

```
1: // $Id: stringset.h,v 1.3 2015-03-25 19:03:01-07 - - $
2:
3: #ifndef __STRINGSET__
4: #define __STRINGSET__
5:
6: #include <string>
7:
8: #include <stdio.h>
9:
10: const std::string* intern_stringset (const char*);
11:
12: void dump_stringset (FILE*);
13:
14: #endif
15:
```

```
1: // $Id: stringset.cpp,v 1.8 2015-03-25 19:03:01-07 - - $
2:
3: #include <unordered_set>
4: using namespace std;
5:
6: #include "stringset.h"
7:
8: using stringset = unordered_set<string>;
9:
10: stringset set;
11:
12: const string* intern_stringset (const char* string) {
13:     pair<stringset::const_iterator,bool> handle = set.insert (string);
14:     return &*handle.first;
15: }
16:
17: void dump_stringset (FILE* out) {
18:     size_t max_bucket_size = 0;
19:     for (size_t bucket = 0; bucket < set.bucket_count(); ++bucket) {
20:         bool need_index = true;
21:         size_t curr_size = set.bucket_size (bucket);
22:         if (max_bucket_size < curr_size) max_bucket_size = curr_size;
23:         for (stringset::const_local_iterator itor = set.cbegin (bucket);
24:             itor != set.cend (bucket); ++itor) {
25:             if (need_index) fprintf (out, "stringset[%4lu]: ", bucket);
26:             else fprintf (out, "          %4s ", "");
27:             need_index = false;
28:             const string* str = &*itor;
29:             fprintf (out, "%22lu %p->\"%s\\\"\\n",
30:                     set.hash_function() (*str),
31:                     str, str->c_str());
32:         }
33:     }
34:     fprintf (out, "load_factor = %.3f\\n", set.load_factor());
35:     fprintf (out, "bucket_count = %lu\\n", set.bucket_count());
36:     fprintf (out, "max_bucket_size = %lu\\n", max_bucket_size);
37: }
38:
```

```
1: // $Id: main.cpp,v 1.3 2015-03-25 19:03:01-07 - - $
2:
3: #include <string>
4: using namespace std;
5:
6: #include <assert.h>
7: #include <stdio.h>
8: #include <stdlib.h>
9: #include <string.h>
10:
11: #include "stringset.h"
12:
13: int main (int argc, char **argv) {
14:     for (int i = 1; i < argc; ++i) {
15:         const string* str = intern_stringset (argv[i]);
16:         printf ("intern (\"%s\") returned %p->\"%s\"\\n",
17:             argv[i], str, str->c_str());
18:     }
19:     dump_stringset (stdout);
20:     return EXIT_SUCCESS;
21: }
22:
```

```
1: # $Id: Makefile,v 1.8 2014-10-07 17:37:02-07 - - $
2:
3: GPP      = g++ -g -O0 -Wall -Wextra -std=gnu++11
4: GRIND    = valgrind --leak-check=full --show-reachable=yes
5:
6: all : teststring
7:
8: teststring : main.o stringset.o
9:             ${GPP} main.o stringset.o -o teststring
10:
11: %.o : %.cpp
12:       ${GPP} -c $<
13:
14: ci :
15:     cid + Makefile stringset.h stringset.cpp main.cpp
16:
17: spotless : clean
18:     - rm teststring Listing.ps Listing.pdf test.out test.err
19:
20: clean :
21:     -rm stringset.o main.o
22:
23: test : teststring
24:     ${GRIND} teststring * * * >test.out 2>test.err
25:
26: lis : test
27:     mkpspdf Listing.ps stringset.h stringset.cpp main.cpp \
28:         Makefile test.out test.err
29:
30: # Dependencies.
31: main.o: main.cpp stringset.h
32: stringset.o: stringset.cpp stringset.h
33:
```

```
1: intern ("HEADER.html") returned 0x4c30158->"HEADER.html"
2: intern ("Makefile") returned 0x4c30228->"Makefile"
3: intern ("RCS") returned 0x4c302e8->"RCS"
4: intern ("main.cpp") returned 0x4c303b8->"main.cpp"
5: intern ("main.o") returned 0x4c30478->"main.o"
6: intern ("stringset.cpp") returned 0x4c30548->"stringset.cpp"
7: intern ("stringset.h") returned 0x4c30618->"stringset.h"
8: intern ("stringset.o") returned 0x4c306e8->"stringset.o"
9: intern ("test.err") returned 0x4c307b8->"test.err"
10: intern ("test.out") returned 0x4c30888->"test.out"
11: intern ("teststring") returned 0x4c30958->"teststring"
12: intern ("HEADER.html") returned 0x4c30158->"HEADER.html"
13: intern ("Makefile") returned 0x4c30228->"Makefile"
14: intern ("RCS") returned 0x4c302e8->"RCS"
15: intern ("main.cpp") returned 0x4c303b8->"main.cpp"
16: intern ("main.o") returned 0x4c30478->"main.o"
17: intern ("stringset.cpp") returned 0x4c30548->"stringset.cpp"
18: intern ("stringset.h") returned 0x4c30618->"stringset.h"
19: intern ("stringset.o") returned 0x4c306e8->"stringset.o"
20: intern ("test.err") returned 0x4c307b8->"test.err"
21: intern ("test.out") returned 0x4c30888->"test.out"
22: intern ("teststring") returned 0x4c30958->"teststring"
23: intern ("HEADER.html") returned 0x4c30158->"HEADER.html"
24: intern ("Makefile") returned 0x4c30228->"Makefile"
25: intern ("RCS") returned 0x4c302e8->"RCS"
26: intern ("main.cpp") returned 0x4c303b8->"main.cpp"
27: intern ("main.o") returned 0x4c30478->"main.o"
28: intern ("stringset.cpp") returned 0x4c30548->"stringset.cpp"
29: intern ("stringset.h") returned 0x4c30618->"stringset.h"
30: intern ("stringset.o") returned 0x4c306e8->"stringset.o"
31: intern ("test.err") returned 0x4c307b8->"test.err"
32: intern ("test.out") returned 0x4c30888->"test.out"
33: intern ("teststring") returned 0x4c30958->"teststring"
34: stringset[ 0]: 14383043818818809721 0x4c30958->"teststring"
35:                13646535705723827550 0x4c30158->"HEADER.html"
36: stringset[ 1]: 2099682443743551108 0x4c30478->"main.o"
37: stringset[ 4]: 994128771139992428 0x4c302e8->"RCS"
38: stringset[ 6]: 12311697016174216101 0x4c30548->"stringset.cpp"
39: stringset[10]: 8902767590177878864 0x4c30228->"Makefile"
40: stringset[11]: 12900681736301144686 0x4c306e8->"stringset.o"
41: stringset[12]: 11225442421023521097 0x4c30888->"test.out"
42: stringset[16]: 17041606903804112922 0x4c303b8->"main.cpp"
43: stringset[17]: 9633715851746635993 0x4c307b8->"test.err"
44: stringset[20]: 1356443961750385397 0x4c30618->"stringset.h"
45: load_factor = 0.478
46: bucket_count = 23
47: max_bucket_size = 2
```

```
1: ==27911== Memcheck, a memory error detector
2: ==27911== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al
.
3: ==27911== Using Valgrind-3.9.0 and LibVEX; rerun with -h for copyright i
nfo
4: ==27911== Command: teststring HEADER.html Makefile RCS main.cpp main.o s
tringset.cpp stringset.h stringset.o test.err test.out teststring HEADER.html M
akefile RCS main.cpp main.o stringset.cpp stringset.h stringset.o test.err test
.out teststring HEADER.html Makefile RCS main.cpp main.o stringset.cpp stringse
t.h stringset.o test.err test.out teststring
5: ==27911==
6: ==27911==
7: ==27911== HEAP SUMMARY:
8: ==27911==      in use at exit: 0 bytes in 0 blocks
9: ==27911==    total heap usage: 46 allocs, 46 frees, 1,652 bytes allocated
10: ==27911==
11: ==27911== All heap blocks were freed -- no leaks are possible
12: ==27911==
13: ==27911== For counts of detected and suppressed errors, rerun with: -v
14: ==27911== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 6 from 6)
```