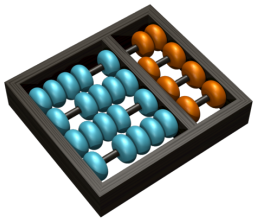


Exercício 2

Aluno: Caio Augusto Alves Nolasco

RA: 195181

Instituto de Computação

Universidade Estadual de Campinas

Campinas, 24 de Novembro de 2020.

Sumário

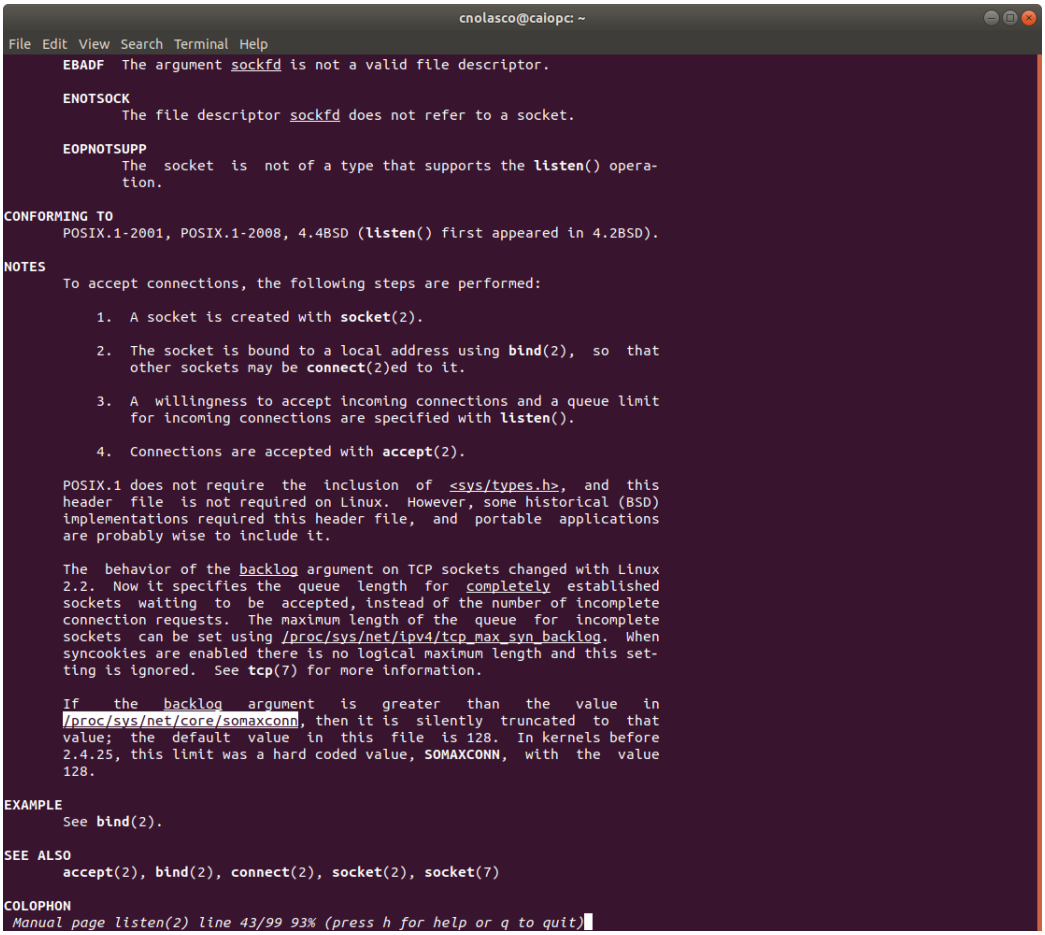
1 Questão 1

De maneira geral, o backlog do socket do servidor informa o número máximo de conexões pendentes suportadas por um socket TCP. De forma mais específica, o tamanho definido pelo valor do parâmetro backlog da função listen() define o valor máximo da soma dos tamanhos de duas filas de conexões no socket: uma fila para conexões que ainda esperam o término do three way handshake e outra fila para conexões que já tiveram o handshake concluído.

2 Questão 2

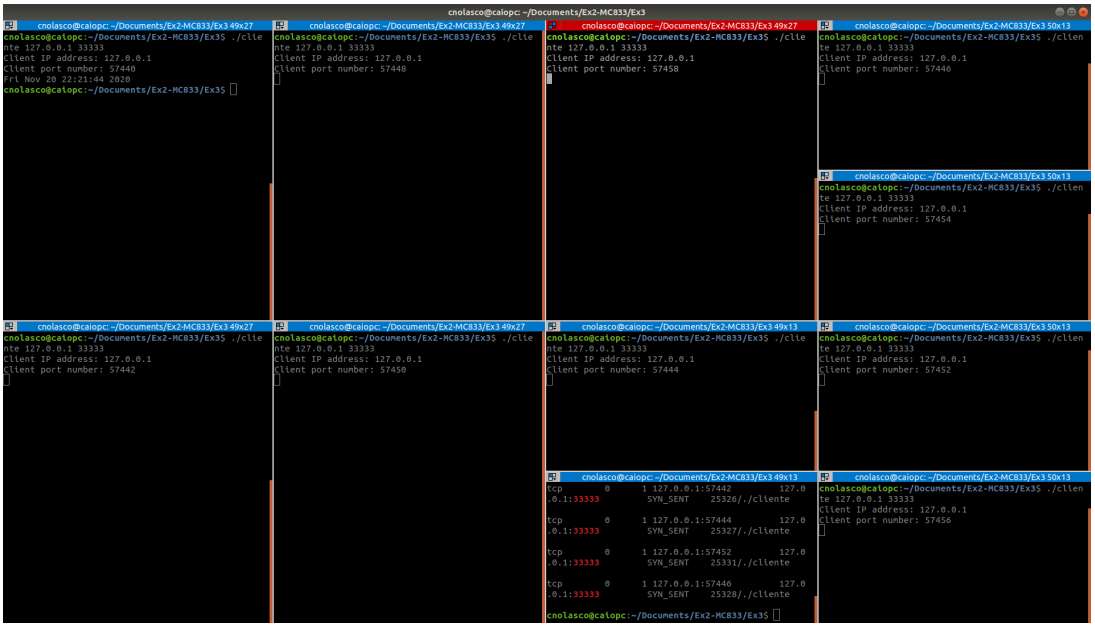
Para kernels do Linux 2.2 em diante, o valor de backlog especifica o tamanho máximo de file para conexões estabelescidas que esperam ser aceitas. O valor padrão de backlog é salvo em uma variavel de sistema no endereço /proc/sys/net/core/somaxconn, com seu valor padrão igual a 128.

A seguir, uma foto da manpage da chamada listen()

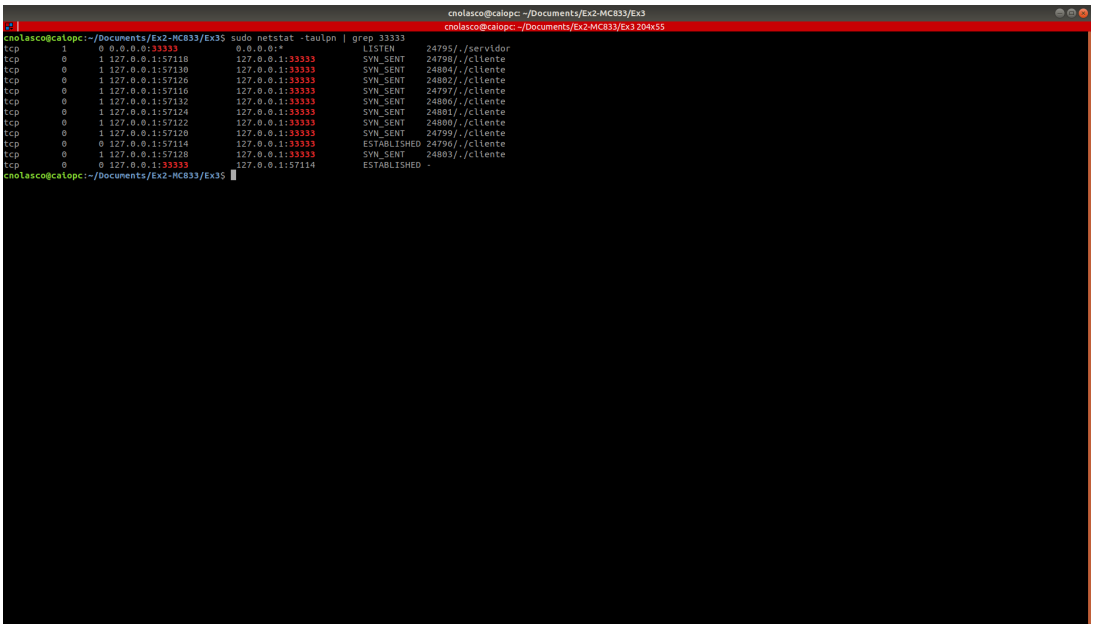


3 Questão 3

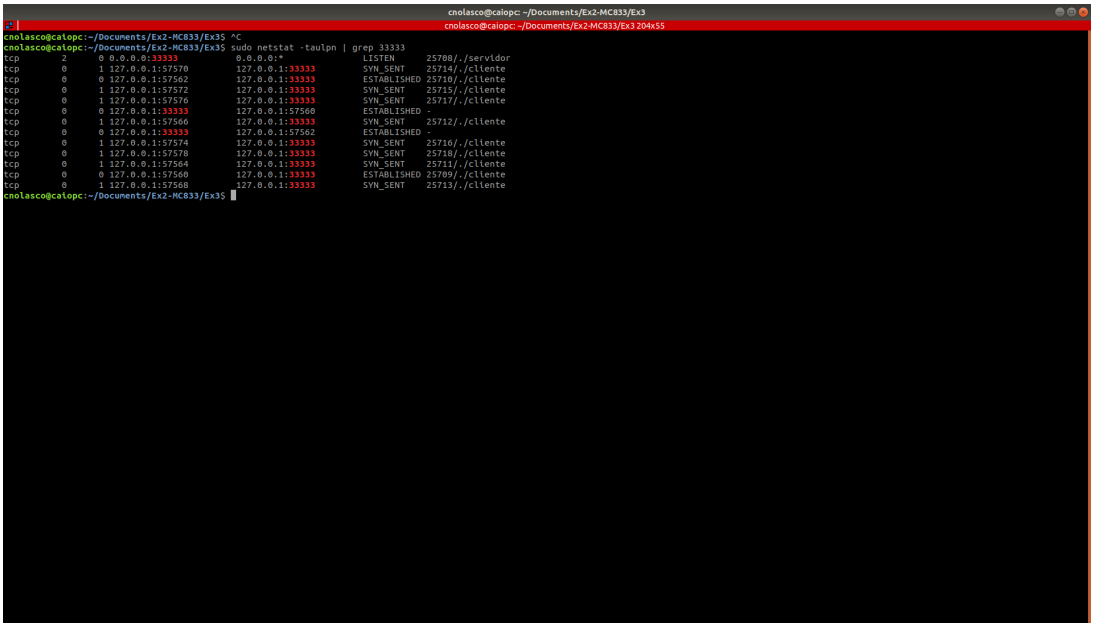
Os dez clientes são conectados simultaneamnte usando o programa Terminator para executar o cliente. Simultaneamente, o comando netstat pertinente para a porta do servidor também é chamado nesse momento.



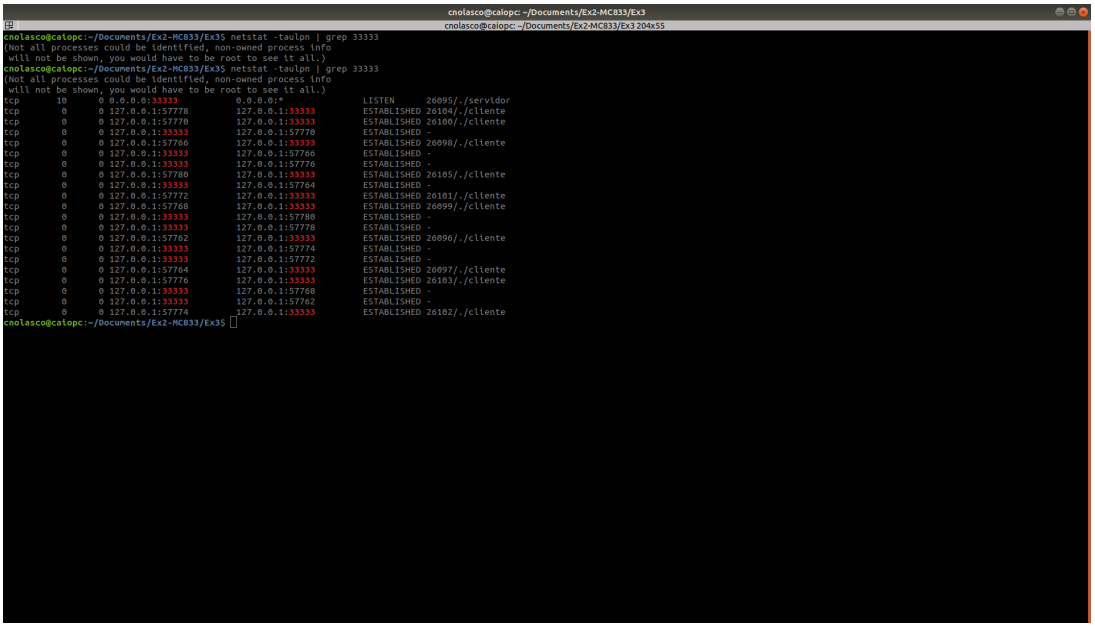
Com o valor de backlog setado para 0, apenas um cliente conesegue se conectar imediatamente ao servidor.



Similarmente, com o backlog igual a 1, dois clientes se conectam imediatamente ao servidor.



Com o backlog setado para 9, todos os cliente conseguem se conectar instantaneamente ao servidor.



4 Questão 4

O programa atualmente não faz uma chamada a função wait() e nem a função waitpid(), logo os processos filhos se tornam zumbis. Para lidar com os filhos zumbis, faz-se uma chamada para reconhecer o sinal SIGCHLD, e então uma chamada para wait().

```
void sig_chld() {

    pid_t pid;
    int stat;

    pid = wait(&stat);
    printf("child %d terminated\n", pid );
}
```

```
    bzero(&servaddr, sizeof(servaddr));
    servaddr.sin_family      = AF_INET;
    servaddr.sin_addr.s_addr = INADDR_ANY; //Adota como endereço de domínio o IP da máquina do servidor
    servaddr.sin_port        = htons(atoi(argv[1]));  //Permite que o sistema atribua um numero de porta temporário

    Bind(listenfd, (struct sockaddr *)&servaddr, sizeof(servaddr)); //Unir o socket com a porta de numero especificada

    Listen(listenfd, 9); //Iniciar a escuta por conexões, usando o descritor de socket listenfd criado

    Signal(SIGCHLD, sig_chld);

    for ( ; ; ) {
        sleep(3);
        connfd = Accept(listenfd, (struct sockaddr *) NULL, NULL); //Aceita a conexão, e inicia outro descritor de socket para esta conexão

        if ((childpid = Fork()) == 0)
        {
            Close(listenfd);
            str_echo(connfd, servaddr);
            exit(0);
        }

        Close(connfd);
    }
    return(0);
}
```