Atividade 6 - Servidor TCP com

Gabriel Hidasy Rezende

October 21, 2015

1 Questão 1:

O servidor em questão nunca aceitou conexões concorrentes, mas dava a impressão de o fazer já que a tarefa executada por cliente era realizada muito rapido e a conexão encerrada em sequencia.

A mudança deixou o processamento por cliente mais lento, evidenciando o carater serial do servidor, quando dois clientes se conectam ao mesmo tempo o primeiro recebe a resposta enquanto o segundo espera o sleep terminar antes de receber dados.

Ambas as conexões são bem sucedidas já que temos um buffer (de tamanho 10, definido em `listen(sockfd,buffer_size)` para coneções ainda não aceitas.

Mas fazendo um numero grande de conexões num curto intervalo de tempo começamos a ter falhas de conexão

2 Questão 2

Utilizei os programas fornecidos como base, eles seguem anexos em outro pdf

3 Questão 3

Ainda utilizando os mesmos como base, seguem anexos em pfg

4 Questão 4

Porque o no trecho apresentado o processo pai (que roda o Accept) fecha o socket usado para se comunicar com o cliente, mas o filho permanece com ele aberto até terminar sua operação (uma escolha sensivel dado que o filho

trata da conexão do cliente). Da mesma forma o filho fecha o socket de listen (que ele não vai usar), mas o pai permance com ele aberto, afinal ele trata de receber mais clientes.

5 Questão 5

Após executar o servidor iniciei 3 conexões, a saida de pstree -A o seguinte trecho:

```
|-xfce4-terminal-+-zsh---pstree
                 |-zsh-+-3*[cliente]
                       |-servidor---3*[servidor]
```

O que claramente indica que o servidor que eu rodei no meu terminal tem 3 filhos que são instancias dele mesmo, como esperado pelos multiplos fork

O mesmo pode ser visto na saida do `ps o pid,ppid,command`

```
PID    PPID COMMAND
16267  3765 zsh
17074 16267 ./servidor
17118 16267 ./cliente 127.0.0.1 12344
17119 17074 ./servidor
17229 16267 ./cliente 127.0.0.1 12344
17230 17074 ./servidor
17676 16267 ./cliente 127.0.0.1 12344
17677 17074 ./servidor
20224 14169 ps o pid,ppid,command
```

Onde claramente temos 3 uma instancia do servidor rodando para cada cliente, alám de uma adicional que é a que roda Accept, as 3 ultimas instancias de servidor tem o mesmo PPID (parent PID), que é igual ao PID da instancia principal do servidor (17074)

6 Questão 6

Como era de se esperar pela implementação, em situações normais (o servidor permanece e o cliente deixa a sessão (seja com Bye ou com um `SIGINT`), o cliente fica com o estado `TIME_WAIT` (testei com netstat -a -p). Em geral a maquina que inicia o processo de desconexão (`close(fd)`) termina nesse estado.