code/conc/sharing.c

```
1   #include "csapp.h"
2   #define N 2
3   void *thread(void *vargp);
4
5   char **ptr;  /* Global variable */
6
7   int main()
8   {
9       int i;
10      pthread_t tid;
11      char *msgs[N] = {
12          "Hello from foo",
13          "Hello from bar"
14      };
15
16      ptr = msgs;
17      for (i = 0; i < N; i++)
18          Pthread_create(&tid, NULL, thread, (void *)i);
19      Pthread_exit(NULL);
20  }
21
22  void *thread(void *vargp)
23  {
24      int myid = (int)vargp;
25      static int cnt = 0;
26      printf("[%d]: %s (cnt=%d)\n", myid, ptr[myid], ++cnt);
27      return NULL;
28  }
```

code/conc/sharing.c

图 12-15　说明共享不同方面的示例程序

12.4.2　将变量映射到内存

多线程的 C 程序中变量根据它们的存储类型被映射到虚拟内存：

- *全局变量*。全局变量是定义在函数之外的变量。在运行时，虚拟内存的读/写区域只包含每个全局变量的一个实例，任何线程都可以引用。例如，第 5 行声明的全局变量 ptr 在虚拟内存的读/写区域中有一个运行时实例。当一个变量只有一个实例时，我们只用变量名(在这里就是 ptr)来表示这个实例。
- *本地自动变量*。本地自动变量就是定义在函数内部但是没有 static 属性的变量。在运行时，每个线程的栈都包含它自己的所有本地自动变量的实例。即使多个线程执行同一个线程例程时也是如此。例如，有一个本地变量 tid 的实例，它保存在主线程的栈中。我们用 tid.m 来表示这个实例。再来看一个例子，本地变量 myid 有两个实例，一个在对等线程 0 的栈内，另一个在对等线程 1 的栈内。我们将这两个实例分别表示为 myid.p0 和 myid.p1。
- *本地静态变量*。本地静态变量是定义在函数内部并有 static 属性的变量。和全局变量一样，虚拟内存的读/写区域只包含在程序中声明的每个本地静态变量的一个实例。例如，即使示例程序中的每个对等线程都在第 25 行声明了 cnt，在运行时，虚拟内存的读/写区域中也只有一个 cnt 的实例。每个对等线程都读和写这个实例。