



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



TAREA. BACKLOG

ALUMNOS. MORALES HERNÁNDEZ CARLOS JESÚS

RAMÍREZ HIDALGO MARCO ANTONIO

BOLETAS: 2020630267, 2020630442

GRUPO. 3CM16

MATERIA. APLICACIONES PARA

COMUNICACIONES EN RED

PROF. AXEL ERNESTO MORENO CERVANTES

Backlog

En el protocolo TCP, las conexiones se tratan de forma incompleta a menos que se tenga lugar un reconocimiento entre el servidor y el cliente. Estas conexiones se llaman solicitudes de conexión pendientes. Se establece un valor máximo para estas solicitudes de conexión pendientes y se puede considerar una reserva de solicitudes en espera del puerto TCP para que el escucha acepte la solicitud. De esta forma, el parámetro backlog define la longitud máxima a la que puede llegar la cola de conexiones pendientes. Si una petición de conexión llega estando la cola llena las conexiones se rechazan con la bandera RST y el cliente puede recibir un error con una indicación de ECONNREFUSED o, si el protocolo subyacente acepta retransmisiones, la petición puede no ser tenida en cuenta, de forma que un reintento tenga éxito.

A continuación, se muestra el número máximo de solicitudes pendientes por plataforma.

Plataforma	Número máximo de solicitudes pendientes
AIX	100
HP-UX	20
Linux	100
IBM	255
Solaris	100
Servidor Windows	100
z/OS	255

Backlog en C

Para habilitar un socket para recibir peticiones de conexión y proceder al establecimiento de dicha conexión, es necesario utilizar la función `listen()`, esta función se invoca únicamente desde el servidor y solo se aplica a sockets de tipo `SOCK_STREAM`.

El formato de la función es “`int listen (int sockfd, int backlog);`” donde `sockfd` es el identificador de socket obtenido en la función `socket()`, que será utilizado para recibir conexiones y el `backlog` es el número máximo de conexiones en la cola de entrada. Las conexiones entrantes quedan en estado de espera en esta cola hasta que sean aceptadas mediante la función `accept()`.

Backlog en Java

En el caso de Java, la especificación se realiza a través del constructor del objeto `ServerSocket` el cual tiene la siguiente estructura “`public ServerSocket(int port, int backlog);`” donde el `backlog` debe tomar un valor mayor a cero, en caso contrario se utilizará la configuración por default.

Referencias

- IBM. (20/04/2021). "Utilización de la opción de reserva de escucha de TCP". (11/10/2022). Recuperado de:
<https://www.ibm.com/docs/es/ibmmq/7.5?topic=ssfksj-7-5-0-com-ibm-mq-con-doc-q016110--htm>
- Ubuntu Manuals. (s.f). "Ubuntu Manpage: listen – espera conexiones en un conector (socket)". (11/02/2022). Recuperado de:
<http://manpages.ubuntu.com/manpages/bionic/es/man2/listen.2.html>
- Universidad de Cantabria. (s.f). "Manual de Sockets en C". (11/03/2022). Recuperado de:
https://ocw.unican.es/pluginfile.php/1357/course/section/1683/PIRPractica4_ManualSocketsC.pdf
- Oracle. (s.f). "Class ServerSocket". (11/02/2022). Recuperado de:
<https://docs.oracle.com/javase/8/docs/api/java/net/ServerSocket.html>