

在linux用gdb查看stl中的数据结构

在linux用gdb或者cgdb计较不爽的地方是无法打印STL的东西，所有啊去网上找了找解决方案

<https://www.douban.com/note/182826844/?qq-pf-to=pcqq.c2c>

本帖把怎么配置这个东西写出了，万一以后忘了，可以回头找找。

首先是下载gdb文件 <https://sourceware.org/gdb/wiki/STLSupport> -----》找到网页里面的

...the pretty-printers are being used and instead of the expanded details, which are more human-readable format. To print the classes in the old style, use the /r (raw) switch in th if the Python pretty-printers were not loaded.

gdb-stl-views is a set of GDB macros that can display the contents of many STL containers: `priority_queue`, `bitset`, `string`, and `widestring`. Written and currently maintained by Dan Marines code. In place of a legal notice, here is a blessing: May you do good and not evil. May you find freely, never taking more than you give!

- You can download [it here](#) or [here](#)
 - Tutorials and an alternative download are [hosted at yolinux.com](#).
- **gdb++** is a Perl script which extends gdb. It comes bundled as part of the Devel: GDB. Reflex follow the [gdb++ usage instructions](#). Developed by Stanford PhD student Antal Novak.
- There are other options. Tom Mairn wrote a set of GDB macros similar to Dan's ([http://thr](#)) but it doesn't cover as wide a variety of containers. Glad Mishne wrote a different set of mac unmainained and it works only with SGI's STL implementation, which very few GCC users use
- Iterators: how to display the item the iterator points at (tested on gdb 6 with a list): `print *iter_`

HomePage | RecentChanges | FindPage | HelpContents | STLSupport
只读网页 | 信息 | 附件 | 更多操作: ▾

附件'stl-views-1.0.3.gdb'

下载

当前配置不允许嵌入文件 stl-views-1.0.3.gdb 因为它的 mimetype application/octet-

附件

如果要在正文中引用附件，请参考下表，使用 `attachment:filename`，请不要用 (T

- [\[下载\] \[查看\]](#) (2008-09-15 15:06:10, 17.5 KB) [[attachment:stl-views-1.0.3.gdb]

你无权为此页上载附件。

然后点击进去下载stl_views_1.0.3.gdb

接下来把这个东西当到linux下，位置随便放，我放到了我的用户目录下

```
[yongchao@fsdev20 ~]$ pwd
/home/yongchao
[yongchao@fsdev20 ~]$ ll
total 68
drwxr-xr-x  2 yongchao fsdev  4096 Mar 21 18:11 CMakeList_Learning
drwxr-xr-x  2 yongchao fsdev  4096 Feb 23 11:11 cscope
-rwxr-xr-x  1 yongchao fsdev    27 Feb 18 14:08 env_compile.sh
lrwxrwxrwx  1 yongchao fsdev    40 Apr 29 17:40 etc -> /home/yongchao/git/farestar/FareStar/etc
drwxr-xr-x  3 yongchao fsdev  4096 Apr 17 16:17 farestar
drwxr-xr-x 24 yongchao fsdev  4096 Mar 17 15:55 farestar_uuid
drwxr-xr-x  4 yongchao fsdev  4096 May  6 09:39 git
drwxr-xr-x 14 yongchao fsdev  4096 Apr 27 18:31 git2
drwxr-xr-x  5 yongchao fsdev  4096 Apr 18 19:55 myCPP11
drwxr-xr-x  3 yongchao fsdev  4096 Apr 29 10:38 mypython
-rw-r--r--  1 yongchao fsdev 2133 Feb 18 13:46 shellConfig.txt
-rw-r--r--  1 yongchao fsdev 17901 May  6 16:25 stl-views-1.0.3.gdb
drwxr-xr-x  2 yongchao fsdev  4096 May  6 16:30 temp
drwxr-xr-x  4 yongchao fsdev  4096 May  2 18:17 Unix_Learning
[yongchao@fsdev20 ~]$
```

然后写一个小程序测试一下，

```
[yongchao@fsdev20 temp]$ pwd
/home/yongchao/temp
[yongchao@fsdev20 temp]$ ls
main.cpp  test
[yongchao@fsdev20 temp]$ cat main.cpp
#include <map>
using namespace std;
int main()
{
    map<int, char> myMap;
    myMap[1] = 'a';
    myMap[2] = 'b';
    return 0;
}
[yongchao@fsdev20 temp]$ g++ -o test main.cpp -g
```

接着cgdb test 进入调试模式，然后加载刚才的stl_views_1.0.3.gdb

```
~
~
~
~
1 #include <map>
2 using namespace std;
3 int main()
4 {
5     map<int, char> myMap;
6     myMap[1] = 'a';
7     myMap[2] = 'b';
8     return 0;
9 }
~
~
~
/home/yongchao/temp/main.cpp
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/lic
This is free software: you are free to change and redistribute
There is NO WARRANTY, to the extent permitted by law. Type "s
and "show warranty" for details.
This GDB was configured as "x86_64-pc-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online a
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"..
Reading symbols from test...done.
(gdb) b main
Breakpoint 1 at 0x400941: file main.cpp, line 6.
(gdb) source /home/yongchao/stl-views-1.0.3.gdb
```

接着就可以看看pmap的命令了

```
(gdb) help pmap
Prints std::map<TLeft and TRight> or std::multimap<TLeft and TRight> information. Works for std::multimap as well.
Syntax: pmap <map> <TtypeLeft> <TypeRight> <valLeft> <valRight>: Prints map size, if T defined all elements or just element(s) with val(s)
Examples:
pmap m - prints map size and definition
pmap m int int - prints all elements and map size
pmap m int int 20 - prints the element(s) with left-value = 20 (if any) and map size
pmap m int int 20 200 - prints the element(s) with left-value = 20 and right-value = 200 (if any) and map size
(gdb)
```

pmap variable----->打印variable这个map的定义和map里面的个数

pmap variable int int(就是单纯的两个int) ----->打印pmap的元素和map的个数

pmap variable int int 20----->打印索引是20的map的值 和map的个数

pmap variable int int 20 200----->打印索引是20 值是200的map值和map的个数

```

1 #include <map>
2 using namespace std;
3 int main()
4 {
5     map<int, char> myMap;
6     myMap[1] = 'a';
7     myMap[2] = 'b';
8     return 0;
9 }

```

/home/yongchao/temp/main.cpp

```

(gdb) pmap myMap int int
No symbol "myMap" in current context.
(gdb) r
Starting program: /home/yongchao/temp/test
warning: File "/usr/local/lib64/libstdc++.so

Breakpoint 1, main () at main.cpp:6
(gdb) n
(gdb)
(gdb)
(gdb) pmap myMap int int
elem[0].left: $37 = 1
elem[0].right: $38 = 97
elem[1].left: $39 = 2
elem[1].right: $40 = 98
Map size = 2
(gdb)

```

东西很好用，就怕以后链接失效找不到这个文件了，下面会把这个文件东西拷贝在下面。以后用到了，自己建立一个stl_views_1.0.3.gdb，然后把下面的东西拷贝进去使用。
