Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'endservent.3p' command

$ man endservent.3p

ENDSERVENT(3P) POSIX Programmer's Manual ENDSERVENT(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

endservent, getservbyname, getservbyport, getservent, setservent ? net? work services database functions

SYNOPSIS

```
#include <netdb.h>
void endservent(void);
struct servent *getservbyname(const char *name, const char *proto);
struct servent *getservbyport(int port, const char *proto);
struct servent *getservent(void);
void setservent(int stayopen);
```

DESCRIPTION

These functions shall retrieve information about network services. This information is considered to be stored in a database that can be ac? cessed sequentially or randomly. The implementation of this database is unspecified.

The setservent() function shall open a connection to the database, and set the next entry to the first entry. If the stayopen argument is non-

zero, the net database shall not be closed after each call to the get?
servent() function (either directly, or indirectly through one of the
other getserv*() functions), and the implementation may maintain an
open file descriptor for the database.

The getservent() function shall read the next entry of the database,
opening and closing a connection to the database as necessary.

The getservbyname() function shall search the database from the begin?
ning and find the first entry for which the service name specified by
name matches the s_name member and the protocol name specified by proto
matches the s_proto member, opening and closing a connection to the
database as necessary. If proto is a null pointer, any value of the
s_proto member shall be matched.

The getservbyport() function shall search the database from the begin?
ning and find the first entry for which the port specified by port
matches the s_port member and the protocol name specified by proto
matches the s_proto member, opening and closing a connection to the
database as necessary. If proto is a null pointer, any value of the
s_proto member shall be matched. The port argument shall be a value ob?
tained by converting a uint16_t in network byte order to int.

The getservbyname(), getservbyport(), and getservent() functions shall
each return a pointer to a servent structure, the members of which
shall contain the fields of an entry in the network services database.

The endservent() function shall close the database, releasing any open
file descriptor.

These functions need not be thread-safe.

RETURN VALUE

Upon successful completion, getservbyname(), getservbyport(), and get?
servent() return a pointer to a servent structure if the requested en?
try was found, and a null pointer if the end of the database was
reached or the requested entry was not found. Otherwise, a null
pointer is returned.

The application shall not modify the structure to which the return
value points, nor any storage areas pointed to by pointers within the

structure. The returned pointer, and pointers within the structure, might be invalidated or the structure or the storage areas might be overwritten by a subsequent call to getservbyname(), getservbyport(), or getservent(). The returned pointer, and pointers within the struc? ture, might also be invalidated if the calling thread is terminated.

ERRORS

No errors are defined.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

The port argument of getservbyport() need not be compatible with the port values of all address families.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

endhostent(), endprotoent(), htonl(), inet_addr()

The Base Definitions volume of POSIX.1?2017, <netdb.h>

COPYRIGHT

nel.org/doc/man-pages/reporting_bugs.html .