
Inhaltsverzeichnis

1	Einführung	1
1.1	TCP/IP-Grundlagen	2
1.1.1	Netzwerkschicht	3
1.1.2	Internet-Schicht	4
1.1.3	Transportschicht	5
1.1.4	Anwendungsschicht	6
1.2	Internet-Standards	6
1.3	Unix-Standards	7
2	Programmieren mit Unix-Prozessen	9
2.1	Unix-Prozesse	9
2.1.1	Prozeßgruppen und Sessions	10
2.1.2	Kontrollierendes Terminal	12
2.1.3	Verwaiste Prozesse und verwaiste Prozeßgruppen	13
2.1.4	Prozeßumgebung	14
2.1.5	Lebenszyklus	15
2.1.6	User- und Gruppen-ID	20
2.2	Ein- und Ausgabe	21
2.2.1	Dateideskriptoren	21
2.2.2	Elementare Ein- und Ausgabe	24
2.2.3	Standerdeingabe und -ausgabe	35
2.2.4	Ausgabe über den Syslog-Dienst	41
2.3	Buffer-Overflows und Format-String-Schwachstellen	45
2.3.1	Buffer-Overflows	47
2.3.2	Format-String-Schwachstellen	54
2.3.3	Geeignete Gegenmaßnahmen	57
2.4	Signale	58
2.4.1	Signale behandeln	59
2.4.2	Signale blockieren	68
2.4.3	Signale annehmen	72
2.4.4	Signale generieren	74

2.5	Prozeßkontrolle	76
2.5.1	Was bin ich? Prozeß-IDs und mehr	77
2.5.2	Neue Prozesse erzeugen	79
2.5.3	Prozesse synchronisieren	83
2.5.4	Zombie-Prozesse	89
2.5.5	Andere Programme ausführen	90
2.5.6	User- und Group-IDs wechseln	94
2.6	Dæmon-Prozesse	96
3	Programmieren mit POSIX-Threads	103
3.1	Grundlagen	104
3.2	Synchronisation	115
3.2.1	Race Conditions und kritische Bereiche	116
3.2.2	Gegenseitiger Ausschluß	120
3.2.3	Bedingungsvariablen	126
3.3	Pthreads und Unix-Prozesse	135
3.3.1	Threadsichere und eintrittsinvariante Funktionen	135
3.3.2	Fehlerbehandlung und errno	137
3.3.3	Signalverarbeitung	138
3.3.4	fork() und exec() in Pthreads-Programmen	144
4	Grundlagen der Socket-Programmierung	147
4.1	Erste Schritte mit telnet und inetd	147
4.1.1	Das telnet-Kommando als Netzwerk-Client	147
4.1.2	Einfache Netzwerkdienste mit dem inetd	152
4.2	IP-Namen und IP-Adressen	156
4.2.1	Das Domain Name System	157
4.2.2	IPv4-Adressen	159
4.2.3	IPv6-Adressen	164
4.2.4	Netzwerkdarstellung von IP-Adressen	168
4.3	Sockets	179
4.3.1	Socket anlegen	180
4.3.2	Socket-Strukturen	182
4.3.3	Client-seitiger TCP-Verbindungsaufbau	184
4.3.4	Socket-Adressen zuweisen	189
4.3.5	Annehmende Sockets	192
4.3.6	TCP-Verbindungen annehmen	194
4.3.7	Drei-Wege-Handshake und TCP-Zustandübergänge ..	199
4.3.8	Kommunikation über UDP	205
4.3.9	Standardeingabe und -ausgabe über Sockets	212
4.3.10	Socket-Adressen ermitteln	213
4.3.11	Multiplexing von Netzwerkverbindungen	218
4.3.12	Socket-Optionen	223
4.4	Namensauflösung	227

5	Netzwerkprogrammierung in der Praxis	235
5.1	Aufbau der Testumgebung	236
5.1.1	Funktionsumfang der Testumgebung	237
5.1.2	Hilfsfunktionen für die Socket-Kommunikation	238
5.1.3	Der Test-Client	249
5.2	Iterative Server	255
5.2.1	Sequentielle Verarbeitung der Anfragen	256
5.2.2	Clientbehandlung	259
5.2.3	Hilfsfunktionen zur Laufzeitmessung	262
5.2.4	Eigenschaften und Einsatzgebiete	264
5.3	Nebenläufige Server mit mehreren Threads	266
5.3.1	Abgewandelte Signalbehandlung	267
5.3.2	Ein neuer Thread pro Client	268
5.3.3	Das Hauptprogramm als Signalverarbeiter	270
5.3.4	Eigenschaften und Einsatzgebiete	272
5.4	Nebenläufige Server mit Prethreading	274
5.4.1	Clientbehandlung mittels paralleler Accept-Handler	275
5.4.2	Das Hauptprogramm als Signalverarbeiter	277
5.4.3	Eigenschaften und Einsatzgebiete	279
5.5	Nebenläufige Server mit mehreren Prozessen	281
5.5.1	Anpassung der Signalbehandlung	282
5.5.2	Ein neuer Prozeß pro Client	284
5.5.3	Das Hauptprogramm	286
5.5.4	Eigenschaften und Einsatzgebiete	287
5.6	Nebenläufige Server mit Preforking	289
5.6.1	Buchführende Signalbehandlung	290
5.6.2	Parallele Accept-Handler in mehreren Prozessen	292
5.6.3	Preforking im Hauptprogramm	294
5.6.4	Eigenschaften und Einsatzgebiete	296
5.7	Zusammenfassung	298
6	Netzwerkprogrammierung mit SSL	301
6.1	Strategien zur Absicherung des Datenverkehrs	302
6.1.1	Datenverschlüsselung	304
6.1.2	Hashfunktionen und Message Authentication Codes	307
6.1.3	Digitale Signaturen	308
6.1.4	Zertifizierungsstellen und digitale Zertifikate	309
6.1.5	Praktische Absicherung des Datenverkehrs	310
6.2	SSL-Grundlagen	313
6.2.1	Datentransfer über SSL	314
6.2.2	Anwendungsprotokolle um SSL erweitern	317
6.2.3	SSL-Verbindungen interaktiv testen	321
6.3	OpenSSL-Basisfunktionalität	323
6.3.1	Das Konzept der BIO-API	324
6.3.2	Lebenszyklus von BIO-Objekten	325

6.3.3	Ein-/Ausgabe über BIO-Objekte	326
6.3.4	BIO-Quellen/Senken und BIO-Filter	329
6.3.5	Fehlerbehandlung	342
6.3.6	Thread-Support	345
6.3.7	Pseudozufallszahlengenerator	352
7	Client-/Server-Programmierung mit OpenSSL	357
7.1	Initialisierung der ssl-Bibliothek	357
7.2	Der SSL-Kontext	359
7.2.1	Ein unvollständiger SSMTP-Client	360
7.2.2	SSL-Optionen, SSL-Modi und Chiffrenfolgen	364
7.3	Sicherer Umgang mit X.509-Zertifikaten	368
7.3.1	Zertifikatsüberprüfung aktivieren	372
7.3.2	Zertifikatsüberprüfung per Callback nachbereiten	374
7.3.3	Identitätsabgleich mit digitalen Zertifikaten	380
7.3.4	SSL-Kommunikation mit eigener Identität	387
7.4	Client-/Server-Beispiel: SMTP mit STARTTLS	389
7.4.1	Ein SMTP-Client mit STARTTLS	389
7.4.2	Ein SMTP-Server mit STARTTLS	397
7.5	Zusammenfassung	406
A	Anhang	409
A.1	Zertifikate erstellen mit OpenSSL	409
A.1.1	Aufbau einer Zertifizierungsstelle	409
A.1.2	Neue Zertifikate ausstellen	412
A.1.3	Vertrauenswürdige Zertifizierungsstellen	414
A.2	Barrieren mit POSIX-Threads	415
	Literaturverzeichnis	423
	Sachverzeichnis	427