NAME

getservent, getservbyport, getservbyname, setservent, endservent - get service entry

LIBRARY

```
Standard C Library (libc, -lc)
```

SYNOPSIS

```
#include <netdb.h>

struct servent *
getservent();

struct servent *
getservbyname(const char *name, const char *proto);

struct servent *
getservbyport(int port, const char *proto);

void
setservent(int stayopen);

void
endservent(void);
```

DESCRIPTION

The **getservent()**, **getservbyname()**, and **getservbyport()** functions each return a pointer to an object with the following structure containing the broken-out fields of a line in the network services data base, */etc/services*.

The members of this structure are:

s_name The official name of the service.

s_aliases A zero terminated list of alternate names for the service.

s_port The port number at which the service resides. Port numbers are returned in network byte order.

s proto The name of the protocol to use when contacting the service.

The **getservent()** function reads the next line of the file, opening the file if necessary.

The **setservent**() function opens and rewinds the file. If the *stayopen* flag is non-zero, the net data base will not be closed after each call to **getservbyname**() or **getservbyport**().

The **endservent**() function closes the file.

The **getservbyname**() and **getservbyport**() functions sequentially search from the beginning of the file until a matching protocol name or port number (which must be specified in network byte order) is found, or until EOF is encountered. If a protocol name is also supplied (non- NULL), searches must also match the protocol.

FILES

/etc/services /var/db/services.db

DIAGNOSTICS

Null pointer returned on EOF or error.

SEE ALSO

getprotoent(3), services(5), services_mkdb(8)

HISTORY

The **getservent()**, **getservbyport()**, **getservbyname()**, **setservent()**, and **endservent()** functions appeared in 4.2BSD.

BUGS

These functions use a thread-specific data storage; if the data is needed for future use, it should be copied before any subsequent calls overwrite it. Expecting port numbers to fit in a 32 bit quantity is probably naive.