```
 1: // $Id: ristring.cpp,v 1.1 2021-04-25 14:10:20-07 - - $
 2:
 3: //
 4: // NAME
 5: //    ristring - reference counted immutable string
 6: //
 7: // DESCRIPTION
 8: //    Shows how to use reference counting on immutable objects.
 9: //    If this were changed to mutable, then we should probably
10: //    implement them as copy-on-write (COW).
11: //
12:
13: #include <cstdlib>
14: #include <cstring>
15: #include <iostream>
16:
17: using namespace std;
18:
19: //////////////////////////////////////////////////////////////////
20: // ristring.h
21: //////////////////////////////////////////////////////////////////
22:
23: class ristring {
24:    private:
25:       class repr_t;
26:       repr_t *repr;
27:       ristring () = delete;
28:       void decrement ();
29:    public:
30:       // override implicit members
31:       ristring (const ristring &);              // copy ctor
32:       ristring &operator= (const ristring &);  // operator=
33:       ~ristring ();                             // dtor
34:       // other members
35:       ristring (const char *const);             // "" ctor
36:       ristring &operator= (const char *const); // "" operator=
37:       char operator[] (int index) const;       // charat
38:       int size () const;                        // strlen
39:       friend ostream &operator<< (ostream &, const ristring &);
40:       ostream &show (ostream &, const string &label);
41: };
42:
43: class ristring::repr_t {
44:    friend class ristring;
45:    private:
46:       int ref_count;
47:       const ssize_t isize;
48:       const char *const buffer;
49:       // Default members.
50:       repr_t () = delete;
51:       repr_t (const repr_t &) = delete;
52:       repr_t &operator= (const repr_t &) = delete;
53:       ~repr_t ();
54:       // Ctor and fields.
55:       repr_t (const char *const string);
56:       friend ostream &operator<< (ostream &, const ristring &);
57: };
```

```
 58:
 59: //////////////////////////////////////////////////////////////////
 60: // ristring.cpp
 61: //////////////////////////////////////////////////////////////////
 62:
 63: // strdup(3) calls malloc(3), which is to be freed with free(3),
 64: // not with delete[].  Do not mix malloc/new with free/delete.
 65: const char *strnew (const char *const str) {
 66:    char *tmp = new char[strlen (str) + 1];
 67:    strcpy (tmp, str);
 68:    return tmp;
 69: }
 70:
 71: ristring::ristring (const ristring &that) {
 72:    repr = that.repr;
 73:    ++repr->ref_count;
 74: }
 75:
 76: ristring &ristring::operator= (const ristring &that) {
 77:    if (this != &that) {
 78:       decrement ();
 79:       repr = that.repr;
 80:       ++repr->ref_count;
 81:    }
 82:    return *this;
 83: }
 84:
 85: ristring::ristring (const char *const that) {
 86:    repr = new repr_t (that);
 87: }
 88:
 89: ristring::~ristring () {
 90:    decrement ();
 91: }
 92:
 93: char ristring::operator[] (int index) const {
 94:    return repr->buffer[index];
 95: }
 96:
 97: int ristring::size () const {
 98:    return repr->isize;
 99: }
100:
101: void ristring::decrement () {
102:    --repr->ref_count;
103:    if (repr->ref_count == 0) delete repr;
104: }
105:
106: ostream &ristring::show (ostream &out, const string &label) {
107:    out << label << ": " << static_cast <const void*> (this)
108:        << "->irstring {repr=" << repr
109:        << "-> {" << endl
110:        << "   ref_count=" << repr->ref_count
111:        << ", isize=" << repr->isize
112:        << ", buffer=" << static_cast <const void*> (repr->buffer)
113:        << "->\"" << repr->buffer << "\"" << endl
114:        << "}}" << endl;
115:    return out;
```

```
116: }
117:
```

```
118:
119: ristring::repr_t::repr_t (const char *const string):
120:       ref_count (1), isize (strlen (string)), buffer (strnew (string)) {
121: }
122:
123: ristring::repr_t::~repr_t () {
124:    delete[] buffer;
125: }
126:
127: ostream &operator<< (ostream &out, const ristring &that) {
128:    out << that.repr->buffer;
129:    return out;
130: }
131:
132: //////////////////////////////////////////////////////////////////
133: // main.cpp
134: //////////////////////////////////////////////////////////////////
135:
136: int main (int argc, char **argv) {
137:    cout << argv[0] << " " << argc << endl;
138:    ristring first = "Hello, world!";
139:    first.show (cout, "first") << endl;
140:    cout << first << endl;
141:    for (int index = 0; index < first.size (); ++index) {
142:       cout << "|" << first[index];
143:    }
144:    cout << "|" << endl;
145:    ristring second = "foobar";
146:    second.show (cout, "second") << endl;
147:    second = first;
148:    ristring third = first;
149:    ristring fourth (first);
150:    cout << second << endl;
151:    cout << third << endl;
152:    cout << fourth << endl;
153:    second.show (cout, "fourth") << endl;
154:    return EXIT_SUCCESS;
155: }
156:
157: /*
158: //TEST// valgrind --leak-check=full --show-reachable=yes \
159: //TEST//       --log-file=ristring.out.grind \
160: //TEST//       ristring >ristring.out 2>&1
161: //TEST// mkpspdf ristring.ps ristring.cpp* ristring.out*
162: */
163:
```

```
    1: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting ristring.cpp
    2: $Id: ristring.cpp,v 1.1 2021-04-25 14:10:20-07 - - $
    3: checksource ristring.cpp
    4: checksource: TOTAL COMPLAINTS: 0
    5: ristring.cpp: $Id: ristring.cpp,v 1.1 2021-04-25 14:10:20-07 - - $
    6: cpplint.py.perl ristring.cpp
    7: Done processing ristring.cpp
    8: g++ -Wall -Wextra -Wpedantic -Wshadow -fdiagnostics-color=never -std=gnu
++17 -Wold-style-cast -pthread -g -O0 ristring.cpp -o ristring -lm
    9: rm -f ristring.o
   10: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished ristring.cpp
```

```
ristring 1
first: 0x1ffefff8b8->irstring {repr=0x5c41040-> {
   ref_count=1, isize=13, buffer=0x5c410a0->"Hello, world!"
}}

Hello, world!
|H|e|l|l|o|,| |w|o|r|l|d|!|
second: 0x1ffefff8b0->irstring {repr=0x5c41150-> {
   ref_count=1, isize=6, buffer=0x5c411b0->"foobar"
}}

Hello, world!
Hello, world!
Hello, world!
fourth: 0x1ffefff8b0->irstring {repr=0x5c41040-> {
   ref_count=4, isize=13, buffer=0x5c410a0->"Hello, world!"
}}

```

```
    1: ==13809== Memcheck, a memory error detector
    2: ==13809== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al
.
    3: ==13809== Using Valgrind-3.14.0 and LibVEX; rerun with -h for copyright
info
    4: ==13809== Command: ristring
    5: ==13809== Parent PID: 13808
    6: ==13809==
    7: ==13809==
    8: ==13809== HEAP SUMMARY:
    9: ==13809==     in use at exit: 0 bytes in 0 blocks
   10: ==13809==   total heap usage: 7 allocs, 7 frees, 161 bytes allocated
   11: ==13809==
   12: ==13809== All heap blocks were freed -- no leaks are possible
   13: ==13809==
   14: ==13809== For counts of detected and suppressed errors, rerun with: -v
   15: ==13809== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```