

Aplicação Cliente/Servidor usando MyRPC com *Pooling* em C

Invoker

```
1  static void process_req(struct req *req) {
2      /* ... */
3
4      if (/* ... */) {
5          inst = service_get_instance(req->invoker->service);
6          func = service_get_method(inst, method);
7
8          if (func != NULL) {
9              /* ... */
10
11              func(request, reply);
12
13              /* ... */
14          }
15
16          service_release_instance(req->invoker->service, inst);
17      }
18
19      /* ... */
20 }
```

Service

```
1  struct service_instance *service_get_instance(struct service *const service) {
2      pthread_mutex_lock(&service->instances_mutex);
3
4      while (service->instances_count == 0)
5          pthread_cond_wait(&service->instances_cond, &service->instances_mutex);
6
7      uint_8 i = --service->instances_count;
8      struct service_instance *service_instance = service->instances[i];
9      service->instances[i] = NULL;
10
11     pthread_mutex_unlock(&service->instances_mutex);
12
13     return service_instance;
14 }
15
16 void service_release_instance(struct service *const service, struct service_instance *const service_instance) {
17     pthread_mutex_lock(&service->instances_mutex);
18
19     service->instances[service->instances_count++] = service_instance;
20
21     pthread_cond_signal(&service->instances_cond);
22     pthread_mutex_unlock(&service->instances_mutex);
23 }
```

Avaliação comparativa de desempenho

