

Socket 应用编程实验

孙佳钰 2015K8009929051

2018 年 9 月 27 日

1 实验内容

实验有一个 master 和三个 worker，采用 socket API，目的是统计文件中英文字符的个数。

master 与 worker 建立连接后，将统计任务分给所有的 worker。收到 worker 返回的结果后进行汇总输出。

worker 受到任务后解析并读取文件，开始统计文件中的英文字符个数，并把结果返回给 master。

2 实验流程

1. worker 建立 socket，与监听地址绑定并进行监听。
2. master 建立三个 socket，分别与 worker 申请建立连接。
3. worker 接受连接请求。
4. master 产生命令数据包，分别发送至 worker，然后等待回复。

```
1 for (int i = 0; i < 3; i++) {
2     int tmp;
3     tmp = htonl(frlen);
4     memcpy(message[i], &tmp, 4);
5     tmp = htonl(split * i);
6     memcpy(message[i] + 4, &tmp, 4);
7     if (i == 2) {
8         tmp = htonl(size);
9     } else {
10        tmp = htonl(split * (i + 1) - 1);
11    }
12    memcpy(message[i] + 8, &tmp, 4);
```

```
13     memcpy(message[i] + 12, fn, fnlen);
14
15     // send the orders to the workers
16     if (send(s[i], message[i], 31, 0) < 0) {
17         printf("send %d failed\n", i);
18         return 1;
19     }
20 }
```

5. worker 收到命令数据包后，进行解析并开始工作，然后将结果返回给 master。

```
1 while ((msg_len = recv(cs, msg, sizeof(msg), 0)) > 0) {
2     // work and send the result back to master
3     int len = ntohl(((int*)msg)[0]);
4     char sted[8];
5     memcpy(sted, msg + 4, 8);
6     int start = ntohl(((int*)sted)[0]);
7     int length = ntohl(((int*)sted)[1]) - start + 1;
8     printf("start = %d, length = %d\n", start, length);
9     char *fn = malloc(len * sizeof(char));
10    memcpy(fn, msg + 12, len);
11    printf("fn = %s\n", fn);
12    FILE *fp = fopen(fn, "r");
13    fseek(fp, start, SEEK_SET);
14    int ires[26] = {0};
15    for (int i = 0; i < length; i++) {
16        char c = fgetc(fp);
17        if (c >= 'a' && c <= 'z')
18            ires[c - 'a']++;
19        if (c >= 'A' && c <= 'Z')
20            ires[c - 'A']++;
21    }
22    fclose(fp);
23    void *cres = malloc(104);
24    int tmp;
25    for (int i = 0; i < 26; i++) {
26        printf("%c, %d\n", 'a' + i, ires[i]);
27        tmp = htonl(ires[i]);
28        memcpy(cres + i * 4, &tmp, 4);
29    }
```

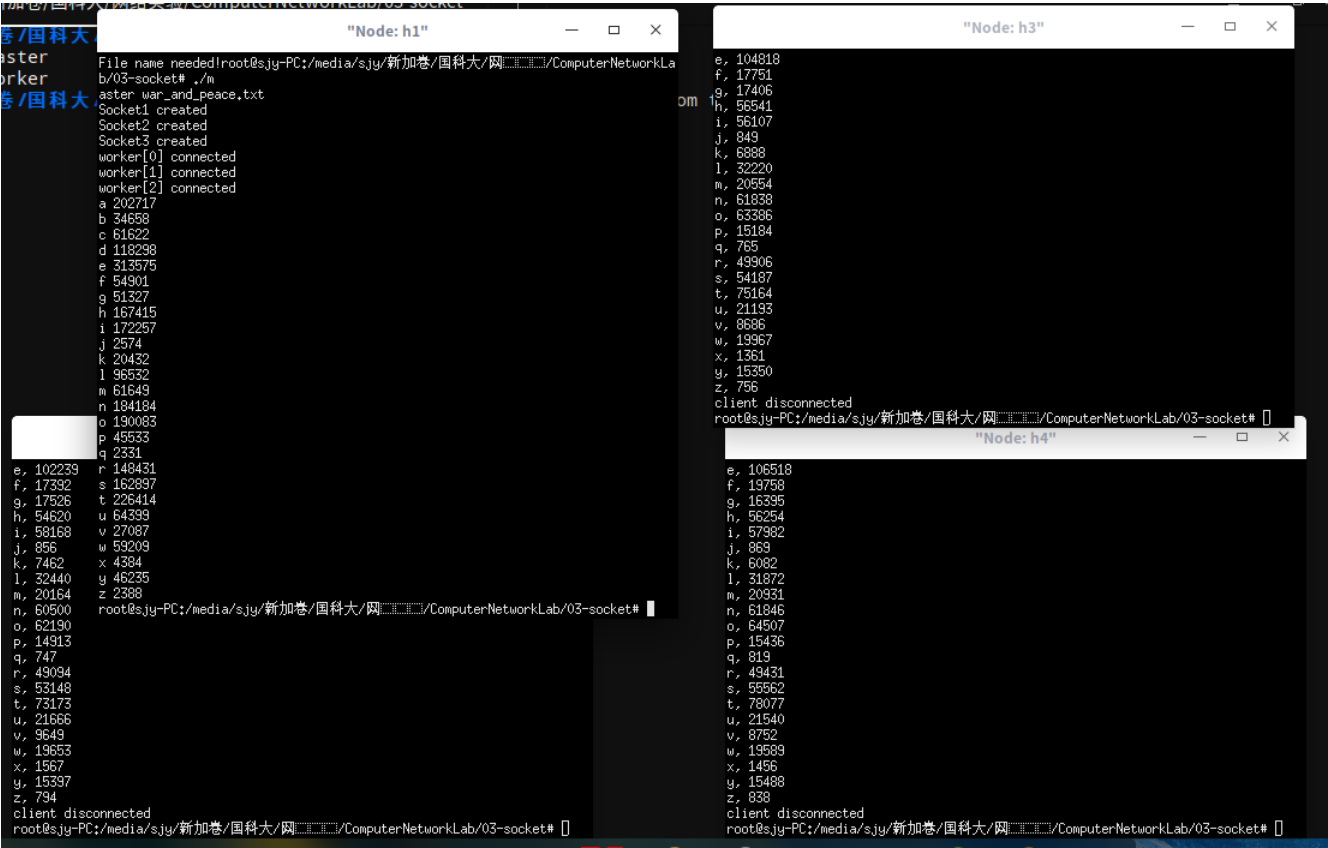
```
29     }  
30     write(cs, cres, 104);  
31 }
```

6. master 收到 worker 返回的结果后进行汇总并输出。

```
1  for (int i = 0; i < 3; i++) {  
2      // receive replies from the workers  
3      if (recv(s[i], worker_reply[i], 104, 0) < 0) {  
4          printf("recv %d failed\n", i);  
5          continue;  
6      }  
7  }  
8  
9  for (int i = 0; i < 26; i++) {  
10     int res = 0;  
11     res += ntohl(((int*)worker_reply[0])[i]);  
12     res += ntohl(((int*)worker_reply[1])[i]);  
13     res += ntohl(((int*)worker_reply[2])[i]);  
14     printf("%c %d\n", 'a' + i, res);  
15 }
```

完整实验代码可见所附文件。

3 实验结果及分析



其中 h1 运行的是 master，输出的是总结果。h2 到 h4 运行的是 worker，输出的是各自统计的结果。