```
// $Id: string_set.h,v 1.3 2018-04-06 15:07:42-07 - - $

#ifndef __STRING_SET__
#define __STRING_SET__

#include <string>
#include <unordered_set>

#include <stdio.h>

struct string_set {
   string_set();
   static std::unordered_set<std::string> set;
   static const std::string* intern (const char*);
   static void dump (FILE*);
};

#endif

```

```
// $Id: string_set.cpp,v 1.6 2019-04-03 17:32:31-07 - - $

#include <string>
#include <unordered_set>
using namespace std;

#include "string_set.h"

unordered_set<string> string_set::set;

string_set::string_set() {
   set.max_load_factor (0.5);
}

string_set set;

const string* string_set::intern (const char* string) {
   auto handle = set.insert (string);
   return &*handle.first;
}

void string_set::dump (FILE* out) {
   static unordered_set<string>::hasher hash_fn
               = string_set::set.hash_function();
   size_t max_bucket_size = 0;
   for (size_t bucket = 0; bucket < set.bucket_count(); ++bucket) {
      bool need_index = true;
      size_t curr_size = set.bucket_size (bucket);
      if (max_bucket_size < curr_size) max_bucket_size = curr_size;
      for (auto itor = set.cbegin (bucket);
           itor != set.cend (bucket); ++itor) {
         if (need_index) fprintf (out, "string_set[%4zu]: ", bucket);
                    else fprintf (out, "           %4s   ", "");
         need_index = false;
         const string* str = &*itor;
         fprintf (out, "%20zu %p->\"%s\"\n", hash_fn(*str),
                  str, str->c_str());
      }
   }
   fprintf (out, "load_factor = %.3f\n", set.load_factor());
   fprintf (out, "bucket_count = %zu\n", set.bucket_count());
   fprintf (out, "max_bucket_size = %zu\n", max_bucket_size);
}

```

```
// $Id: main.cpp,v 1.2 2016-08-18 15:13:48-07 - - $

#include <string>
using namespace std;

#include <assert.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include "string_set.h"

int main (int argc, char** argv) {
   for (int i = 1; i < argc; ++i) {
      const string* str = string_set::intern (argv[i]);
      printf ("intern (\"%s\") returned %p->\"%s\"\n",
              argv[i], str, str->c_str());
   }
   string_set::dump (stdout);
   return EXIT_SUCCESS;
}

```

```
# $Id: Makefile,v 1.7 2019-04-03 17:30:36-07 - - $

COMPILECPP  = g++ -std=gnu++17 -g -O0 -Wall -Wextra -Wold-style-cast
MAKEDEPCPP  = g++ -std=gnu++17 -MM

MKFILE   = Makefile
DEPFILE  = Makefile.dep
SOURCES  = string_set.cpp main.cpp
HEADERS  = string_set.h
OBJECTS  = ${SOURCES:.cpp=.o}
EXECBIN  = test_string_set
SRCFILES = ${HEADERS} ${SOURCES} ${MKFILE}

all : ${EXECBIN}

${EXECBIN} : ${OBJECTS}
        ${COMPILECPP} ${OBJECTS} -o ${EXECBIN}

%.o : %.cpp
        ${COMPILECPP} -c $<

ci :
        cid + ${SRCFILES}

clean :
        -rm ${OBJECTS} ${DEPFILE}

spotless : clean
        - rm ${EXECBIN} Listing.ps Listing.pdf test.out test.grind

${DEPFILE} :
        ${MAKEDEPCPP} ${SOURCES} >${DEPFILE}

dep :
        - rm ${DEPFILE}
        ${MAKE} --no-print-directory ${DEPFILE}

include ${DEPFILE}

test : ${EXECBIN}
        valgrind --log-file=test.grind --leak-check=full ${EXECBIN} * *
* >test.out 2>&1

lis : test
        mkpspdf Listing.ps ${SRCFILES} ${DEPFILE} test.out test.grind

again : ${SRCFILES}
        make --no-print-directory spotless dep ci test lis
```

```
string_set.o: string_set.cpp string_set.h
main.o: main.cpp string_set.h
```

```
intern ("HEADER.html") returned 0x5a230b8->"HEADER.html"
intern ("Makefile") returned 0x5a231e8->"Makefile"
intern ("Makefile.dep") returned 0x5a23338->"Makefile.dep"
intern ("RCS") returned 0x5a233f8->"RCS"
intern ("main.cpp") returned 0x5a23598->"main.cpp"
intern ("main.o") returned 0x5a23658->"main.o"
intern ("string_set.cpp") returned 0x5a23728->"string_set.cpp"
intern ("string_set.h") returned 0x5a237f8->"string_set.h"
intern ("string_set.o") returned 0x5a238c8->"string_set.o"
intern ("test_string_set") returned 0x5a23b08->"test_string_set"
intern ("HEADER.html") returned 0x5a230b8->"HEADER.html"
intern ("Makefile") returned 0x5a231e8->"Makefile"
intern ("Makefile.dep") returned 0x5a23338->"Makefile.dep"
intern ("RCS") returned 0x5a233f8->"RCS"
intern ("main.cpp") returned 0x5a23598->"main.cpp"
intern ("main.o") returned 0x5a23658->"main.o"
intern ("string_set.cpp") returned 0x5a23728->"string_set.cpp"
intern ("string_set.h") returned 0x5a237f8->"string_set.h"
intern ("string_set.o") returned 0x5a238c8->"string_set.o"
intern ("test_string_set") returned 0x5a23b08->"test_string_set"
intern ("HEADER.html") returned 0x5a230b8->"HEADER.html"
intern ("Makefile") returned 0x5a231e8->"Makefile"
intern ("Makefile.dep") returned 0x5a23338->"Makefile.dep"
intern ("RCS") returned 0x5a233f8->"RCS"
intern ("main.cpp") returned 0x5a23598->"main.cpp"
intern ("main.o") returned 0x5a23658->"main.o"
intern ("string_set.cpp") returned 0x5a23728->"string_set.cpp"
intern ("string_set.h") returned 0x5a237f8->"string_set.h"
intern ("string_set.o") returned 0x5a238c8->"string_set.o"
intern ("test_string_set") returned 0x5a23b08->"test_string_set"
string_set[   1]:  1516108490113098673 0x5a238c8->"string_set.o"
                   2099682443743551108 0x5a23658->"main.o"
                  17041606903804112922 0x5a23598->"main.cpp"
string_set[   3]: 13646535705723827550 0x5a230b8->"HEADER.html"
string_set[   5]:  8902767590177878864 0x5a231e8->"Makefile"
string_set[   6]: 15286446792580072886 0x5a23728->"string_set.cpp"
string_set[  11]:   994128771139992428 0x5a233f8->"RCS"
string_set[  15]:  2246613038755228464 0x5a23338->"Makefile.dep"
string_set[  20]:  7517842887488357474 0x5a237f8->"string_set.h"
string_set[  35]:  9799794095794511326 0x5a23b08->"test_string_set"
load_factor = 0.270
bucket_count = 37
max_bucket_size = 3
```

```
==17516== Memcheck, a memory error detector
==17516== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==17516== Using Valgrind-3.14.0.GIT and LibVEX; rerun with -h for copyright info
==17516== Command: test_string_set HEADER.html Makefile Makefile.dep RCS main.cpp main.o string_set.cpp string_set.h string_set.o test_string_set HEADER.html Makefile Makefile.dep RCS main.cpp main.o string_set.cpp string_set.h string_set.o test_string_set HEADER.html Makefile Makefile.dep RCS main.cpp main.o string_set.cpp string_set.h string_set.o test_string_set
==17516== Parent PID: 17515
==17516== 
==17516== 
==17516== HEAP SUMMARY:
==17516==     in use at exit: 0 bytes in 0 blocks
==17516==   total heap usage: 44 allocs, 44 frees, 1,805 bytes allocated
==17516== 
==17516== All heap blocks were freed -- no leaks are possible
==17516== 
==17516== For counts of detected and suppressed errors, rerun with: -v
==17516== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```