FUJITSU Software BS2000 POSIX-NSL

Version 10.0A
Oktober 2016

Readme-Datei

Alle Rechte vorbehalten, insbesondere gewerbliche Schutzrechte. Änderung von technischen Daten sowie Lieferbarkeit vorbehalten. Haftung oder Garantie für Vollständigkeit, Aktualität und Richtigkeit der angegebenen Daten und Abbildungen ausgeschlossen. Wiedergegebene Bezeichnungen können Marken und/oder Urheberrechte sein, deren Benutzung durch Dritte für eigene Zwecke die Rechte der Inhaber verletzen kann.

© 2016 Fujitsu Technology Solutions GmbH

Die Marke Fujitsu und das Fujitsu Logo sind Marken oder registrierte Marken von Fujitsu Limited in Japan und in anderen Ländern. BS2000 ist eine Marke von Fujitsu Technology Solutions GmbH in Europa und in anderen Ländern.

1 Einleitung

Die vorliegende Readme-Datei enthält Hinweise zum Software-Produkt POSIX-NSL, das die NSL-Bibliothek für die Entwicklung von POSIX-Software bereitstellt. Die NSL-Bibliothek enthält die Funktionen für die Services TLI, XDR und RPC.

2 Produktbestandteile und Installation

2.1 Installation im BS2000

Folgende Dateien können mit IMON im BS2000 installiert sein:

	MODE	SHARE	ACCESS	MIGRATE
$<TSOS>.SYSLIB.POSIX-NSL.100	COM	Y	R	I
$<TSOS>.SKMLIB.POSIX-NSL.100	X86	Y	R	I
$<TSOS>.SYSRME.POSIX-NSL.100.D	COM	Y	R	I
$<TSOS>.SYSRME.POSIX-NSL.100.E	COM	Y	R	I
$<TSOS>.SYSSII.POSIX-NSL.100	COM	Y	R	A

- Die SYSLIB.POSIX-NSL.100 enthält die Header-Dateien und die Module im /390-Format zum Compilieren und Binden im BS2000 sowie zum Installieren der NSL-Bibliothek /usr/lib/libnsl.a in das POSIX-Dateisystem.

- Die SKMLIB.POSIX-NSL.100 enthält die Header-Dateien und die Module im X86-Format zum Compilieren und Binden im BS2000 sowie zum Installieren der NSL-Bibliothek /usr/lib/libnsl.a bzw. /usr/lib/X86/libnsl.a in das POSIX-Dateisystem.

2.2 Installation im POSIX-Dateisystem

Die Installation erfolgt mit dem POSIX-Installationsprogramm (/START-POSIX-INSTALLATION, Funktion Paketinstallation). Siehe hierzu das Handbuch „POSIX Grundlagen“.

Für die Installation im POSIX-Dateisystem werden ca. 9 MB temporärer Speicher unter /tmp und ca. 9 MB permanenter Speicher pro installierter Variante unter /usr/lib bzw. /usr/lib/X86 benötigt.

Die Installation kann einige Zeit in Anspruch nehmen.

Falls die Installation nicht korrekt durchgeführt werden kann, wird die Textdatei /tmp/install.libnsl.err erstellt, die Hinweise auf die Fehlerursache enthält.

Installierte Bibliotheken:

Ist nur die BS2000-Datei SYSLIB.POSIX-NSL.100 installiert, so wird nur die Bibliothek `/usr/lib/libnsl.a` mit der /390 Code Ausprägung im POSIX-Dateisystem erzeugt.

Sind verschiedene Varianten der BS2000-Bibliotheken installiert, so werden auch verschiedene Varianten der NSL-Bibliotheken generiert:

`/usr/lib/libnsl.a`	mit /390 Code
`/usr/lib/X86/libnsl.a`	mit X86 Code

In diesem Fall erfolgt das Einbinden der X86-Variante dann durch Angabe der Compiler-Option
`-L /usr/lib/X86`

Installierte Dateien:

```
/usr/bin/rpcgen
/usr/bin/rpcgen_cpp

/usr/include/netconfig.h
/usr/include/netdb.h
/usr/include/netdir.h
/usr/include/tiuser.h
/usr/include/arpa.ftp.h
/usr/include/arpa/ftp.h
/usr/include/arpa.inet.h
/usr/include/arpa/inet.h
/usr/include/arpa.nameser.h
/usr/include/arpa/nameser.h
/usr/include/arpa.telnet.h
/usr/include/arpa/telnet.h
/usr/include/arpa.tftp.h
/usr/include/arpa/tftp.h
/usr/include/net.af.h
/usr/include/net/af.h
/usr/include/net.if.h
/usr/include/net/if.h
/usr/include/net.if_arp.h
/usr/include/net/if_arp.h
/usr/include/net.route.h
/usr/include/net/route.h
/usr/include/net.strioc.h
/usr/include/net/strioc.h
/usr/include/netinet.arp.h
/usr/include/netinet/arp.h
/usr/include/netinet.icmp_var.h
/usr/include/netinet/icmp_var.h
/usr/include/netinet.if_ether.h
/usr/include/netinet/if_ether.h
/usr/include/netinet.in.h
/usr/include/netinet/in.h
/usr/include/netinet.insrem.h
/usr/include/netinet/insrem.h
/usr/include/netinet.in_pcb.h
/usr/include/netinet/in_pcb.h
```

```
/usr/include/netinet.in_systm.h
/usr/include/netinet/in_systm.h
/usr/include/netinet.in_var.h
/usr/include/netinet/in_var.h
/usr/include/netinet.ip.h
/usr/include/netinet/ip.h
/usr/include/netinet.ip_icmp.h
/usr/include/netinet/ip_icmp.h
/usr/include/netinet.ip_str.h
/usr/include/netinet/ip_str.h
/usr/include/netinet.ip_var.h
/usr/include/netinet/ip_var.h
/usr/include/netinet.llcloop.h
/usr/include/netinet/llcloop.h
/usr/include/netinet.nihdr.h
/usr/include/netinet/nihdr.h
/usr/include/netinet.symredef.h
/usr/include/netinet/symredef.h
/usr/include/netinet.tcp.h
/usr/include/netinet/tcp.h
/usr/include/netinet.tcpip.h
/usr/include/netinet/tcpip.h
/usr/include/netinet.tcp_debug.h
/usr/include/netinet/tcp_debug.h
/usr/include/netinet.tcp_fsm.h
/usr/include/netinet/tcp_fsm.h
/usr/include/netinet.tcp_seq.h
/usr/include/netinet/tcp_seq.h
/usr/include/netinet.tcp_timer.h
/usr/include/netinet/tcp_timer.h
/usr/include/netinet.tcp_var.h
/usr/include/netinet/tcp_var.h
/usr/include/netinet.udp.h
/usr/include/netinet/udp.h
/usr/include/netinet.udp_var.h
/usr/include/netinet/udp_var.h
/usr/include/rpc.auth.h
/usr/include/rpc/auth.h
/usr/include/rpc.auth_des.h
/usr/include/rpc/auth_des.h
/usr/include/rpc.auth_sys.h
/usr/include/rpc/auth_sys.h
/usr/include/rpc.auth_unix.h
/usr/include/rpc/auth_unix.h
/usr/include/rpc.clnt.h
/usr/include/rpc/clnt.h
/usr/include/rpc.clnt_soc.h
/usr/include/rpc/clnt_soc.h
/usr/include/rpc.des_crypt.h
/usr/include/rpc/des_crypt.h
/usr/include/rpc.key_prot.h
/usr/include/rpc/key_prot.h
/usr/include/rpc.nettype.h
/usr/include/rpc/nettype.h
/usr/include/rpc.pmap_clnt.h
/usr/include/rpc/pmap_clnt.h
/usr/include/rpc.pmap_prot.h
/usr/include/rpc/pmap_prot.h
/usr/include/rpc.pmap_rmt.h
/usr/include/rpc/pmap_rmt.h
/usr/include/rpc.raw.h
```

```
/usr/include/rpc/raw.h
/usr/include/rpc.rpc.h
/usr/include/rpc/rpc.h
/usr/include/rpc.rpcb_clnt.h
/usr/include/rpc/rpcb_clnt.h
/usr/include/rpc.rpcb_prot.h
/usr/include/rpc/rpcb_prot.h
/usr/include/rpc.rpcent.h
/usr/include/rpc/rpcent.h
/usr/include/rpc.rpc_com.h
/usr/include/rpc/rpc_com.h
/usr/include/rpc.rpc_mp.h
/usr/include/rpc/rpc_mp.h
/usr/include/rpc.rpc_msg.h
/usr/include/rpc/rpc_msg.h
/usr/include/rpc.svc.h
/usr/include/rpc/svc.h
/usr/include/rpc.svc_auth.h
/usr/include/rpc/svc_auth.h
/usr/include/rpc.svc_soc.h
/usr/include/rpc/svc_soc.h
/usr/include/rpc.types.h
/usr/include/rpc/types.h
/usr/include/rpc.xdr.h
/usr/include/rpc/xdr.h
/usr/include/sys.byteorder.h
/usr/include/sys/byteorder.h
/usr/include/sys.cmn_err.h
/usr/include/sys/cmn_err.h
/usr/include/sys.netconfig.h
/usr/include/sys/netconfig.h
/usr/include/sys.socket.h
/usr/include/sys/socket.h
/usr/include/sys.t_kuser.h
/usr/include/sys/t_kuser.h
/usr/include/sys.tiuser.h
/usr/include/sys/tiuser.h
/usr/include/sys.xti_inet.h
/usr/include/sys/xti_inet.h
```

3 Funktionen

Die NSL-Bibliothek enthält die Funktionen für die Services TLI, XDR und RPC.

Folgende TLI-Funktionen werden unterstützt:

```
int     t_accept ( int fd, int resfd, struct t_call *call );
char   *t_alloc ( int fd, inst struct_type, int fields );
int     t_bind ( int fd, struct t_bind *req, struct t_bind *ret );
int     t_close ( int fd );
int     t_connect ( int fd, struct t_call *sndcall,
                    struct t_call *rcvcall );
int     t_error ( char *errmsg );
int     t_free ( char *ptr, int struct_type );
int     t_getinfo ( int fd, struct t_info *info );
int     t_getprotaddr ( int fd, struct t_bind *boundaddr,
                        struct t_bind *peeraddr );
int     t_getstate ( int fd );
int     t_listen ( int fd, struct t_call *call );
int     t_look ( int fd );
int     t_open ( char *name, int oflag, struct t_info *info );
int     t_optmgmt ( int fd, struct t_optmgmt *req,
                    struct t_optmgmt *ret );
        unterstuetzte Optionen: TCP_NODELAY, SO_BROADCAST,
                                SO_KEEPALIVE;
int     t_rcv ( int fd, char *buf, unsigned nbytes, int *flags );
int     t_rcvconnect ( int fd, struct t_call *call );
int     t_rcvdis ( int fd, struct t_discon *discon );
int     t_rcvrel ( int fd );
int     t_rcvudata ( int fd, struct t_unitdata *unitdata,
                     int *flags );
int     t_rcvuderr ( int fd, struct t_uderr *uderr );
int     t_snd ( int fd, char *buf, unsigned nbytes, int flags );
int     t_snddis ( int fd, struct t_call *call );
int     t_sndrel ( int fd );
int     t_sndudata ( int fd, struct t_unitdata *unitdata );
int     t_sync ( int fd );
int     t_unbind ( int fd );
```

Die XDR-Funktionen `xdr_string()`, `xdr_char()` und `xdr_u_char()` ENCODEieren nach ASCII und DECODEieren nach EBCDIC. Die XDR-Funktionen `xdr_string_noc()`, `xdr_char_noc()` und `xdr_u_char_noc()` bieten die gleiche Funktionalität ohne ASCII/EBCDIC-Konvertierung.

4 Einschränkungen

Die Funktions-Entries sind nur in Großschreibung unterstützt.

Alle Funktionen der Bibliothek sind mit der Compiler-Option ENUM-TYPE=LONG (bzw. cc/c89 Option -XL ...) übersetzt. Programme, die RPC- oder XDR-Funktionen benutzen, müssen mit derselben Option übersetzt werden. Ansonsten kann es sein, dass das Programm nicht korrekt funktioniert (wg. unterschiedlicher Ausrichtung der Daten).