Backlog

López Manríquez Ángel
3CM7

May 28, 2019

The listen function completes the binding necessary for a socket and creates a connection request queue for incoming requests.

1 Last updated

Added for PUT00.

2 Format

```
 #include <sys/socket.h>
int listen(int s, int backlog);
```

Where **s** is the socket descriptor and **backlog** is the maximum length for the queue of pending connections. If backlog is less than 0, its value is set to 0. The maximum value for backlog is 32 767. If the value passed is greater than 32 767, the value is set to 32 767.

3 Normal return

Return code 0 indicates that the function was successful.

4 Error return

A return code equal to -1 indicates an error. You can get the specific error code by calling sock_errno. See z/TPF C/C++ Language Support User's Guide for more information about socket errors.

- **SOCOPNOTSUP** The s parameter is not a socket descriptor that supports the listen function.

- **SOCNOTSOCK** The s parameter is not a valid socket descriptor.
- **SOCINVAL** The socket is not in the correct state for listening.
- **E1052STATE** The socket was closed because the system was in or cycling down to 1052 state.

5 Programming considerations

- The listen function applies only to stream sockets. The function performs the following tasks:
  - It completes the binding necessary for socket s if bind has not been called for s.
  - It creates a connection request queue, which is the length of the backlog parameter, to queue incoming connection requests. After the queue is full, additional connection requests are rejected.
- The listen function indicates a readiness to accept client connection requests. This function transforms an active socket into a passive socket. Once called, s can never be used as an active socket to start connection requests. Calling listen is the third of four steps that a server performs to accept a connection. This function is called after allocating a stream socket with socket, and after binding a name to s with bind. The listen function must be called before calling accept.

Sources `https://www.ibm.com/support/knowledgecenter/en/SSB23S_1.1.0.15/gtpc2/cpp_listen.html`