

Was bietet Linux?

Linux ist in aller Munde. Im Netzwerkbereich hat Linux sich erfolgreich etabliert und gilt als feste Größe. Für Software-Entwicklung bietet Linux ebenso viele Vorzüge wie für (heterogene) Netzwerke.

Linux im Netzwerk

Ohne das Internet gäbe es kein Linux - ohne Linux wäre das Internet gerade in Deutschland sicherlich weniger interessant. In der Bundesrepublik werden bereits über **40% aller Webserver** in der Kombination "Linux mit Apache" betrieben. Im weltweiten Durchschnitt laufen ca. 25% der News- und FTP-Server unter Linux.

Die Einsatzgebiete für Linux im Netzwerkbereich sind vielfältig: Von Intranetservern bis hin zu vollwertigen, leistungsfähigen Internetservern, vom Gateway für ein mittelständisches Unternehmen bis zum ISDN-Router reicht das Spektrum.

Internetserver

Ein beliebtes Einsatzgebiet sowohl im industriellen als auch universitären Umfeld ist Linux in Kombination mit **Apache als WWW-Server**. Als Frontend auf der Clientseite arbeiten Sie wie gewohnt mit dem **Netscape Navigator**. Da Ihnen von all diesen Werkzeugen (ausser den kommerziellen Datenbanken) die Quellen der Software zur Verfügung stehen, können Sie beliebige Anpassung vornehmen. Email-Dienste sind inzwischen ein Alltagswerkzeug in jedem Unternehmen geworden.

Linux liefert **unterschiedliche SMTP-Server** mit: Ihnen stehen von dem sehr weit verbreiteten **sendmail** bis **qmail** alle möglichen, **relay-festen** SMTP-Server zur Verfügung. Ergänzende Filtermechanismen wie **procmail** arbeiten außerdem mit **IMAP und POP3-Servern** zusammen, um nicht nur die Zustellung, sondern auch die Abholung von Emails sicherzustellen. Im Office-Einsatz können diese Dienste z.B. mittels LDAP auch heterogenen Umgebungen zur Verfügung gestellt werden.

Gateways/Router

Linux ist für den Netzwerkeinsatz optimiert: Linux bringt von Haus aus einen kompletten TCP/IP-Stack mit, auf den Bootprotokolle wie **DHCP oder bootp** aufsetzen. Beide Protokolle ermöglichen die dynamische Zuteilung von IP-Adressen an Clients im Netzwerk und sichern den flexiblen Office-Einsatz.

TCP/IP umfasst unterschiedliche Routing-Protokolle, die ebenfalls unter Linux verfügbar sind. Nameservices werden mit **DNS/bind** realisiert.

Sämtliche Services liegen im Quellcode vor und können auf Sicherheit getestet und geprüft werden. Linux beherrscht ausserdem IP-basierte Mechanismen wie **Masquerading**, Paket-Forwarding usw. **IPSec und IPv6** wird unterstützt und ergänzt den Aufbau von **VPN** (Virtual Private Networks) und verschlüsselten Tunneln. Linux beherrscht mit Kernel 2.2.x priority-basiertes Routing.

Intranetserver

Die Möglichkeiten, die Linux als Intranetserver bietet, kommen einem Intranet ebenfalls zu Gute: Linux unterstützt Protokolle anderer Betriebssysteme und kann damit **Netware-, Windows und Macintosh-Clients** bedienen. Durch Samba kann Linux auf Basis des SMB-Protokolls einen NT-Domänenserver komplett ersetzen.

Mit der **IPX-Unterstützung (Netware)** können Linux-Server (oder Clients) in ein bereits bestehendes Netware-Netz integriert werden. Auch Macintosh-Clients können die Möglichkeiten eines Print, Fax- oder Fileservers nutzen, denn Linux bietet mit der Software "**netatalk**" Unterstützung für das Protokoll Apple-Talk. Für die Datensicherung setzen Sie unter Linux unterschiedliche Backup-Lösungen ein. Sämtliche Dienste können ausserdem auf Basis von RAID-Systemen gelöst werden, wenn Sie eine hohe Ausfallsicherheit benötigen.

Linux im Office-Einsatz

Linux kann für EDV-Anwender mit **Officepaketen wie StarOffice** oder Wordperfect inzwischen auch als Büro-Arbeitsplatz eingesetzt werden. Serverseitig bietet Linux mit Produkten wie **OpenLDAP** die Möglichkeit, sowohl Benutzerrechte, Nameservice, Email-Daten und Adressverwaltungen zentral mit zu verwalten und von überall zugreifbar zu halten.

Unterschiedliche Szenarien können mit Linux realisiert werden: Internet-Server auf Linuxbasis mit heterogenen Clients, Intranet-Fähigkeiten mit LDAP, abgestufte Datenzugriffe mit Verschlüsselungsmechanismen und Zugriffsrechten, gemeinsames **File-Sharing**, zentrale Druckdienste oder **organisiertes Dokumentenmanagement**. Diese Fähigkeiten können beliebig mit der **Netzwerktechnik kombiniert** werden; z.B. über verschlüsselte Tunnel, die unterschiedliche Standorte sicher miteinander verbinden.

Firewalls und Paketfilter

Linux bietet sich bis zu einem gewissen Einsatzgrad als Firewall an. Firewalling unter Linux arbeitet auf Basis der **Paketfilter ipfwadm und ipchains**. Anhand von Rulesets werden Regeln aufgestellt, die **Protokoll-, Dienst-, Netz-, Port-**

und IP-abhängig arbeiten. Pakete können fragmentiert geprüft werden. Die Regeln erlauben eine Kombination von unterschiedlich abgestuften Ge- und Verboten und können komplett protokolliert werden.

Monitoring

Für Sicherheitsanforderungen stehen Ihnen **Überwachungswerkzeuge** zur Verfügung, die sowohl die Erreichbarkeit eines Servers als auch die Systemintegrität überwachen. Zum Test der Sicherheit können Sie mit **Diagnosewerkzeugen** arbeiten, die potentielle Schwachstellen überprüfen und melden.

Verschlüsselung

Anwendern und Anwenderinnen steht in Office-Umgebungen die Möglichkeit offen, Daten verschlüsselt zu transportieren: **Sicherer Email-Verkehr** (PGP), **digitale Signaturen** oder partiell **verschlüsselte Filesysteme** (cryptoFS) garantieren den Schutz vertraulicher Unternehmenskommunikation.

Software entwickeln mit Linux

Die **offenen Quellen** von Linux ermöglichen sinnvolle Weiterentwicklungen nach den Bedürfnissen des Projekts und liefern viele Millionen Zeilen Code zum beispielhaften Nachvollziehen der Programmstruktur. Die Werkzeugervielfalt eines Unix-Systems bietet den Entwicklern ineinandergreifende Tools, die je nach Erfordernissen miteinander kombiniert werden können.

Hoch -und Scriptsprachen

Linux bietet vielfältige Sprachen und ergänzende Werkzeuge zur Softwareentwicklung auf Linux und für Linux: Von "Klassikern" wie **Fortran, Scheme oder Pascal** bis hin zu moderneren, objektorientierten Sprachen wie **Java** sind die meisten Compiler oder Interpreter im Quellcode verfügbar. Der Linux-Kernel selbst und die meisten Werkzeuge des Systems sind in **C** programmiert; ein immer größer werdender Teil der Anwendungen liegt inzwischen auch in **C++** vor.

Compiler

Durch diese starke Bindung an C erhalten Entwickler unter Linux besonders vielfältige Unterstützung durch Werkzeuge für diese Sprache. Insbesondere der mitgelieferte **GNU-C-Compiler** bietet außerdem sehr gute Fähigkeiten, plattformübergreifenden bzw. **architektur-unabhängigen Code** zu erzeugen:

Beispielsweise können auf einem PC die Bibliotheken und der Assembler für Embedded Systems-Architekturen installiert werden: Der gcc kann als **Crosscompiler** arbeiten und problemlos den entsprechenden Binärcode erzeugen.

GUI-Bibliotheken

Verschiedene Bibliotheken zur Entwicklung von GUIs stehen ebenfalls zur Verfügung: Das lizenzpflichtige Motif hat sein GPL-Pendant in "**lesstif**" gefunden. Der beliebte Desktop KDE arbeitet mit der Qt-Bibliothek der norwegischen Firma Troll Tech: **Qt** bietet ein sehr leistungsfähiges Set an C++-Bibliotheken, mit der GUI-basierte Anwendungen entwickelt werden können. Sowohl KDE als auch der alternative, **GTK-basierte Desktop GNOME** arbeiten mit Corba: Als Middleware eingesetzt kommen unter Linux **ORBs** in unterschiedlichen Implementationen wie **MICO** oder **Orbit** in Frage.

Scriptsprachen

In der Welt der Scriptsprachen hat sich **Perl** mit seinen besonderen Fähigkeiten im Bereich für Textprocessing/-parsing hervorgetan. Alternative, sehr beliebte Scriptsprachen sind auch das teilweise objektorientierte **Python**, **PHP** für Web-Programmierung oder **Tcl/Tk**.

Tools

Im Office-Einsatz spielen gerade die Skriptsprachen ihre Stärke aus: Kleine Werkzeuge und **flexible "Helferleins"** sind auch von weniger versierten Programmierern und erfahrenen Anwendern zu realisieren. Sprachen wie Perl oder Python sind leicht zu erlernen und liegen nicht nur für Linux/Unix-Systeme vor, sondern auch für Windows- oder Macintosh-Rechner.

Netzwerk-APIs

Durch die umfassende Netzwerkintegration von Linux arbeiten Entwickler bei systemnaher Programmierung mit dem ganzen Spektrum standardisierter Netzwerk-Technologien: **Socket-Programmierung** - mit der neuen glibc auch **Threading** - sind ebenso in das System integriert wie **SUNs RP-Calls**. Javas **RMI**-Technologie funktioniert mit den aktuellen JDKs ebenfalls unter Linux.

Datenbanken

Gerade im Rahmen von Netzwerken ist nicht nur die netztechnische Schicht selbst für Entwickler wichtig: Netzwerkfähige Komponenten wie **Datenbanken** sind unter Linux vielfältig vorhanden: Die Anbieter Software AG (**Adabas**), **Sybase**, **Oracle**, **Informix** oder IBM (**DB2**) ergänzen das Angebot an freien Datenbanken wie z.B. **mySQL** oder **Postgres**. Alle Datenbanken können über die **jeweilige API** mit den unter Linux vorhandenen Sprachen (z.B. **C**, **Java** oder **Perl**) angesprochen werden.

Desktops

Unter Linux ist auch die Software für den Office-Einsatz netzwerktauglich: Sowohl die GUI, mit der der Anwender arbeitet, ist netzwerkfähig als auch die Produkte: Der **einsteigerfreundliche Desktop "KDE"**, StarOffice als Textverarbeitung oder das Grafikprogramm "The Gimp": Alles, was Anwender an ihrem Arbeitsplatz benötigen, kann auch über das Netzwerk gesteuert und benutzt werden.

Werkzeuge und Debugger

Zur **Organisation von Softwareentwicklung** jenseits der reinen Programmiersprachen bietet Linux Werkzeuge in den Bereichen Versioning-Systeme, Debugging-Werkzeuge, Modellierung und natürlich Entwicklungsumgebungen an:

Versioning

Das Versionsmanagement-Tool "CVS" verwaltet die mehreren Millionen Code-Zeilen von Projekten wie KDE, Apache oder XFree, an denen weltweit verteilt arbeitende Entwickler via Internet ihren Sourcecode einpflegen. CVS kann in unterschiedliche Entwicklungswerkzeuge eingebunden werden oder wird von vorn herein (z.B. in Emacs) unterstützt.

Debugging

Zur Fehlersuche stehen den Entwicklern nicht nur das grafische Frontend "ddd" zur Verfügung, der die **Debugger von C (gdb)**, **C++** oder **Perl** bedient, sondern auch Werkzeuge, um sichere Programmierung zum **Schutz vor Buffer Overflows** wie z.B. "Electric Fence". Für einige Programmiersprachen wie C oder Java sind "**Code-Beautifier**" verfügbar, mit denen Sourcecode sauber und

projekt-einheitlich formatiert werden kann. **Parser wie lex und yacc** erleichtern zusätzlich die Arbeit an großen Softwareprojekten.

Make

Mit "Makefiles", die mit dem Werkzeug "make" arbeiten, wird das Kompilieren von Code erleichtert: Komplexe Aufrufe, das **dynamische Linken** von Bibliotheken, **abhängiges Kompilieren**, unterschiedliche Testversionen - all das ergänzt die Arbeit der Softwareentwicklung in passender Weise.

Entwicklungsumgebungen

Grafische Entwicklungsumgebungen wie "**KDeveloper**" oder der dem Code Warrior aus der Mac-Welt nachempfundene "**Code Crusader**" (beide C, C++) sind für diejenigen interessant, die sich mit XEmacs oder gar vi nicht anfreunden können. Einige Java-IDEs in Java (**NetBeans**) programmiert stehen auch in der Linuxwelt zur Verfügung. Viele Werkzeuge kommerzieller Natur wie SNIFF+ können produktiv eingesetzt werden. Einige der beliebtesten IDEs aus der Windows-Welt wie Delphi stehen kurz vor der Portierung. Zur Modellierung von Software steht unter Linux natürlich auch die Arbeit mit **UML** zur Verfügung.