
socket 应用编程实验

熊子威

2015K8009915050

1 实验内容

本实验目标是构建一个基于 socket 的分布式字符统计程序并在 mininet 中运行这个程序。程序分为 master 和 worker 两个部分。master 通过 socket 接口向两个 worker 分发任务（需要统计字符数的文件路径，该文件中那个部分需要当前 worker 去统计），而两个 worker 则通过 socket 接口接受来自 master 的任务分配然后进行字母数目的统计，统计完毕后将二十六个字母的数目再各自发回给 master，然后 master 将两者的统计数据加和并打印出来。

2 实验流程

因为第一次使用 socket 并进行网络编程，对各个 API 的工作方式不是很了解，因此写代码时是一边测试一边迭代进行的。

首先编写一个简单的 master 和 worker，确保 master 能够和一个 worker 连接上，然后这个 worker 发送一条打招呼信息，然后让 worker 将这条信息打印出来。

然后修改这个程序，添加多线程，让 master 同时和两个 worker 打招呼，确认 listen() 这个接口是否只需要主线程监听即可。结果是只需要主线程监听，两个工作线程负责 accept 然后处理数据传输和处理即可。

然后修改这个程序，让 master 按照顺序分别向两个 worker 发送两条消息，确认 socket 对消息发送和接受的顺序是如何处理的。然后发现 send 和 recv 是匹配在一起不会发生乱序之后，进一步修改程序。

这一步的修改是为 mastert 添加计算文件总长度并依据 worker 的不同地址而分别分配不同的任务并且让 worker 打印出分配来顺便检查工作分配是否正确（文件名是否正确接受，文本划分的起止位置是否正确）。

这一步确保能够正确工作之后为 worker 添加字符统计的代码，并分别打印结果来确认统计结果正确。

最后一步为 master 和 worker 分别添加接收统计结果和发送统计结果的代码，并为 master 添加汇总结果的代码。

全部完成并却运转正确，和老师的 reference 输出一致之后，重新修改了 master 代码的结构，将一些复杂冗长的操作用函数封装，提高代码的可读性。

3 实验结果

起始阶段：

```

XTerm
Node: h1
root@Folio:~/Course/Networks/Lab/03/03-socket# ./master war and peace.txt
Workers conf:
10.0.0.2
10.0.0.3
Socket created
bind done
Waiting for incoming connections...

Node: h2
root@Folio:~/Course/Networks/Lab/03/03-socket#
  
```

执行结束：

```

XTerm
Node: h1
d: 118298
e: 313575
f: 54901
g: 51327
h: 167415
i: 172257
j: 2574
k: 20432
l: 96532
m: 61649
n: 184184
o: 190083
p: 45533
q: 2331
r: 148431
s: 162897
t: 226414
u: 64399
v: 27087
w: 59209
x: 4384
y: 46235
z: 2388
root@Folio:~/Course/Networks/Lab/03/03-socket#

Node: h2
d: 59337
e: 159494
f: 28956
g: 24997
h: 84350
i: 85794
j: 1265
k: 9497
l: 48002
m: 31181
n: 92888
o: 96756
p: 23441
q: 1178
r: 74738
s: 82121
t: 116465
u: 32084
v: 13142
w: 29662
x: 2153
y: 23106
z: 1250
root@Folio:~/Course/Networks/Lab/03/03-socket#
  
```

可以看到两个 worker 成功并且正确地向 master 发送了相关的数据，而 master 也成功接受并且完成了

最后的结果汇总。

4 结果分析

可以从 master 的主要代码来分析

```

32 int main(int argc, char *argv[]) {
33     if(argc != 2) {
34         printf("You have to input one and only one file name\n");
35         return -1;
36     }
37
38     if(get_conf(argv[1]) != 0) {
39         return -1;
40     }
41
42     // setup a server
43     if(setup_server() != 0) {
44         printf("Server setup failed\n");
45         return -1;
46     }
47
48     // main work
49     if(main_work(argv[1]) != 0)
50         return -1;
51
52     show_results();
53     return 0;
54 }

```

```

172
173 void* handle(void* in) {
174     int cs;
175     struct sockaddr_in client;
176
177     accept_and_check(&cs, &client);
178
179     // dispatch some work
180     puts("dispatching...");
181     dispatch(cs, &client, (char *)in);
182
183     puts("confirming...");
184     // receiving confirmation of start of counting
185     if(confirm(cs, &client))
186         return NULL;
187
188     puts("receiving...");
189     // receiving counting result
190     if(receive(cs, &client) != 0)
191         return NULL;
192     return in;
193 }
194
195 int main_work(char *in) {
196     pthread_t p1;
197     pthread_t p2;
198     void *status = NULL;
199     pthread_create(&p1, NULL, handle, in);
200     pthread_create(&p2, NULL, handle, in);
201
202     pthread_join(p1, &status);
203     pthread_join(p2, &status);
204
205     if(status == NULL)
206         return -1;
207     return 0;
208 }
209

```

master 进行了几个阶段的确认，建立 server 时，分配任务时，确认任务分发成功时，最后获得结果时，都需要确认成功才能进行下一步。相应的，worker 也需要进行响应。因此最后的结果如下图所示：

```

03-socket git:(master) X wps 00-实验报告模板.docx
x - o "Node: h1"
bind done
Waiting for incoming connections...

Connection accepted
Client address is 10.0.0.3
dispatching...
Server will send '17' to clients
Server will send 'war and peace.txt' to clients
start and end point for 10.0.0.3 is 1679776 and 3359550
confirming...
Confirmation of 10.0.0.3 received: Client is ready
P1
receiving...
Connection accepted
Client address is 10.0.0.2
dispatching...
Server will send '17' to clients
Server will send 'war and peace.txt' to clients
start and end point for 10.0.0.2 is 0 and 1679775
confirming...
Confirmation of 10.0.0.2 received: Client is ready
P2
receiving...
results from two clients are:

root@Folio:~/Course/Networks/Lab/03/03-socket# ./worker
Socket created
Connected
Server's reply: Hello, this is a reply from server
Message length is: 17
file path length is 17, file path is war and peace.txt
partition is 1679776 and 3359550
Start counting
Count result
a: 101487
b: 17620
c: 32273
d: 59337
e: 159494
f: 28956
g: 34007

root@Folio:~/Course/Networks/Lab/03/03-socket# ./worker
Socket created
Connected
Server's reply: Hello, this is a reply from server
Message length is: 17
file path length is 17, file path is war and peace.txt
partition is 0 and 1679775
Start counting
Count result
a: 101230
b: 17038
c: 29349
d: 58961
e: 154081

```