

## 效果

---

```
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
Hello, world
```

## 结论

---

- 练习 Socket 编程
  - 一个进程利用 send 函数发信息
  - 一个进程利用 recv 接收信息并显示

虽然可以用 fork，但还是开两个 shell 分别运行 send 和 recv 效果更好

## 代码

---

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <stdio.h>

void sender()
{
    int sockfd;
    struct sockaddr_in serv_addr;
```

```
sockfd = socket(AF_INET, SOCK_STREAM, 0);
bzero(&serv_addr, sizeof(serv_addr));

serv_addr.sin_family = AF_INET;
serv_addr.sin_port = htons(6000);
serv_addr.sin_addr.s_addr = inet_addr("127.0.0.1");

connect(sockfd, (struct sockaddr *)&serv_addr, sizeof(serv_addr));
int num = 0;
while (1)
{
    char buffer[256];
    bzero(buffer, 256);
    sprintf(buffer, "%d", num++);
    sleep(1);
    write(sockfd, buffer, strlen(buffer));
}

close(sockfd);
}

void receiver()
{
    int sockfd, newsockfd;
    struct sockaddr_in serv_addr, cli_addr;
    socklen_t clilen;
    char buffer[256];

    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    bzero((char *)&serv_addr, sizeof(serv_addr));

    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(6000);
    serv_addr.sin_addr.s_addr = INADDR_ANY;

    bind(sockfd, (struct sockaddr *)&serv_addr, sizeof(serv_addr));

    listen(sockfd, 5);

    clilen = sizeof(cli_addr);

    newsockfd = accept(sockfd, (struct sockaddr *)&cli_addr, &clilen);

    while (1)
    {
        bzero(buffer, 256);
        read(newsockfd, buffer, 255);
        printf("%s\n", buffer);
    }

    close(newsockfd);
    close(sockfd);
}
```

```
int main()
{
    if (fork())
    {
        sender();
    }
    else
    {
        receiver();
    }
    return 0;
}
```

## maybe 前端

```
ventus@ventus-virtual-machine: /Desktop/code/os/os_homework/socket$ ./my_back
HTTP/1.1 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Mon, 09 Oct 2023 18:46:52 GMT
Content-Type: text/html
Content-Length: 872
Last-Modified: Fri, 25 Aug 2023 02:03:32 GMT
Connection: keep-alive
ETag: "64e80bf4-368"
Accept-Ranges: bytes

<!DOCTYPE html><html><head><title>Quasar App</title><meta charset=utf-8><meta name=description content="A Quasar Project"><meta nam
e=format-detection content="telephone=no"><meta name=msapplication-tap-highlight content=no><meta name=viewport content="user-scala
ble=no, initial-scale=1, maximum-scale=1, minimum-scale=1, width=device-width"><link rel=icon type=image/png sizes=128x128 href="/icons
/favicon-128x128.png"><link rel=icon type=image/png sizes=96x96 href="/icons/favicon-96x96.png"><link rel=icon type=image/png sizes
=32x32 href="/icons/favicon-32x32.png"><link rel=icon type=image/png sizes=16x16 href="/icons/favicon-16x16.png"><link rel=icon typ
e=image/ico href="/favicon.ico"> <script type="module" crossorigin src="/assets/index.e33ee310.js"></script>
<link rel="stylesheet" href="/assets/index.cf738233.css">
</head><body><div id=q-app></div></body></html>
```