

```
1: // $Id: string_set.h,v 1.2 2016-08-18 15:13:48-07 - - $
2:
3: #ifndef __STRING_SET__
4: #define __STRING_SET__
5:
6: #include <string>
7: #include <unordered_set>
8: using namespace std;
9:
10: #include <stdio.h>
11:
12: struct string_set {
13:     string_set();
14:     static unordered_set<string> set;
15:     static const string* intern (const char*);
16:     static void dump (FILE*);
17: };
18:
19: #endif
20:
```

```
1: // $Id: string_set.cpp,v 1.4 2016-09-21 16:56:20-07 - - $
2:
3: #include <string>
4: #include <unordered_set>
5: using namespace std;
6:
7: #include "string_set.h"
8:
9: unordered_set<string> string_set::set;
10:
11: string_set::string_set() {
12:     set.max_load_factor (0.5);
13: }
14:
15: const string* string_set::intern (const char* string) {
16:     auto handle = set.insert (string);
17:     return &*handle.first;
18: }
19:
20: void string_set::dump (FILE* out) {
21:     static unordered_set<string>::hasher hash_fn
22:         = string_set::set.hash_function();
23:     size_t max_bucket_size = 0;
24:     for (size_t bucket = 0; bucket < set.bucket_count(); ++bucket) {
25:         bool need_index = true;
26:         size_t curr_size = set.bucket_size (bucket);
27:         if (max_bucket_size < curr_size) max_bucket_size = curr_size;
28:         for (auto itor = set.cbegin (bucket);
29:              itor != set.cend (bucket); ++itor) {
30:             if (need_index) fprintf (out, "string_set[%4zu]: ", bucket);
31:             else fprintf (out, "          %4s  ", "");
32:             need_index = false;
33:             const string* str = &*itor;
34:             fprintf (out, "%22zu %p->\"%s\"\\n", hash_fn(*str),
35:                     str, str->c_str());
36:         }
37:     }
38:     fprintf (out, "load_factor = %.3f\\n", set.load_factor());
39:     fprintf (out, "bucket_count = %zu\\n", set.bucket_count());
40:     fprintf (out, "max_bucket_size = %zu\\n", max_bucket_size);
41: }
42:
```

```
1: // $Id: main.cpp,v 1.2 2016-08-18 15:13:48-07 - - $
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 "string_set.h"
12:
13: int main (int argc, char** argv) {
14:     for (int i = 1; i < argc; ++i) {
15:         const string* str = string_set::intern (argv[i]);
16:         printf ("intern (\"%s\") returned %p->\"%s\"\\n",
17:             argv[i], str, str->c_str());
18:     }
19:     string_set::dump (stdout);
20:     return EXIT_SUCCESS;
21: }
22:
```

```
1: # $Id: Makefile,v 1.4 2016-09-21 16:56:20-07 - - $
2:
3: GPP      = g++ -std=gnu++14 -g -O0 -Wall -Wextra
4: MKDEP    = g++ -std=gnu++14 -MM
5:
6: MKFILE    = Makefile
7: DEPFILe   = Makefile.dep
8: SOURCES   = string_set.cpp main.cpp
9: HEADERS   = string_set.h
10: OBJECTS   = ${SOURCES:.cpp=.o}
11: EXECBIN   = test_string_set
12: SRCFILES  = ${HEADERS} ${SOURCES} ${MKFILE}
13:
14: all : ${EXECBIN}
15:
16: ${EXECBIN} : ${OBJECTS}
17:             ${GPP} ${OBJECTS} -o ${EXECBIN}
18:
19: %.o : %.cpp
20:             ${GPP} -c $<
21:
22: ci :
23:         cid + ${SRCFILES}
24:
25: clean :
26:         -rm ${OBJECTS} ${DEPFILe}
27:
28: spotless : clean
29:         - rm ${EXECBIN} Listing.ps Listing.pdf test.out
30:
31: ${DEPFILe} :
32:         ${MKDEP} ${SOURCES} >${DEPFILe}
33:
34: dep :
35:         - rm ${DEPFILe}
36:         ${MAKE} --no-print-directory ${DEPFILe}
37:
38: include ${DEPFILe}
39:
40: test : ${EXECBIN}
41:         ${EXECBIN} * * * >test.out 2>&1
42:
43: lis : test
44:         mkpspdf Listing.ps ${SRCFILES} ${DEPFILe} test.out
45:
46: again : ${SRCFILES}
47:         make --no-print-directory spotless dep ci test lis
```

09/21/16
16:56:20

\$cmpps104a-wm/Assignments/code/string_set/
Makefile.dep

1/1

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

```
1: intern ("HEADER.html") returned 0x24af048->"HEADER.html"
2: intern ("Makefile") returned 0x24af0b8->"Makefile"
3: intern ("Makefile.dep") returned 0x24af068->"Makefile.dep"
4: intern ("RCS") returned 0x24af168->"RCS"
5: intern ("main.cpp") returned 0x24af1b8->"main.cpp"
6: intern ("main.o") returned 0x24af238->"main.o"
7: intern ("string_set.cpp") returned 0x24af288->"string_set.cpp"
8: intern ("string_set.h") returned 0x24af2d8->"string_set.h"
9: intern ("string_set.o") returned 0x24af328->"string_set.o"
10: intern ("test.err") returned 0x24af378->"test.err"
11: intern ("test_string_set") returned 0x24af3c8->"test_string_set"
12: intern ("HEADER.html") returned 0x24af048->"HEADER.html"
13: intern ("Makefile") returned 0x24af0b8->"Makefile"
14: intern ("Makefile.dep") returned 0x24af068->"Makefile.dep"
15: intern ("RCS") returned 0x24af168->"RCS"
16: intern ("main.cpp") returned 0x24af1b8->"main.cpp"
17: intern ("main.o") returned 0x24af238->"main.o"
18: intern ("string_set.cpp") returned 0x24af288->"string_set.cpp"
19: intern ("string_set.h") returned 0x24af2d8->"string_set.h"
20: intern ("string_set.o") returned 0x24af328->"string_set.o"
21: intern ("test.err") returned 0x24af378->"test.err"
22: intern ("test_string_set") returned 0x24af3c8->"test_string_set"
23: intern ("HEADER.html") returned 0x24af048->"HEADER.html"
24: intern ("Makefile") returned 0x24af0b8->"Makefile"
25: intern ("Makefile.dep") returned 0x24af068->"Makefile.dep"
26: intern ("RCS") returned 0x24af168->"RCS"
27: intern ("main.cpp") returned 0x24af1b8->"main.cpp"
28: intern ("main.o") returned 0x24af238->"main.o"
29: intern ("string_set.cpp") returned 0x24af288->"string_set.cpp"
30: intern ("string_set.h") returned 0x24af2d8->"string_set.h"
31: intern ("string_set.o") returned 0x24af328->"string_set.o"
32: intern ("test.err") returned 0x24af378->"test.err"
33: intern ("test_string_set") returned 0x24af3c8->"test_string_set"
34: string_set[ 0]: 13646535705723827550 0x24af048->"HEADER.html"
35:                7517842887488357474 0x24af2d8->"string_set.h"
36: string_set[ 1]: 2099682443743551108 0x24af238->"main.o"
37: string_set[ 2]: 15286446792580072886 0x24af288->"string_set.cpp"
38: string_set[ 4]: 994128771139992428 0x24af168->"RCS"
39: string_set[ 7]: 9799794095794511326 0x24af3c8->"test_string_set"
40:                1516108490113098673 0x24af328->"string_set.o"
41: string_set[ 10]: 8902767590177878864 0x24af0b8->"Makefile"
42: string_set[ 16]: 17041606903804112922 0x24af1b8->"main.cpp"
43: string_set[ 17]: 9633715851746635993 0x24af378->"test.err"
44: string_set[ 22]: 2246613038755228464 0x24af068->"Makefile.dep"
45: load_factor = 0.478
46: bucket_count = 23
47: max_bucket_size = 2
```