

# SET(CMAKE_EXE_LINKER_FLAGS "${CMAKE_EXE_LINKER_FLAGS} -fprofile-arcs -ftest-coverage") ENDIF()
```

编译时使用命令如下：

```
cmake -DENABLE_COVERAGE=ON ..
```

编译后查看文件目录：build一路向下

```
******/build/CMakeFiles/ad_server.dir/src

$ll -rw-rw-r-- 1 mobdev mobdev 1431748 Jun 22 18:24A.cpp.gcno-rw-rw-r-- 1 mobdev mobdev 2138256 Jun 22 18:24A.cpp.o-rw-rw-r-- 1 mobdev mobdev 314180 Jun 22 18:25B.cpp.gcno-rw-rw-r-- 1 mobdev mobdev 871944 Jun 22 18:25B.cpp.o-rw-rw-r-- 1 mobdev mobdev 2053332 Jun 22 18:25C.cpp.gcno-rw-rw-r-- 1 mobdev mobdev 3931584 Jun 22 18:25C.cpp.o-rw-rw-r-- 1 mobdev mobdev 1063036 Jun 22 18:24D.cpp.gcno-rw-rw-r-- 1 mobdev mobdev 3280472 Jun 22 18:24 D.cpp.o
```

如上表示编译成功

2 拷贝bin文件到目录B，注意保持目录B与目录A的为同一次编译结果(我是检查md5文件)

如果是从机器A到机器B，注意配置gcov_prefix交叉编译。

3 启动你的服务，按原方式执行测试用例。case执行完成后进入下一步

4 生成gcda文件。vim test.sh,内容如下：

```
#!/bin/sh

SERVER_NAME=$1pid=`ps -ef | grep $SERVER_NAME | grep -v "grep" | awk '{print $2}'`

echo $pid

gdb-q attach $pid <<__eof__>

p __gcov_flush()

__EOF__
```

执行命令:sh test.sh your_servermane

5 检查：因为这次我的编译和测试在同台机器的不同目录，因为gcda文件会生成到编译时的gcno同目录，检查其中已有gcda文件。如下：

```
******/build/CMakeFiles/ad_server.dir/src

$ll -rw-rw-r-- 1 mobdev mobdev 71772 Jun 22 18:55A.cpp.gcda-rw-rw-r-- 1 mobdev mobdev 1431748 Jun 22 18:24A.cpp.gcno-rw-rw-r-- 1 mobdev mobdev 2138256 Jun 22 18:24A.cpp.o-rw-rw-r-- 1 mobdev mobdev 18452 Jun 22 18:55B.cpp.gcda-rw-rw-r-- 1 mobdev mobdev 314180 Jun 22 18:25B.cpp.gcno-rw-rw-r-- 1 mobdev mobdev 871944 Jun 22 18:25 B.cpp.o
```

二 生成html页面

进入gcda和gcno文件所在目录，执行

```
lcov-c -o result.info -b . -d . //生成info文件

genhtml result.info -o Report //生成html文件

tar cvf Report.tar.gz Report //压缩文件

sz Report.tar.gz //下载到windows系统
```

解压打开其中的index.html即可看到测试 代码覆盖率

三 生成xml报告

1 安装gcovr

```
cd /usr/local

wget https://github.com/gcovr/gcovr/archive/3.2.tar.gz

tar -xvf 3.2.tar.gz

cd gcovr-3.2/scripts

cp gcovr/usr/bin
```

2 在编译路径下执行 gcovr -r .即可查看覆盖率情况

3 为了使其生成Cobertura可用的xml文件，可以使用命令

```
gcovr -r . --output yourdir/coverage.xml -xml-pretty

vim coverage.xml即可看到xml报告
```

四 问题

1 描述：无法生成gcda文件

原因如下：

用户代码调用 exit 正常结束时，gcov_exit 函数将到调用，其继续调用 __gcov_flush 函数输出统计数据到 *.gcda 文件中

若用户进程异非调用 exit 正常退出，覆盖率统计数据就无法输出，也就无从生成报告了。后台服务程序若非专门设计，一旦启动就很少主动退出。用 kill 杀死进程强制退出时就不会调用 exit，因此没有覆盖率统计结果产生。

解决：执行(一)4步骤即可。

2 描述:无法组合gcda和gcno文件:"stamp mismatch with notes file"

原因如下：

网上找了好多答案，说是编译版本不一致，导致时间戳不一致，我一直没有理解。

确认自己这儿是因为生成gcno文件的编译版本和生成gcda文件的版本不一致。

我的操作：1编译二进制文件ad.server，生成了gcno文件2拷贝ad.server进测试环境，测试生成了gcda3中途又编译了一次ad.server，生成了新的gcno文件.

即gcno和gcda文件使用的ad.server并不是同一次编译的结果。

最后通过对比两个地方的ad.server的md5发现了不同，我好蠢= =

解决：解决很简单，确保你的版本就ok了，可以对比二进制文件的md5.

查看时间戳：hexdump -e '""%x\n"' -s8 -n4 A.cpp.gcno 可以看到时间戳。

```
hexdump -e '""%x\n"' -s8 -n4 A.cpp.gcda 可以看到时间戳
```

3 描述：想要让gcda文件生成在指定目录

场景:有时候是在机器A编译,机器B运行，这种情况就会不能生成gcda文件，提示找不到目录

解决：gcov的交叉编译

操作如下：

```
vim /etc/profile 进行配置

export GCOV_PREFIX="/data/ad_server/ad_server.dir" //gcda的目标路径

export GCOV_PREFIX_STRIP=9 //向上数你的路径到需要配置的那个
```

比如我的编译路径是/data/code/adserver/6/5/4/3/2/*.gcno,我就设置了9，应该也可以设置export GCOV_PREFIX_STRIP=999等很大然后配置全路径。我没试

```
source /etc/profile 使其生效

重启服务进程

执行测试代码，exit()退出，查看profile的设置路径，已经有gcda文件

拷贝gcno文件到gcda路径，统计代码覆盖率

blingbling的就大功告成了！
```

我的问题：配置了n次该路径一直不生效，最后要哭了。今天偶然发现需要重启进程，忧伤极了

4 描述：无法使用gcovr生成覆盖率--得到的覆盖率为0%

```
$gcovr -r .------------------------------------------------------------------------------GCC Code Coverage Report

Directory: .------------------------------------------------------------------------------File Lines Exec Cover Missing------------------------------------------------------------------------------

------------------------------------------------------------------------------TOTAL0 0 --%

------------------------------------------------------------------------------
```

解决:执行gcovr -r .在编译的大路径下即可，或者在执行时指定source 路径 如 gcovr -r /home/test/mytest/XXXX/

目录不对，只有gcda和gcno、cpp.o文件无法生成覆盖率。需要源码--猜测

5 描述：使用 --remove 后未在指定路径看到 -o 后的新的 info 文件

解决：实际生成了，使用命令 find / -name "new_file.info" 发现在根目录下，不知道为什么会这样，使用时更新为 指定绝对路径 就 ok 了

dev头文件mingw.h 04-04
dev的头文件，方便下载了DEV-CPP但是缺失了头文件的用户使用。

paddlepaddle_gpu-2.0.0b0-cp36-cp36m-linux_aarch64.whl 10-23
为Jetson Xavier NX 编译好的paddlepaddle库，可以直接pip install使用

gcovr 安装_GCOV&&LCOV&GCOVR入门 weixin_39869929的博客 491
一、概述代码覆盖率(Code coverage)是衡量软件测试质量的一个重要指标。它描述了当一个特定的测试套件(test suite)运行时，程序源代码被执行的程度。例如，一些更...

关于驱动程序的头文件 EvilSwell—一無論做什麼,记得为自己而做,那就毫无恐... 564
驱动程序： #include 是在linux-2.6.29/include/linux下面寻找源文件。 #include 是在linux-2.6.29/arch/arm/include/asm下面寻找源文件。 #include 是在linux-2.6.29/arch/ar...

dev自定义的头文件可以放在别的文件夹里 yuanyun的博客 364
我们自己写的include"x.h"不一定要和项目放在同一级文件夹里了！ 除了常规的 ./上一级下一级调用还有一种方法！ 头文件放到哪里都可以。 就是建一个文件夹把头文件全...

linux error: jni.h: No such file or directory AinUser的博客 618
情境 这是我在通过cpp文件，生成dll文件的时候，遇到的 # 原因 虽然我在cpp文件中引入了，但是生成dll文件的时候，好像并没有加载到 # 解决 1. window环境 如果是wi...

DevC++ 自定义头文件并使用 橘猫吃不胖的博客 9829
1.打开DevC++，点击【新建】——选择项目 2.选择Basic——Console Application——选择C语言/C++项目——输入名称——点击确定 3.选择保存位置，保存类型为Dev-C...

DEV-C++如何使用自定义的头文件 Labolasyai的博客 3627
方法一： 将函数定义放在头文件中 将.c文件直接改为.h文件就可以直接引用，无需IDE进行任何操作（如果在同一个文件夹下）。 #include"demo.h" 但是这只能被一个源...

DevC++如何安装 自定义头文件并使用 m0_51955470的博客 2409
首先我们打开DevC++，然后点击新建，新建一个控制台应用程序 取一个喜欢的文件名，然后会出现如下界面，点击新建单元 将头文件源码放入 找到空白位置，右键，...

（单开）关于页目录项 weixin_39869931的博客 665
引用一位老铁的话 就是相当于一个一本书的目录 一个H1的标题就是目录号，在H1下的所有H2标题都是该目录号索引的页表，每个H2标题里的H3就是第三级页表，而这...

C/C++代码覆盖率统计：gcov/lcov/gcovr以及集成到jenkins 至绽至强的博客 1695
使用gcov/lcov/或gcovr工具进行代码覆盖率统计时，都需要在编译选项中加入"-fprofile-arcs -ftest-coverage"两个选项，才能生成.gcno和gcda文件。 如果是CMAKE工程，...

使用gcov和lcov测试代码覆盖率 qq_40855100的博客 284
参考文章 gcov介绍 gcov是Linux下GCC自带的一个C/C++代码覆盖率分析工具,因此只要安装了gcc，就不需要再次安装了。 lcov介绍 gcov能够生成代码覆盖信息，但是不...

linux下的c++项目代码覆盖率的产生方法,展示C代码覆盖率的gcovr工具简介及相关命令使用示例... weixin_39823322的博客 115
最近，因为要展示某项目的单元测试的代码覆盖率，我无意间在网上找到了gcovr工具，使用之后，觉得这个工具相当的不错，于是便写下这篇文章，可供相关的开发人员...

gcovr 安装_C/C++代码覆盖率统计工具：gcov&&gcovr安装和简单使用 weixin_39565777的博客 835
gcov安装Linux ver： gcovr是gcc的自带功能 属于GNU 不需特别安装Windows ver： 在windows下安装可以使用gcovr的gcc 之前试过mingw和Cygwin64 Terminal，在后续可...

C++中"头文件"和"源文件"介绍 xidfluca的专栏 2255
一、C++ 编译概述 通常，在一个 C++ 程序中，只包含 cpp 文件和 .h 文件两类文件。其中，.cpp 文件被称作 C++ 源文件，里面放的都是 C++ 的源代码；而 .h 文件则被...

codeblocks怎么新建c++文件_如何在devc创建工程文件 weixin_39900830的博客 200
从计划始到，主要用于C语言入门学习。首先，新建一个文件夹devpro。 src文件夹存放源代码。打开devc++。新建项目选择windows application，取一个名字，选择C语言...

DEVC++中的"万能头文件"<bits/stdc++.h> 最新发布 ian20120124_的博客 610
目录前言第一部分这个头文件相信会一点DEVC++的人都不会陌生 但是它其实说起来万能 但实际上讲+ 并不是完全万能的 打个比方说就不包含在这个头文件里面的我们来...