

# MC833 - Exercício 4

Felipe Santana Dias - 215775

## 1. Com o conhecimento adquirido em aula explique qual a relação entre backlog e número de conexões.

O backlog é um parâmetro responsável por especificar a quantidade de conexões pendentes que serão mantidas na fila de conexão, ajudando o controle do servidor especialmente em momentos de pico de conexões simultâneas.

## 2. Descreva o processo de troca de pacotes durante o estabelecimento da conexão com as filas mantidas pelo TCP para um socket de escuta.

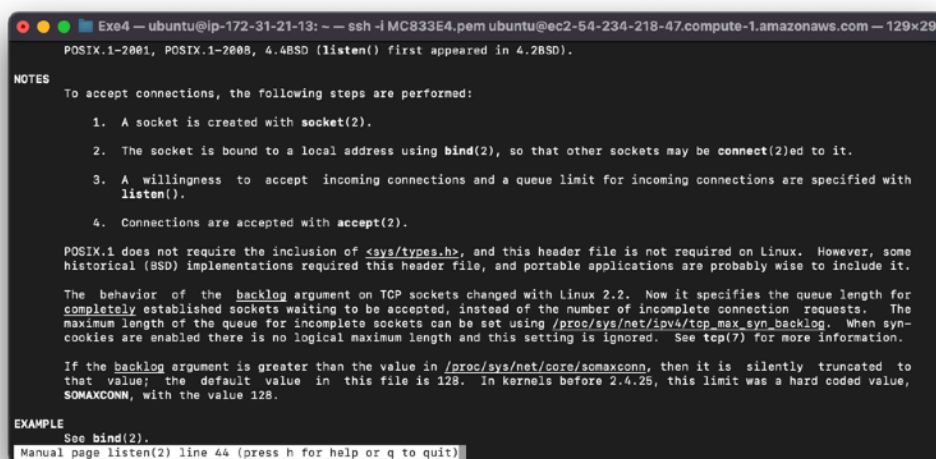
Para estabelecer uma conexão com as filas mantidas pelo TCP, o mecanismo three-way handshake é iniciado quando o cliente envia o SYN para o servidor e aguarda uma resposta confirmando sua conexão. Se a ocupação das filas não exceder ao valor determinado pelo backlog e o sistema operacional, o servidor retorna um pacote SYN/ACK aceitando a conexão. Quando o cliente envia o ACK, a conexão é completa.

Caso as filas estejam completas, o cliente não recebe nenhuma resposta ao enviar o SYN, reenviando o pacote depois de algum tempo na esperança de conectar-se. O processo repete-se até que o servidor retorne o SYN/ACK e inicie a conexão.

## 3. Descreva como é a implementação do backlog de um socket TCP no kernel linux (para versão 2.2 ou mais recentes), e em seguida indique o valor padrão do backlog. (Inclua figuras no seu relatório verificando esse valor padrão do backlog). Qual a diferença entre os valores de net.ipv4.tcp\_max\_syn\_backlog e net.core.somaxconn?

No kernel linux a partir da versão 2.2, o argumento backlog especifica o tamanho da fila de sockets com conexões completamente estabelecidas, sendo o valor padrão de 128 requisições.

O valor em net.ipv4.tcp\_max\_syn\_backlog define o tamanho máximo da fila de conexões incompletas enquanto o valor em net.core.somaxconn define o tamanho máximo da fila de conexões completas.



```
Exe4 - ubuntu@ip-172-31-21-13: ~ ssh -i MC833E4.pem ubuntu@ec2-54-234-218-47.compute-1.amazonaws.com - 129x29
POSIX.1-2001, POSIX.1-2008, 4.4BSD (listen()) first appeared in 4.2BSD).

NOTES
To accept connections, the following steps are performed:

    1. A socket is created with socket(2).
    2. The socket is bound to a local address using bind(2), so that other sockets may be connect(2)ed to it.
    3. A willingness to accept incoming connections and a queue limit for incoming connections are specified with
        listen().
    4. Connections are accepted with accept(2).

POSIX.1 does not require the inclusion of <sys/types.h>, and this header file is not required on Linux. However, some
historical (BSD) implementations required this header file, and portable applications are probably wise to include it.

The behavior of the backlog argument on TCP sockets changed with Linux 2.2. Now it specifies the queue length for
completely established sockets waiting to be accepted, instead of the number of incomplete connection requests. The
maximum length of the queue for incomplete sockets can be set using /proc/sys/net/ipv4/tcp_max_syn_backlog. When syn-
cookies are enabled there is no logical maximum length and this setting is ignored. See tcp(7) for more information.

If the backlog argument is greater than the value in /proc/sys/net/core/somaxconn, then it is silently truncated to
that value; the default value in this file is 128. In kernels before 2.4.25, this limit was a hard coded value,
SOMAXCONN, with the value 128.

EXAMPLE
See bind(2).
Manual page listen(2) line 44 (press h for help or q to quit)
```

- 4. Modifique o código do servidor do exercício anterior de modo que o valor do backlog passado para a função listen seja um argumento na linha de comando. Realize experimentos a fim de verificar quantos clientes conseguem de imediato conectar-se ao servidor (conexões em estado ESTABLISHED) para valores de backlog desde 0 até 10. Elabore um esquema para tentar conectar 10 clientes de forma simultânea. Os resultados obtidos condizem com o esperado?**

Os resultados foram condizentes com o esperado, sendo o número de conexões em estado ESTABLISHED foi correspondente ao valor definido pelo sistema operacional para cada valor de backlog, i.e. no caso em que o backlog era 0, foram realizadas 3 conexões de imediato, sendo esse o tamanho máximo da fila.

- 5. É correto afirmar que o código na versão atual gera processo zumbi? Explique. Se a sua resposta foi sim, então altere o código da questão 4 de modo que os processos criados pelo fork sejam corretamente finalizados ao invés de permanecerem no estado zumbi quando um cliente encerra sua conexão. Indique as mudanças feitas no código.**

Na versão atual do código, caso o servidor seja encerrado abruptamente será gerado processos zumbis por não finalizar as atividades realizadas pelo filho servidor. Para essa situação, uma das soluções é inserir a função wait antes do encerramento, obrigando o servidor pai a esperar que todos os processos dos servidores filhos sejam finalizados corretamente para, aí sim, encerrá-lo.