

Barbara Leipholz-Schumacher

Freie Software im Informatikunterricht: Kollege Computer verständlich machen

1. Das Euregio-Kolleg in Würselen

Müssen wir wirklich soviel Geld für Software-Lizenzen ausgeben? Dieser Seufzer vieler Informatiklehrerinnen und -lehrer muss nicht sein, wenn freie und kostenfreie Software genutzt wird. Um dies in die Realität umzusetzen, wird nur etwas Mut und Experimentierfreude benötigt. Dual-Boot-Systeme mit der Möglichkeit, wahlweise Linux oder MS-Windows zu starten, ermöglichen den schrittweisen Umstieg auf Linux. Pädagoginnen und Pädagogen, die sich entscheiden, im Unterricht freie Software einzusetzen, werden feststellen, dass ihnen für ihre Unterrichtsthemen eine Fülle an geeigneten Programmen zur Verfügung steht, die im Unterricht unterstützend eingesetzt werden kann.

Am Euregio-Kolleg (**Kasten 1: Euregio-Kolleg, ist im Anschluss an diesen Text beigefügt**) in Würselen werden seit zirka acht Jahren in Informatikkursen sehr gute Erfahrungen mit dem ausschließlichen Einsatz von freier Software gemacht. Ergänzend zu den ausgewählten Programmen werden kostenfreie Java-Applets eingesetzt, die von Pädagogen oder Universitäten für die Unterstützung der Lehre an Schulen und Universitäten entwickelt wurden. Da der Informatik-Etat nun nicht mehr durch teure Programmlicenzen belastet wird, steht mehr Geld für die Hardware zur Verfügung. Weil keine kostenpflichtigen Probelizenzen angeschafft werden müssen, kann ohne Risiko getestet werden, welche Software für den Unterricht geeignet ist. Dies alles schafft Raum für Innovationsmöglichkeiten im Informatikunterricht.

Definitionen von Software

Im folgenden wird zwischen drei Arten von Software unterschieden. Freie Software unter der GNU-Lizenz (1984 gestartetes Projekt, eine freie, vollständige Unix-artige Betriebsumgebung) wird mit * gekennzeichnet. Kostenfreie, nicht kommerzielle Java-Applets im Internet erhält ** als Kennung und kommerzielle Software, deren Lizenzen generell oder speziell für die Ausbildung kostenfrei sind, wird mit *** gekennzeichnet.

Arbeitsbedingungen am Euregio-Kolleg

Das Netz am Kolleg, welches auf Linux-Basis* für den Informatikunterricht entwickelt wurde, wird mittlerweile auch außerhalb des Informatikunterrichts regelmäßig genutzt. Alle Arbeitsplatzrechner im Netz nutzen zwei Server:

- ⑩ Den Kommunikations-Server, der folgende Dienste für alle Rechner am Kolleg bereitstellt:
- ⑩ Internetzugang über den Proxy-Server Squid*
- ⑩ Kommunikation per Email im lokalen Netz und im Internet über SMTP* und UUCP*.
- ⑩ den lokalen WWW-Server Apache* zur Veröffentlichung von WWW-Seiten im Intranet.
- ⑩ Den Datei-Server, der allen Arbeitsplatzrechnern im Netz sowohl für Linux via NFS* (Net-File-System) als auch für MS-Windows mittels Samba* die Heimatverzeichnisse der Benutzer zur Verfügung stellt. Zusätzlich bietet er CDs im Netz zum Einsatz im Unterricht und zur selbständigen Arbeit der Studierenden an. Fachlehrer und -lehrerinnen können Ordner im Netz nutzen, in denen sie Informationen für die Studierenden bereit stellen und die gemeinsame Arbeit im Netz ermöglichen. Durch die Zugriffsrechte von Linux können alle Ordner und Dateien entsprechend gesichert werden.

Alle Arbeitsplatzrechner können wahlweise unter Linux oder MS-Windows gestartet werden. Im praktischen Betrieb zeigt sich, dass die Wartung von Linux mit wesentlich geringerem Zeitaufwand verbunden ist als die Wartung von Windows. Da aber einige Pädagogen und Pädagoginnen Software von Schulbuchverlagen nutzen, die nur unter Windows läuft, ist es zur Zeit leider nicht möglich, auf das Betriebssystem von Microsoft ganz zu verzichten.

Keine eigenen Rechner sind kein Problem für Schüler

Im allgemeinen haben weder Lehrerinnen und Lehrer noch Studierende an Schulen eigene Rechner, als Ersatz sollte ihnen an jedem Computer ihr virtueller persönlicher Arbeitsplatz zur Verfügung gestellt werden. Aus diesem Grund erhalten alle Studierenden und alle Pädagogen am Euregio-Kolleg einen persönlichen Account (**Kasten 2: Kurs für pädagogische Schulnetzbetreuerinnen und -betreuer, Text ist im Anschluss beigefügt**).

Nach Anmeldung mit Benutzerkennung und Passwort

☉ finden sie ihren persönlichen Desktop mit ihren eigenen Dateien an jedem beliebigen Rechner am Kolleg vor.

☉ können Lehrerinnen und Lehrer Dateien für den Unterricht auf den PC des Lehrerzimmers kopieren und anschließend im entfernt liegenden Unterrichtsraum verwenden.

☉ haben Studierende, die während der Unterrichtsstunde eine Arbeit begonnen haben, die Möglichkeit, diese in einer anschließenden Freistunde im Selbstlernzentrum des Kollegs fortsetzen.

☉ stehen Lesezeichen, die bei der Internet-Recherche im Browser gesetzt werden, auch später an anderen Rechnern zur Verfügung.

☉ kann das eigene Emailkonto überall genutzt werden.

Von den verschiedenen gängigen Desktops wird unter Linux KDE* eingesetzt – das K Desktop Environment. Alle im folgenden genannten Programme, die mit einem K beginnen, wurden speziell für KDE entwickelt.

Der Informatikunterricht

Im folgenden wird am Beispiel des Informatikunterrichts am Euregio-Kolleg gezeigt, wie mit freier und kostenfreier Software gearbeitet wird. Da die Schwerpunktthemen des Unterrichts auf mehrere Semester verteilt sind, werden hier die Inhalte thematisch gegliedert. Der Kasten „Informatik am Euregio-Kolleg“ (Kasten 3, ist an diesen Text angehängt) zeigt den gesamten Unterrichtsgang. Alle dargestellten Themen eignen sich auch für den Unterricht an Gymnasien.

Einführung in die Handhabung des PCs

Im 2. Semester erhalten die Studierenden eine Einführung in die Nutzung der PCs, des Intranet und des Internet. Mit ihrer Benutzerkennung und Passwort melden sie sich an und finden als Desktop KDE vor. Sie lernen Internetrecherchen durchzuführen und gefundene Informationen zu bewerten. Um relevante Web-Seiten zur Wiederverwendung zu speichern, legen sie in ihrem Heimatverzeichnis geeignete Ordner an, in denen die Dateien übersichtlich

angeordnet werden. Speichern und Wiederfinden von Dokumenten erfolgt mit Konqueror*, dem KDE-Explorer. Anschließend werden die erarbeiteten Ergebnisse in Web-Dokumenten präsentiert und können auf dem Server im Intranet zur Verfügung gestellt werden. Gearbeitet wird mit dem Netscape-Navigator*** zur Internetrecherche, dem Netscape-Messenger*** oder Kmail* zur Kommunikation per Email und dem Netscape-Composer* zur Erstellung von HTML-Seiten.

Der Einsatz des schulinternen Mail-Servers bewirkt, dass Email unmittelbar nach dem Versenden vom Empfänger abgerufen werden kann, so dass keine Zeitverzögerung im Unterricht entsteht. Zu Officeanwendungen wie Textverarbeitung, Präsentation, Tabellenkalkulation usw. wird StarOffice 5.2*** eingesetzt, das in seiner Funktionalität mit MS-Office vergleichbar ist. Eine freie Alternative ist OpenOffice*.

Einführung in Aufbau und Funktionsweise eines PCs

Die Studierenden lernen, wie Informationen intern dargestellt werden. Zur Überprüfung der Umrechnung von Zahlen zwischen dem Dezimal, Dual-, Oktal- und Hexadezimalsystem wird der KDE-Taschenrechner Kcalc* genutzt. In der anschließenden Unterrichtsreihe zur Schaltalgebra können logische Schaltungen mit dem KDE-Programm Klogic* entworfen und simuliert werden. Ein Java-Applet** zur Simulation eines von Neumann-Rechners, der es ermöglicht, selber kleine Programme zu schreiben und zu testen, rundet die Unterrichtsreihe ab.

Einführung in die Nutzung eines Betriebssystems

Nach einer allgemeinen Einführung in die Aufgaben eines Betriebssystems wird in das Dateisystem unter Linux sowie das Konzept der Zugriffsrechte eingeführt. Die Studierenden versuchen nun, in die Heimatverzeichnisse ihrer Mitstudierenden zu wechseln und herauszufinden, welche Dateien sie lesen, kopieren, verändern oder löschen können. Dies motiviert sie, ihre Ordner gegenüber Zugriffen durch andere zu sichern. In einen gemeinsamen Ordner kopieren die Studierenden solche Dateien, die von den anderen gelesen, aber nicht geändert werden können. Dies befähigt die Studierenden, auch in anderen Fächern im Netz gemeinsam arbeiten zu können.

Einführung in eine Programmiersprache

Zur Einführung in die Programmierung entschieden sich die Informatiklehrerinnen und -lehrer am Euregio-Kolleg für das Objektorientierte Paradigma. Unter Linux kann aus einer Reihe von plattformunabhängigen, problemorientierten Programmiersprachen mit geeigneten Entwicklungsumgebungen gewählt werden. Als gängige objektorientierte Programmiersprachen für Schulen stehen Oberon*, C++*, Java*** und Python* zur Verfügung. Nach dem vergleichenden Einsatz von C++, Python und Java im Unterricht, fiel die Entscheidung für Java. Als Entwicklungsumgebung zur Erstellung der Java-Applets und Applikationen wird entweder mit dem freien Emacs* gearbeitet oder mit Forte***, der kostenfreien visuellen Java-Entwicklungsumgebung von SUN. Von den Studierenden entwickelte Java-Applets können auf dem lokalen WWW-Server im Intranet veröffentlicht werden.

Zur Einführung in objektorientierte Konzepte wird die verbreitete Modellierungssprache UML (Unified Modelling Language) verwendet. Das Software-Tool Together***, das objektorientierte Analyse und Design unterstützt, hat sich im Unterricht gut bewährt. Als freie Software steht ArgoUML* zur Verfügung. Das KDE-Programm Kuml* befindet sich in der Entwicklung und ist für den Unterrichtseinsatz zur Zeit noch nicht geeignet.

Kylix, das Delphi für Linux von Borland basiert wie Delphi auf Object Pascal und erleichtert Delphi-Programmierern die Umstellung auf Linux. Die Kylix Open Edition*** ist kostenfrei, die Einschränkung, dass damit erstellte Software nur unter der GNU-Lizenz vertrieben werden darf, bedeutet für Schulen keinen Nachteil. Für eine Unterrichtsreihe in logischer Programmierung stehen verschiedene freie Prolog-Versionen zur Verfügung, wie SWI-Prolog* und GNU-Prolog*.

Computer-Netzwerke

1. Grundlagen

Mit der zunehmenden Vernetzung der PCs am Arbeitsplatz sowie im privaten Bereich spielen Sicherheitsrisiken in Netzwerken eine immer größere Rolle. Die Themen Netzwerke und -

sicherheit wurden deswegen in das schulinterne Curriculum integriert. Um Sicherheitsrisiken in Netzen realistisch einschätzen zu können, wird in die Funktionsweise von Netzen eingeführt. Das KDE-Programm Ksnuffle* zeigt detailliert den Netzverkehr an.

Ein Doppelklick auf ein Datenpaket zeigt sämtliche Informationen dieses Paketes. Hieran kann veranschaulicht werden, wie der Datenaustausch im Netz erfolgt, wie die Protokolle TCP und IP arbeiten, aber auch wie jeder Netzverkehr von jedem Rechner, über den die Daten gesendet werden, vollständig überwacht werden kann. Zum Verständnis der Protokolle werden das Hypertext Transfer Protokoll HTTP, das Simple Mail Transfer Protokoll SMTP und das File Transfer Protokoll FTP analysiert. Eine Analyse der Routingtabellen der Rechner im Intranet schließt das Thema ab und leitet über zu Routingverfahren im Internet. Das Kommando traceroute* zeigt den Studierenden, welchen Weg ihre Datenpakete im Netz nehmen und macht ihnen bewusst, wie einfach jeder Datenverkehr im Netz abgehört werden kann.

Auch Algorithmen müssen sitzen

Es schließt sich eine Unterrichtsreihe zu Algorithmen und Datenstrukturen mit dem Schwerpunkt Graphen an. Der Dijkstra-Algorithmus zur Ermittlung des kürzesten Weges und das verteilte Bellman-Ford-Verfahren zeigen, wie Routingtabellen gebildet werden. Eine Sammlung von kostenfreien Java-Applets** zur Simulation der Algorithmen werden von Universitäten im Internet angeboten.

2. Client-Server-Anwendungen

Um das Verständnis für Anwendungen im Netz zu vertiefen, erstellen die Studierenden in Java eigene Client-Server-Anwendungen. Sie entwickeln ein Protokoll zwischen Client und Server, legen den Port des Servers fest und implementieren die Applikationen in Gruppenarbeit.

Sicherheit in Netzen und Kryptografie

1. Einstieg

Einen Einstieg in die Thematik „Sicherheit in Netzen“ bietet die Kommunikation per Email. Hier wird thematisiert, dass Emails genau wie Postkarten von jedem gelesen werden können.

Aus Gründen des Datenschutzes wird den Studierenden an einem Test-Account am Linux-Mail-Server gezeigt, dass die Administratorin des Netzes jede eingehende Mail problemlos lesen, ändern, löschen und kopieren kann. Gleiches gilt für jede ausgehende Mail. Dies überzeugt die Studierenden, dass zumindest jeder Provider, sowohl der des Empfängers, als auch der des Senders Zugriff auf die Mail haben. Das Beispiel ECHELON zeigt, dass das Belauschen elektronischer Kommunikation bereits systematisch und in großem Stil betrieben wird. ECHELON wird von den USA, Kanada, Großbritannien, Australien und Neuseeland gemeinsam unterhalten. Über 120 Stationen hören Satellitenverbindungen und Transatlantikkabel ab, um Daten über Wirtschaftsunternehmen, Regierungen, internationale Institutionen und Individuen zu gewinnen. Auf europäischer Ebene wird im Projekt ENFOPOL eine Maßnahme zur Überwachung des Telekommunikationsverkehrs einschließlich des Internet geplant.

Um die Sicherheit von Sender und Empfänger zu gewährleisten, setzen Nutzung von Email-Kommunikation, E-Commerce, Online-Banking usw. eine Reihe von Bedingungen voraus. Am Beispiel einer Bestellung im Internet werden Lösungsverfahren diskutiert: der Käufer möchte nicht, dass sein Kaufverhalten von anderen beobachtet werden kann, der Anbieter muss sicher sein können, dass der Absender wirklich Autor der Bestellung ist und dass diese Bestellung nicht nachträglich unberechtigt verändert wurde.

Es ergeben sich somit folgende Forderungen:

- ⑩ Vertraulichkeit - die Informationen können nur vom Empfänger gelesen werden
- ⑩ Integrität - die Daten werden genau so empfangen, wie sie vom Absender versendet wurden. Niemand konnte die Nachricht unbemerkt verändern.
- ⑩ Authentizität - der Urheber der Nachricht ist tatsächlich der, der als Absender der Nachricht angegeben wurde.

Diese Forderungen können nur durch den Einsatz von Verschlüsselungsverfahren gewährleistet werden. Ausgehend von historischen Verfahren, wie die Verschlüsselung nach Cäsar, bei dem jedes Zeichen im Text um einen festen Abstand im Alphabet verschoben wurde oder der verbesserten polyalphabetischen Substitution, wo der Text auf ähnliche Weise mittels eines Geheimwortes oder einer Geheimzahl verschlüsselt wird, werden sichere und schnelle symmetrische Verschlüsselungsverfahren wie 3DES und IDEA im Unterricht

besprochen. Auch bei diesen sicheren Verfahren bleibt aber das Risiko der Schlüsselübertragung. Einen anderen Lösungsansatz bieten Public-Key-Verfahren, d.h. jeder Beteiligte an der Kommunikation besitzt einen öffentlichen und einen geheimen Schlüssel. Der Absender verschlüsselt die Nachricht mit dem öffentlichen Schlüssel des Empfängers, die Entschlüsselung kann lediglich vom Empfänger mit seinem geheimen Schlüssel ausgeführt werden. Als typisches Beispiel wird das RSA-Verfahren besprochen. Soll eine Nachricht digital signiert werden, so wird mittels einer Hash-Funktion die Prüfsumme eines Textes gebildet, diese Summe wird mit dem privaten Schlüssel des Absenders verschlüsselt und gemeinsam mit der Nachricht übermittelt. Der Empfänger kann nun ebenfalls den Hashwert des Textes ermitteln, die digitale Unterschrift mit dem öffentlichen Schlüssel des Empfängers entschlüsseln und beide Hashwerte auf Gleichheit überprüfen. Stimmen beide Werte nicht überein, wurde der Text nachträglich unberechtigt verändert.

Die Digitale Signatur verständlich machen

Um Verschlüsselung und digitale Signatur anschaulich vorzuführen, wird mit dem freien Programm GNU P_g* (GNU Privacy-Guard) gearbeitet. Kmail* bietet hier gegenüber dem Netscape-Messenger*** den Vorteil, dass Verschlüsselungsfunktionen von GNU P_G* bzw. PGP*** (Pretty Good Privacy) bereits integriert sind. Um das Programm verwenden zu können, erzeugt jeder Nutzer zuerst sein Schlüsselpaar, einen öffentlichen und einen privaten Schlüssel. Der öffentliche Schlüssel wird exportiert und an alle anderen Studierenden versendet. Diese importieren ihn in ihren virtuellen Schlüsselbund. Nun kann der verschlüsselte und signierte Emailaustausch beginnen. Die Mail an einen Studierenden wird jeweils mit dessen öffentlichem Schlüssel verschlüsselt. Nur der Empfänger kann die Nachricht mit seinem privaten Schlüssel entschlüsseln. Anschließend werden Verfahren zu einer sicheren Authentisierung in Netzen, z.B. beim Online-Banking erarbeitet und diskutiert.

2. Kryptografische Algorithmen

Einige der Verschlüsselungsalgorithmen bieten gute Übungsaufgaben beim Erlernen der Algorithmik. Einfache symmetrische Verfahren wie die Verschlüsselung nach Cäsar, die Verschlüsselung durch ein Geheimwort, aber auch asymmetrische Verfahren wie das RSA-Verfahren können schon im Anfangsunterricht implementiert werden. Als Vorkenntnisse

werden lediglich einfache Wiederholschleifen und der Datentyp String benötigt. Bei der Einführung des RSA-Verfahrens werden Berechnungen mit sehr großen Zahlen notwendig. Diese Berechnungen können problemlos mit dem freien bc* durchgeführt werden, ein Programm, das direkt von der Konsole gestartet wird und Rechenoperationen mit beliebiger Stellenzahl durchführt.

3. Cookies

Als weiteres Sicherheitsrisiko wird die Protokollierung des Surfverhaltens eines Netznutzers thematisiert. Die Studierenden analysieren ihre Cookies-Datei im Ordner `.netscape/cookies` und sehen, dass viele Internet-Server für sie eine eindeutige ID erzeugt haben, selbst Webserver, die sie nach ihrem Wissen niemals besucht haben. Hier hilft es, nur die unerwünschten Einträge zu löschen. Jeder Browser bietet zwar die Möglichkeit, das Setzen von Cookies zu verbieten, dies hat aber zur Folge, dass viele Webseiten nicht angezeigt werden. Zur Demonstration erzeugt ein CGI-Skript auf dem WWW-Server im Intranet für jeden Studierenden eine eigene ID und zeigt den Studierenden bei jedem Zugriff auf die Seite an, wie häufig sie diese Seite bereits besucht haben und wann der letzte Besuch war. In einer Diskussion über Vor- und Nachteile von Cookies zeigt sich, dass viele Studierende Cookies eher als Vorteil gesehen haben und ihnen erst jetzt die Risiken bewusst werden. Diese entstehen besonders durch die Möglichkeit, viele Daten eines Internet-Nutzers zusammenzuführen und damit ein Surfprofil zu erzeugen, wie dies z.B. vom Online-Werber Doubleclick angestrebt wurde. Doubleclick optimiert die Internetwerbeangebote vieler Anbieter und hat dadurch die Möglichkeit, umfangreiche Datensammlungen anzulegen, um daraus Nutzerprofile zu erstellen.

Datenbanken

Wegen der großen Verbreitung relationaler Datenbanken sollte die Behandlung dieses Themas im Informatikunterricht nicht fehlen. Nach einer Modellierung in Entity-Relationship-Diagrammen wird ein relationales Datenbanksystem zur Simulation eingesetzt. Unter Linux stehen viele verschiedene relationale Datenbanksysteme zur Verfügung. Für den Unterricht hat sich die am weitesten verbreitete freie Datenbank mysql* bewährt, die im Netz von allen Clients aus genutzt werden kann. Nach der Erstellung einer Datenbasis können die

Studierenden komplexe Abfragen in der Abfragesprache SQL durchführen. Als Alternative kann eine Online-Datenbank mit Lamp* (Linux, Apache, Mysql und PHP) genutzt werden, die ein Video-Center simuliert. Zusätzlich werden sämtliche Arbeitsblätter zu der Fallstudie zur Verfügung gestellt [<http://bics.be.schule.de/infs/datenbanken/online.html>].

Entwicklung von Betriebssystemen

Als vertiefendes Thema aus dem Bereich Betriebssysteme eignet sich die Problematik von Multi-Tasking-Systemen. Eine Einführung in das Prozesskonzept von Unix zeigt, wie auf einem PC viele Prozesse parallel laufen können. Nach dem Zeitscheibenverfahren wird nacheinander jedem Prozess kurz CPU-Zeit zugewiesen, dann wird er suspendiert und der nächste Prozess wird aktiv. In Bruchteilen von Sekunden erfolgt der Wechsel zwischen den einzelnen Prozessen, so dass der Benutzer eines Programms gar nicht merkt, dass noch andere Programme gleichzeitig laufen. KSysGuard*, ein KDE-Programm zur Systemüberwachung, stellt alle laufenden Prozesse dar. Mittels der tree-Darstellung kann leicht nachvollzogen werden, wie die Prozesse der Reihe nach gestartet werden und welcher Prozess zur Zeit ausgeführt wird.

Da ein Server im Netz viele Anfragen parallel bedient, stellt sich die Frage nach der Konsistenz der Daten. Es werden Szenarien entwickelt, wie Doppelbuchungen entstehen können. Zur Demonstration werden zwei Programme parallel gestartet, die auf die gleiche Datei zugreifen und sie verändern. Da bei jedem Testlauf andere Fehler auftreten, erkennen die Studierenden, wie komplex eine Fehlersuche in solchen Situationen ist. Als Lösung werden Verfahren zur Vermeidung solcher Fehler durch Synchronisation paralleler Prozesse besprochen.

Bei den meisten Unterrichtsthemen erweist sich Linux als wesentlich einfacher und methodisch geeigneter für den Unterrichtseinsatz als MS-Windows. Die ausgewählte Software hat den Vorteil, dass Studierende sie auch zu Hause installieren und nutzen können, ohne Lizenzprobleme. Regelmäßige Aktualisierung durch Software-Updates kann ohne finanzielle Probleme durchgeführt werden.

Dieser Bericht gibt nur einen kleinen Einblick in freie Software und soll ermutigen, selber freie Software im Unterricht zu erproben.

Kasten 1

Euregio-Kolleg

Das Euregio-Kolleg in Würselen ist ein Weiterbildungskolleg, d.h. Erwachsene mit mittlerem Schulabschluss und Berufserfahrung können hier nach 3 Jahren Abitur machen.

Ergänzend zu den Präsenzstudiengängen

⑩ Vollzeit-Kolleg mit 30 Unterrichtsstunden pro Woche

⑩ Teilzeit-Kolleg mit 20 Unterrichtsstunden pro Woche

wird im Sommer ein neuer Studiengang abitur-online erprobt, der aus 50% Präsenz- und 50% Distanzphasen besteht. Die Studierenden nehmen an 2 Abenden jeweils 5 Stunden am Unterricht im Kolleg teil, die weiteren 10 Unterrichtsstunden absolvieren sie nach eigener Zeitplanung am heimischen PC.

Im Rahmen der e-initiative.nrw werden am Euregio-Kolleg Fortbildungsveranstaltungen für Lehrerinnen und Lehrer durchgeführt, zur Nutzung des Internet und neuer Medien im Fachunterricht. Damit Lehrerinnen und Lehrer bei kleineren Fehlern nicht den Unterricht im PC-Raum abbrechen müssen, lernen die Teilnehmerinnen und Teilnehmer in einem ergänzenden Kurs „Erste Hilfe im PC-Raum“, wie kleine Hardware-, Netz- und Software-Probleme selbst behoben werden können. Ein weiterer Kurs wendet sich an die pädagogischen Schulnetzbetreuerinnen und -betreuer, die ein Schulnetz unter Linux verwalten.

Kasten 2

Kurs für pädagogische Schulnetzbetreuerinnen und -betreuer

Zum Einsatz neuer Medien im Unterricht werden Lehrerinnen und Lehrer benötigt, die basierend auf dem Medienkonzept der Schule vernetzte Lernumgebungen konzipieren, bestehende Netze ständig weiterentwickeln, administrieren und Kolleginnen und Kollegen bei dem Einsatz der neuen Medien im Unterricht Hilfestellung geben. Im Kurs für pädagogische Schulnetzbetreuerinnen und -betreuer mit Schwerpunkt Linux lernen die Teilnehmerinnen

und Teilnehmer welche Netzdienste sie basierend auf dem CT/ODS-Server oder einem SuSE-Server an ihrer Schule anbieten können, wie die Systeme administriert und die Arbeit möglichst effizient gestaltet werden kann.

Folgende Inhalte werden an 4 Tagen behandelt:

- ⑩ Anforderung an ein Schulnetz - Einführung in die Nutzung
- ⑩ Netzwerkgrundlagen
- ⑩ Installation und Konfiguration eines Linux-Servers (c't/ODS-Server 3.x, SuSE 7.x)
- ⑩ Server im Intranet
- ⑩ WWW-Server um Dokumente von Pädagogen, Schülerinnen und Schülern im Intranet zur Verfügung zu stellen, geschlossene Bereiche für Arbeitsgruppen
- ⑩ Zugang zum Internet mittels
- ⑩ Proxy-Server Squid (u.a. Sperren von WWW-Seiten, Freigabe des Zugangs für einzelne PCs oder Klassenräume)
- ⑩ Router
- ⑩ Mail im Intranet und Internet
- ⑩ FTP-Server mit persönlichem und anonymem Zugang
- ⑩ Vernetzung der Windows-Arbeitsplatzrechner (Windows 95/98/NT/2000) und Anbindung an den Server
- ⑩ Die wichtigsten Unix-Grundlagen
- ⑩ Benutzer- und Gruppenverwaltung, Disk-Quotas
- ⑩ Samba als NT-Server
- ⑩ Arbeiten im Netz (Ordner und Dateien)
- ⑩ Software und CDs im Netz
- ⑩ Verzeichnis- und Rechtekonzepte für Schulnetzwerke
- ⑩ Domänenanmeldung und Logon-Skripte.
- ⑩ Konzepte zur Datensicherung
- ⑩ Wiederherstellung von Rechnern
- ⑩ Reparatur des Linux-Servers

Ergänzend werden folgende Themen behandelt:

- ⑩ NFS
- ⑩ Linux-Workstation mit KDE, StarOffice (kompatibel zu MS-Office), Gimp zur Bildbearbeitung, Netscape ...

⑩ Linux-PC als Terminal-Server

Kasten 3

Unterrichtssequenz

1. Semester

- ⑩ Grundlegende Informationen zu den Komponenten eines PCs
- ⑩ Einführung in die Handhabung eines PCs
- ⑩ Nutzung von Intranet und Internet
- ⑩ Kommunikation per Email (Sicherheits- und Datenschutzaspekte)
- ⑩ Einführung in das [WWW](#) und den Netscape-Navigator

2. Semester

- ⑩ Informationsrecherche im Internet, Bewertung der Qualität der beschafften Informationen
- ⑩ Einführung in das Publizieren im Netz mit dem Netscape-Composer
- ⑩ Organisation der Dateien auf der Festplatte
- ⑩ Einführung in das algorithmische Denken

3. Semester

- ⑩ Geschichte der Informatik
- ⑩ Einführung in Aufbau und Funktionsweise eines PCs
- ⑩ Einführung in die Nutzung des Betriebssystems Linux
- ⑩ Einführung in die objektorientierte Programmierung mit Java-Applets

4. Semester

- ⑩ Einführung in die Algorithmik
- ⑩ Kryptografie und Sicherheit in Netzen
- ⑩ Objektorientierte Modellierung mit UML

5. Semester

- ⑩ Netzgrundlagen
- ⑩ Sicherheit in Netzen und Datenschutz
- ⑩ Client-Server-Anwendungen
- ⑩ Datenstrukturen (am Beispiel von Graphen)
- ⑩ Algorithmen (z.B. Verfahren zur Bestimmung des kürzesten Weges in Graphen)
- ⑩ NP-Probleme

6. Semester

- ⑩ Modellierung komplexer Probleme
- ⑩ Datenbanken oder
- ⑩ Synchronisation paralleler Prozesse