

راهنمای سوال اول و دوم از تمرین سوم

```
while(1)
{
    printf("\n+++++++ Waiting for new connection ++++++\n\n");
    if ((new_socket = accept(server_fd, (struct sockaddr *)&address, (socklen_t*)&addrlen))<0)
    {
        perror("In accept");
        exit(EXIT_FAILURE);
    }

    char buffer[30000] = {0};
    valread = read( new_socket , buffer, 30000);
    printf("%s\n",buffer );
    // uncomment following line and connect many clients
    // sleep(5);
    write(new_socket , hello , strlen(hello));
    printf("-----Hello message sent-----");
    close(new_socket);
}
return 0;
```

```
printf("\n++++++ Waiting for new connection ++++++\n\n");
if ((new_socket = accept(server_fd, (struct sockaddr *)&address, (socklen_t*)&addrlen))<0)
{
    perror("In accept");
    exit(EXIT_FAILURE);
}
```

در هنگام فراخوانی تابع `accept` سرور منتظر می‌شود تا یک درخواست از سمت کلاینت دریافت شود.

بعد از دریافت یک آبجکت از جنس `socket` بدست می‌آید که راه ارتباطی سرور با کلاینت است. در شکل این سوکت در متغیر `new_socket` ذخیره شده است. برای دریافت اطلاعات از کلاینت از تابع `read` استفاده می‌شود و برای ارسال اطلاعات از تابع `write` استفاده می‌شود.

```
char buffer[30000] = {0};
valread = read( new_socket , buffer, 30000);
printf("%s\n",buffer );
// uncomment following line and connect many clients
// sleep(5);
write(new_socket , hello , strlen(hello));
printf("-----Hello message sent-----");
close(new_socket);
```

در این قسمت یک **buffer** برای پردازش پیام کاربر تعریف شده است. بعد مقداری اطلاعات از سمت کاربر خوانده می شود و در بافر تعریف شده کپی می شود. سپس سرور بدون توجه به محتوای پیام پیغام ذخیره شده در متغیر **hello** را برای کلاینت ارسال می کند. برای ارسال از تابع **write** استفاده شده است. در انتها **socket** میان کاربر و سرور بسته می شود و کار سرویس دادن به کاربر به پایان می رسد.

شما چه کار باید بکنید؟ (سوال اول)

پس از دریافت درخواست از سمت کاربر، شما باید یک پردازش جدید ایجاد کنید و سوکت ارتباطی میان سرور و کلاینت را در دسترس پردازش جدید قرار دهید.

پردازش پدر که حالا کار سرویس را به فرزند خود سپرده است. سوکت ارتباطی خودش را **close** می‌کند و منتظر دریافت درخواست کاربر بعدی می‌ماند.

پردازش فرزند بعد از اتمام سرویس می‌تواند کار خود را پایان دهد.

شما چه کار باید بکنید؟ (سوال دوم)

پس از دریافت درخواست از سمت کاربر، شما باید یک ترد جدید ایجاد کنید و سوکت ارتباطی میان سرور و کلاینت را در دسترس ترد جدید قرار دهید.

پردازه پدر که حالا کار سرویس را به ترد کارگر خود سپرده است، منتظر دریافت درخواست کاربر بعدی می ماند.

ترد بعد از اتمام سرویس می تواند کار خود را پایان دهد.