Frei erhältliche Software für UNIX-Administratoren

J. Schönwälder* S. Petri H. Langendörfer

TU Braunschweig
Institut für Betriebssysteme und Rechnerverbund

Kurzfassung

Das Internet bietet für UNIX-Systeme eine große Anzahl frei erhältlicher Programme an, die den Administrationsaufwand reduzieren helfen. In diesem Beitrag möchten wir einen (durchaus subjektiv geprägten) Überblick über einige der wichtigsten Programme geben, um einerseits neuen UNIX Administratoren den Einstieg zu erleichtern und andererseits zu einem Erfahrungsaustausch anzuregen.

Abstract

The Internet provides a lot of free software for UNIX system administration. This survey of some of the most important programs should help new system administrators to find software that solves their problems. We further would like to motivate an open discussion about experiences with free software for system administration tasks.

Einleitung

Das Betriebssystem UNIX findet in den letzten Jahren im kommerziellen und privaten Bereich immer größere Verbreitung. Damit steigt auch die Anzahl von Personen, die sich mit der (nicht immer ganz einfachen) Administration von UNIX-Systemen befassen müssen. Durch die frühe Verbreitung von UNIX an Universitäten und Forschungslabors wurden jedoch Lösungen für viele mit dem UNIX-Einsatz verbundene Probleme entwickelt, die oftmals frei verfügbar sind. Obwohl diese Programme häufig aus dem konkreten Bedarf entstanden sind und es meist keinen offiziellen Support gibt, sind sie in der Regel sehr zuverlässig und teilweise kommerziellen Lösungen sogar überlegen.

In diesem Beitrag stellen wir eine Sammlung von diesen frei verfügbaren Programmen vor, die sich bei uns im täglichen Einsatz bewährt haben oder die für uns wichtige Anregungen gegeben haben. Die Auswahl ist sicherlich subjektiv geprägt und unvollständig, sollte aber als Ausgangspunkt dienen, um Neueinsteigern etwas Orientierung zu geben. Am Ende des Artikels befindet sich eine Liste mit Literatur und FTP Adressen für die hier genannte Software.

GNU-Software

Das GNU Projekt der Free Software Foundation ist sicherlich eines der bekanntesten Projekte, in dem anerkannt gute Software erstellt wird, die als Ersatz für fehlende oder bestehende (fehlerhafte) Programme dient. Auf unseren Maschinen gehören der GNU Compiler `gcc`, der Debugger `gdb`, der Editor `emacs` und eine Reihe von verbesserten UNIX Kommandos (`gtar, less, bash, ...`) zur Standardausstattung. Insbesondere durch das neuerdings auftretende Verhalten einiger Hersteller, unvollständige UNIX-Systeme auszuliefern (fehlender C-Compiler etc.), ist GNU Software notwendig, um überhaupt einen vernünftigen Betrieb zu ermöglichen.

Sprachen

UNIX-Administratoren verbringen einen nicht unerheblichen Teil ihrer Zeit damit, Standardaufgaben in Form von Skripten zu automatisieren. Dazu ist es unerläßlich, sich mit mindestens einer Skript-Sprache genau auszukennen. Startpunkt sollte immer die Bourne Shell `/bin/sh` sein, da Bourne Shell Skripte relativ portabel sind. Neuere Shells, die auf der Syntax der Bourne Shell aufsetzen (`ksh, bash`), sind zwar angenehmer zu benutzen, aber eben auch nicht so portabel.

Weit verbreitet hat sich auch die Programmiersprache `perl`, die Eigenschaften der Sprache C und der Hilfsprogramme `sed, awk` und der Bourne Shell `sh` kombiniert [17]. Mit Hilfe von `perl` lassen sich oftmals Aufgaben für eine ganze Reihe von heterogenen Maschinen lösen, da man die vielen versteckten Tücken in den verschiedenen UNIX-Versionen durch die einheitliche Schnittstelle von `perl` geschickt umgeht. Es lassen sich sogar `setuid` Skripte in `perl` halbwegs vernünftig schreiben.

Eine andere wichtige Sprache ist die Tool Command Language (`tcl`) [13]. Sie besteht aus einem kleinen portablen Interpreter, der als C Bibliothek vorliegt und eine einfache aber mächtige Kommandosprache bereitstellt. Der Interpreter läßt sich leicht in eigene Programme einbetten und um applikationsspezifische Befehle erweitern. `tcl` allein ist für den System Adminstrator nur von geringem Interesse, bekommt aber im Zusammenhang mit X11 mehr Bedeutung.

X11

Das Grafiksystem X11 gehört mittlerweile zur Standardausstattung von UNIX-Systemen. Allerdings legen die einzelnen Hersteller großen Wert darauf, die vom X-Konsortium gelieferte X11 Version zu erweitern. Dies wird spätestens dann zum Problem, wenn man eine heterogene Landschaft hat und sich die Benutzer fragen, warum denn die Maschinen alle so unterschiedlich zu bedienen sind. Außerdem sind die Zeiten, die einige Hersteller brauchen, um einheitlich ein neues Release zu unterstützen, oftmals viel zu lang. Wir betreiben daher auf unseren Maschinen einheitlich die X11 Version vom X-Konsortium. Das bedeutet beim Installieren etwas mehr Aufwand, zahlt sich aber durch eine einheitliche Oberfläche und meist geringere Probleme bei der Installation von X11 Applikationen wieder aus.

Für den UNIX-Administrator, der den Zugang zu Systemdiensten (z.B. Druckerverwaltung) durch grafische Oberflächen unterstützen soll, ist das X11 Toolkit `tk` das Mittel der Wahl. `tk` ist eine Erweiterung des bereits genannten Kommandointerpreters `tcl` und stellt Motif-ähnliche Widgets in einer Skript-Sprache zur Verfügung [13]. Mit Hilfe von `tk` lassen sich mit relativ geringem Aufwand ansprechend aussehende X11 Oberflächen realisieren. Außerdem gibt es eine Menge von Beispielen und Applikationen wie z.B. `tk`-basierte Mail- und Newsreader, Manualbrowser, Terminverwaltungen in einem zentralen Archiv.

Dateisystem

Die Organisation des eigenen Dateisystems ist ein kritischer Punkt, da ein logischer Aufbau den Benutzern (und auch den Betreibern) viele Fragen erspart. Außerdem sind einmal gemachte Fehler später nur mit Aufwand zu reparieren, und nicht zuletzt erfordert die Installation von Software in einer heterogenen Umgebung die Einhaltung von Konventionen, um den Aufwand für Updates und Erweiterungen in einem vernünftigen Rahmen zu halten.

An der Carnegie Mellon University wurde unter dem Namen `depot` [12, 4] eine Dateisystemorganisation entwickelt, die es erlaubt, Software für verschiedene Hardware-Architekturen und in verschiedenen Versionen zentral über NFS und AFS bereitzustellen. Neuere Erweiterungen unterstützen die Verlagerung der häufig benötigten Software auf lokale Platten, um die Zugriffszeiten und die Abhängigkeit von zentralen Servern zu reduzieren [18].

Das `depot` System geht davon aus, daß Dateisysteme mit Hilfe eines Automounters bei Bedarf in das logische Dateisystem eingehängt werden können. Im Gegensatz zum statischen Importieren von Dateisystemen während des Systemstarts ermöglicht ein Automounter, daß Teilbäume nur bei Bedarf eingehängt werden, und daß unbenutzte Teilbäume wieder losgelassen werden. Dadurch ist es möglich, Dateibäume von vielen Servern verfügbar zu machen, auch wenn sie nur relativ selten benutzt werden, und trotzdem das System bei Ausfall eines der Server nicht gleich aufzuhängen. Ein anderer Vorteil eines Automounter ist, daß die importierten Teilbäume nach einem konsistenten Schema zusammengehängt werden, wodurch die Gefahr verringert wird, daß der Dateibaum zum Gestrüpp wird.

Die meisten modernen Unix-Varianten werden bereits mit einem Automounter ausgeliefert. Oft handelt es sich dabei aber um veraltete oder gar fehlerhafte Versionen, und es wird nur wenig Unterstützung für heterogene Netze geboten. Abhilfe schafft hier der am Imperial College in London entwickelte `amd`, der Teil des 4.4BSD geworden ist [2, 14]. Bei der Entwicklung des `amd` wurde der Einsatz in heterogenen Systemen von vorneherein berücksichtigt. Er bietet u.a. die Möglichkeit, Server und zu importierende Teilbäume in Abhängigkeit von der Systemarchitektur, von der Domain oder der Subnetadresse auszuwählen, so daß z.B. verschiedene Rechnertypen mit der gleichen Konfigurationsdatei automatisch die passenden X11-Programme importieren können. Ein weiterer Vorteil des `amd` ist, daß die Server regelmäßig überprüft und ausgefallene Server ignoriert werden. Bei replizierten Dateibäumen wird in so einem Fall transparent auf einen anderen Server umgeschaltet. Wir benutzen diese Möglichkeit, um die per `rdist` (siehe unten) replizierte Software unter `/usr/local` verfügbar zu machen und einen Lastausgleich zwischen den Servern zu ermöglichen.

Mit dem Hilfsprogramm `fsinfo` kann man aus einer Beschreibung der vorhandenen Rechner und der von ihnen angebotenten Dateibäume für alle beteiligten Rechner die Konfigurationsdateien (`fstab`, `exports`, Automounter-Maps) erzeugen. Eine andere Ergänzung zum `amd` ist der `hlfsd` [19], der es ermöglicht, Mail relativ sicher in das Homedirectory des Benutzers auszuliefern, statt daß er sie von allen Rechnern des LANs zusammensuchen muß.

Zusammen mit einer Shell, die sich den logischen Pfadnamen zum aktuellen Verzeichnis merkt (z.B. `bash`), läßt sich damit eine relativ gute Transparenz in einem NFS Filesystem erreichen. Unsere Benutzer merken oft erst nach mehreren Monaten, daß sich das Dateisystem, das sie benutzen, aus vielen Stücken von verschiedenen Servern zusammensetzt. Allerdings haben wir teilweise auch Probleme mit der Robustheit des `amd`, deren Ursache wir noch nicht lokalisieren konnten.

Das Programm `rdist` ist seit 1986 Bestandteil des 4.3BSD UNIX und wurde 1992 für 4.4BSD neu überarbeitet [5]. `rdist` dient der Verteilung von Dateien auf mehreren Rechnern, wobei `rdist` für den Erhalt der Zugriffsrechte sorgt. Änderungen an Dateien werden automatisch erkannt und die betroffenen Kopien aktualisiert. Wir benutzen `rdist` seit mehreren Jahren um die lokal installierte Software auf einer zweiten Platte zu spiegeln, was zum einen die Last auf zwei `/usr/local` Server aufteilt und andererseits bei einem Serverausfall einen schnellen Wechsel erlaubt.

Eine etwas andere Art der Softwareverteilung ist mit Hilfe des `perl`-Skripts `mirror` möglich.

Das `mirror` Skript benutzt das FTP Protokoll und ist insbesondere für die Betreiber von FTP Servern wichtig, da automatisch ein Abgleich zwischen der lokalen und der Originalversion vorgenommen wird.

Hinter dem Namen `amanda` [8] verbirgt sich der Advanced Maryland Automated Network Disk Archiver, ein Backup-System, mit dem man viele Platten über ein Netzwerk automatisch auf ein Exabyte Laufwerk sichern kann. Um einen hohen Datendurchsatz zu erreichen, können die Daten komprimiert und über einen Spoolbereich auf das Bandlaufwerk geschrieben werden. `amanda` übernimmt die Planung der nächtlichen Backups nach vorgegebenen Richtlinien wie die Mindesthäufigkeit von vollständigen Backups oder der Wichtigkeit einzelner Partitionen.

Drucker

Die Verwaltung von Druckern kann eine Menge Zeit verschlingen, insbesondere wenn jeder Drucker im Netz von überall erreichbar sein soll. Wir benutzen seit einigen Jahren an unserem Institut den Public Line Printer Spooler `plp`, eine erweiterte Re-Implementation der Berkeley `lpd` Software, die sich auf vielen UNIX-Varianten übersetzen läßt. Da beim Entwurf des `plp` besonders Wert auf Netzwerktransparenz gelegt wurde, konnten wir mit dem `plp` auf allen Rechnern einen einheitlichen Zugang zu den Druckern im Netz erreichen. Druckaufträge können nicht nur netzwerktransparent abgegeben werden, sondern auch Warteschlangen abgefragt, Drucker gestartet und angehalten oder Durckaufträge übers Netz gelöscht werden. Zusätzlich erlaubt der `plp`, daß sich mehrere Rechner ein Spooldirectory über NFS teilen können, wobei dann nur auf dem Serverrechner ein Spooldemon zu laufen braucht. Diese Möglichkeit haben wir mit der dynamischen Serverwahl des `amd` kombiniert, indem wir die `plp`-Spooldirectories als replizierten Teilbaum konfiguriert haben.

Der `plp` bietet außerdem einen einfachen aber wirkungsvollen Mechanismus um den Benutzern Prioritäten und Privilegien für bestimmte Drucker zuzuordnen. Damit haben wir für Teilnehmer an Programmierübungen den Zugang zu den Laserdruckern gesperrt, während die Übungsleiter volle Kontrolle über die Praktikumsdrucker haben, ohne root-Passworte oder zusammengehackte `setuid`-Programme. Sehr angenehm ist auch, daß wir dem `plp` leicht beibringen konnten, Postscript-, DVI-, pnm-, GIF-Dateien automatisch zu erkennen, statt unnütz viele Seiten auszudrucken, weil der Benutzer beim `lpr`-Kommando eine magische Option vergessen hat.

Mail

Die Konfiguration einer zuverlässigen E-Mail Zustellung ist aufwendig und sehr von den lokalen Gegebenheiten abhängig. Wir benutzen `sendmail` [6] und haben nur wenig Erfahrungen mit `smail` oder anderen Mail-Transfer-Agents. Alle unserer Rechner bis auf einen "mailhost" benutzen die vom Hersteller mitgelieferte `sendmail`-Konfiguration. Alle Mail, die nicht lokal zustellbar ist, wird an den mailhost weitergereicht. Auf diesem läuft eine selbstübersetzte `sendmail`-Version mit einer angepaßten Konfiguration (siehe auch Kommentare zum X11). Das Einrichten der Konfiguration hat zwar mehr Arbeit gemacht, aber dafür tut sie genau das, was wir wollen, und die Wartung wird erheblich erleichtert, da die dazu passende `sendmail`-Version bei Ausfall des Mailhosts oder Betriebssystemwechsel einfach neu übesetzt werden kann.

Als nützlich haben sich außerdem bei uns Programme zur Auflösung von E-Mail Adressen erwiesen. `expn` ist ein `perl`-Skript, was dies leistet, und `vrfy` ein entsprechendes C Programm.

Allerdings funktioniert das ganze natürlich nur im Internet und auch da nur bei halbwegs vernünftigen SMTP Servern.

Ein weiteres hilfreiches `perl`-Skript erlaubt die halbautomatische Verwaltung von Mailinglisten und trägt den Namen `majordomo` [3]. Benutzer können sich automatisch in Listen ein- und austragen, und es lassen sich auch die meisten administrativen Maßnahmen per E-Mail durchführen.

Sicherheit

Der Sicherheitsproblematik wird in den letzten Jahren wesentlich mehr Aufmerksamkeit gewidmet. Eine gute Sammlung von Informationen zu diesem Thema findet man in [10] und [7].

Ein einfaches Programmpaket zur Überprüfung der Sicherheit eines UNIX Systems ist das Computer Oracle Password and Security System `cops`, eine von Dan Farmer zusammengestellte Sammlung von C-Programmen und Shell-Skripten verschiedener Autoren, die jeweils spezielle Aspekte des UNIX Sicherheitssystems prüfen [9]. `cops` erkennt (oder besser knackt) zu einfache Passwörter, durchsucht das System nach allgemeinen Schreibzugriffsrechten auf Systemdateien und entdeckt `setuid` Zugriffsrechtsveränderungen an Systemdateien. Dies sind sehr einfache, aber doch recht wirkungsvolle Maßnahmen.

Benötigt man mehr Sicherheit in einem UNIX Netzwerk, so ist das am MIT entwickelte Authtifizierungssystem `kerberos` der zur Zeit wohl am meisten verwendete Ansatz. Seit einiger Zeit wird an der Version 5 gearbeitet, zu dem auch ein RFC erschienen ist [11]. Die Installation von `kerberos` ist relativ aufwendig und erfordert, daß alle Applikationen, bei denen Authentifizierung notwendig erscheint, an `kerberos` angepaßt werden müssen. Daher wird man `kerberos` nur in extrem sicherheitskritischen Bereichen einsetzen.

Eine weniger aufwendige Methode ist das Protokollieren von Zugriffen über das Internet, so daß unerlaubte Zugriffe entdeckt und entsprechende Aktionen eingeleitet werden können. Genau das leistet der `tcp_wrapper` [16] auf sehr einfache Art und Weise: Anstatt Server-Programme automatisch vom `inetd` zu starten, wird ein Hilfsprogramm gestartet, das eine Zugriffsentscheidung trifft, entsprechende Protokolle anlegt und schließlich den aufgerufenen Server startet. Bei einigen UNIX-Varianten werden ähnliche Funktionalitäten bereits vom Hersteller mitgeliefert.

Ein ganz anderes Sicherheitsproblem ergibt sich aus den uneingeschränkten Rechten des Benutzers `root`. In größeren Umgebungen ist es erforderlich, die Kontrolle über Systemteile an einzelne Mitarbeiter zu delegieren, ohne dabei sämtliche Rechte (d.h. das `root`-Passwort) weitergeben zu müssen. In diesen Situationen kann man mit `super` oder `sudo` bestimmten Benutzern erlauben, ausgewählte Programme unter `root`-Rechten kontrolliert laufen zu lassen.

Fehlersuche

Zur Fehlersuche sind eine ganze Reihe von Hilfsmitteln für uns unverzichtbar geworden. Leider laufen nicht alle hier erwähnten Programme auf allen UNIX-Varianten. Das vielleicht wichtigste Programm ist `top`. Es zeigt periodisch die aktuell laufenden Prozesse zusammen mit einigen Statusinformationen an. Ohne `top` hat man oftmals Schwierigkeiten eine Vorstellung zu bekommen, was eine Maschine überhaupt treibt.

Hat man ein Programm identifiziert, das sich merkwürdig verhält oder seinen Dienst verweigert, ohne brauchbare Fehlermeldungen zu hinterlassen, so kann man mit `strace` die Systemaufrufe des Prozesses verfolgen. Dies ist oft die einzige Möglichkeit herauszubekommen, warum Programme, die nur in Binärform vorliegen, nicht richtig laufen. Oft kann mit Hilfe von `strace` leicht ausfindig gemacht werden, wo ein Programm z.B. nach Konfigurationsdateien sucht (und sie dann nicht findet). Leider ist `strace` nicht auf allen Maschinen zu installieren.

Eine andere Möglichkeit, Informationen über geöffnete Dateien zu erhalten, bieten die Programme `fuser`, `ofiles` und `lsof`, wobei `lsof` aus dem `ofiles` abgeleitet wurde und wohl die meisten Möglichkeiten bietet. Es werden Ausgaben für lokale Dateien als auch für Internet-Verbindungen und NFS Dateien erzeugt.

Das wohl wichtigste Programm zur Fehlersuche bei IP Verbindungsproblemen ist das Programm `traceroute`. Es verschickt UDP Pakete mit wachsender TimeToLive an einen (hoffentlich) unbenutzten Port und interpretiert die empfangenen ICMP Fehlermeldungen. Damit lassen sich sehr leicht Routing-Schleifen oder ausgefallene Rechner auf einer Route lokalisieren.

Ist ein Zielrechner erreichbar, aber trotzdem ein angeforderter Dienst nicht benutzbar, so kann man mit Hilfe von `tcpdump` den Nachrichtenaustausch zwischen den beteiligten Rechnern verfolgen. Dies ist oft die einzige Möglichkeit, die Versender von illegalen oder unerwünschten Nachrichten herauszufinden. Außerdem lassen sich auch mit Hilfe von `tcpdump` falsch konfigurierte Maschinen auffinden, die unsinnige ARP Packete oder unerwünschte Broadcasts auf das lokale Netz werfen.

Wenn man speziell an der Leistungsfähigkeit von NFS Servern interessiert ist, so kann man mit Hilfe von `nfswatch` Statistiken über den NFS Verkehr erfassen. `nfswatch` hört das zugrundeliegende LAN ab und ermittelt aus den NFS RPC Packeten die Antwortzeiten der NFS Server. `nfswatch` kann sowohl interaktiv oder als Dämon im Hintergrund betrieben werden, womit sich auch Statistiken über längere Zeiträume problemlos erstellen lassen.

Ein integriertes Werkzeug zur Analyse von Netzwerkproblemen und zur Dokumentation der Netzwerkkonfiguration bietet der `tkined` Netzwerkeditor, der durch den erweiterten `tcl` Interpreter `scotty` programmierbar ist und eine Reihe von Standardwerkzeugen bereitstellt. Unter anderem existiert ein SNMP MIB Browser, mit dem Inhalte von SNMP (Simple Network Management Protocol) Variablen abgefragt und angezeigt werden können, und ein Skript zur Ermittlung von Netzwerktopologien [15].

Weitere Informationsquellen

Viele der hier vorgestellten Werkzeuge werden über das Usenet angekündigt und auch verbreitet. Folglich ist der Zugang zu Usenet News enorm wichtig, um auf dem aktuellen Stand zu bleiben. Für viele der kleinen täglichen Probleme existieren Lösungen, die oftmals auf einer der Usenix Tagungen, insbesondere der Large Installation System Administration (LISA) präsentiert wurden. Die Tagungsbände sind zu erträglichen Preisen von der Usenix Association zu beziehen und trotz des technischen Inhalts oft unterhaltsam zu lesen.

Literatur

[1] T.J. Barret. Simulating Fault Tolerance through Network Design. In *Proc. World Conference On Tools and Techniques for System Administration, Networking, and Security*, Arlington, (Virginia), April 1993.

[2] B. Callaghan und T. Lyon. The Automounter. In *Proc. of USENIX Winter Conference*, S. 43–51. USENIX Association, 1989.

[3] D.B. Chapman. Majordomo: How I Manage 17 Mailing Lists Without Answering '-request' Mail. In *Proc. of LISA VI*, S. 135–143. USENIX Association, 1992.

[4] W. Colyer und W. Wong. Depot: A Tool for Managing Software Environments. In *Proc. of LISA VI*, S. 153–159. USENIX Association, 1992.

[5] M.A. Cooper. Overhauling rdist for the '90s. In *Proc. of LISA VI*, S. 175–188, Long Beach, CA, Oktober 1992. USENIX Association.

[6] Bryan Costales, E. Allman und N. Rickert. *Sendmail.* O'Reilly & Associates, Inc., Sebastopol, CA, 1993.

[7] D.A. Curry. Improving the Security of your Unix System. Technical Report ITSTD-721-FR-90-21, SRI International, 1990.

[8] J. da Silva und Ólafur Guðmundsson. The Amanda Network Backup Manager. In *Proc. of LISA VII*, S. 171–182. USENIX Association, 1993.

[9] D. Farmer und E.H. Spafford. The cops security checker system. In *USENIX Conference Proceedings*, S. 165–170, Anaheim, CA, Summer 1990. USENIX.

[10] P. Holbrook und J. Reynolds. Site Security Handbook. RFC 1244, CICNet, ISI, Juli 1991.

[11] J. Kohl und C. Neuman. The Kerberos Network Authentication Service (V5). RFC 1510, Digital Equipment Corporation, ISI, September 1993.

[12] K. Manheimer, B.A. Warsaw, S.N. Clark und W. Rowe. The Depot: A Framework for Sharing Software Installation Across Organizational and UNIX Platform Boundaries. In *Proc. of Lisa IV*, 1990.

[13] J.K. Ousterhout. *Tcl and the Tk Toolkit.* Addison-Wesley, April 1994.

[14] J.S. Pendry und N. Williams. Amd – The 4.4 BSD Automounter. Bericht, Imperial College of Science, Technonoly and Medicine, London, März 1991.

[15] J. Schönwälder und H. Langendörfer. How To Keep Track of Your Network Configuration. In *Proc. 7 th Conference on Large Installation System Administration (LISA VII)*, S. 189–193, Monterey (California), November 1993.

[16] W.Z. Venema. TCP WRAPPER, network monitoring, access control and booby traps. In *Proc. of UNIX Security Symposium III.* USENIX Association, September 1992.

[17] L. Wall und R. Schwartz. *Programming Perl.* O'Reilly, 1991.

[18] W.C. Wong. Local Disk Depot – Customizing the Software Environments. In *Proc. of LISA VII*, S. 51–55. USENIX Association, 1993.

[19] E. Zadok und A. Dupuy. HLFSD: Delivering Email to Your $HOME. In *Proc. of LISA VII*, S. 243–254. USENIX Association, 1993.

FTP Quellen-Verzeichnis

Die folgende Liste gibt an, wo jeweils das Original eines der oben beschriebenen Programme bezogen werden kann. Natürlich ist alles auch direkt in Deutschland verfügbar, so daß diese Liste keine Aufforderung sein soll, internationale Leitungen unnötig zu belasten.

Paket	FTP Server	Dateiname
`GNU, perl`	prep.ai.mit.edu	/pub/gnu
`X11R6`	ftp.x.org	/pub/R6
`X11R6`	ftp.germany.eu.net	/pub/X11/XConsortium/pub/R6
`tcl`	ftp.cs.berkeley.edu	/ucb/tcl/tcl7.3.tar.Z
`tk`	ftp.cs.berkeley.edu	/ucb/tcl/tk3.6.tar.Z
	harbor.ecn.purdue.edu	/pub/tcl
`amd`	usc.edu	/pub/amd/amd.tar.Z
`rdist`	usc.edu	/pub/rdist/rdist-6.1.0.tar.Z
`mirror`	src.doc.ic.ac.uk	/pub/packages/mirror/mirror-2.3.tar.gz
`amanda`	ftp.cs.umd.edu	/pub/amanda/amanda-2.1.1.tar.gz
`plp`		plp-3.3-dist.tar.Z
`sendmail`	FTP.CS.Berkeley.EDU	/ucb/sendmail/sendmail.8.6.9.*.tar.Z
`smail`	ftp.uu.net	/networking/mail/smail/smail-3.1.28.tar.Z
`majordomo`	FTP.GreatCircle.COM	/pub/majordomo/majordomo.1.92.tar.Z
`expn`	idiom.berkeley.ca.us	/pub/muir-programs/expn
`vrfy`	ftp.nikhef.nl	/pub/network/vrfy.tar.Z
`kerberos`	ATHENA-DIST.MIT.EDU	/pub/kerberos/dist/940615
`tcp_wrapper`	ftp.win.tue.nl	/pub/security/tcp_wrappers_6.3.shar.Z
`sudo`	ftp.cs.colorado.edu	/pub/sysadmin/utilities/cu-sudo.v1.3.1pl1.tar.Z
`super`	astro.caltech.edu	/users/will/super-3.4.9.shar
`top`	eecs.nwu.edu	/pub/top/top-3.2.tar.gz
`strace`	ftp.cs.few.eur.nl	/pub/sun/strace.tar.Z
`lsof`	vic.cc.purdue.edu	/pub/lsof_3.02.tar.gz
`tcpdump`	ftp.ee.lbl.gov	/tcpdump-3.0.tar.Z
`nfswatch`	gatekeeper.dec.com	/pub/net/ip/nfs/nfswatch4.1.tar.Z
`tkined`	ftp.ibr.cs.tu-bs.de	/pub/local/tkined/tkined.tar.gz
`scotty`	ftp.ibr.cs.tu-bs.de	/pub/local/tkined/scotty.tar.gz