EuroBSDCon 2004 Karlsruhe, Germany:

An Introduction to Sysadmin Training in the Virtual Unix Lab

Hubert Feyrer <hubert@feyrer.de>

October 31st, 2004

Abstract

The Virtual Unix Lab (vulab) is an interactive course system which allows students to do Unix system administration exercises. Machines are installed on which students can do their assignments with full ”root”-access. At the end, the system checks which parts were done correctly, and gives a feedback on the exercise result. Access to the lab is via the Internet via a web-browser as well as standard Unix clients (ssh, telnet, ftp). Some detail on exercise-verification are outlined in this paper.

Contents

1 Introduction & Background **2**

2 The Virtual Unix Lab **2**

3 A Tour trough the Virtual Unix Lab **2**
3.1 User Area 2
3.2 Admin Area 11
3.3 Creating New Exercises 18

4 Setup **28**
4.1 Hardware 28
4.2 Lab Machine Installation 30
4.3 Restricting Access to Lab Machines 31
4.4 Software 31

5 Current Status **32**

6 Future Perspectives **32**

References **32**

1 Introduction & Background

A problem in teaching Unix system administration is the lack of machines available on which students can practice with full system administrator privileges. Without system administrator (root) privileges, many things cannot be practiced. On the other handside, when handing out root privileges, the lab machines are in an unknown state, requiring reinstallation of the lab machines for future students to get a known safe & well-configured environment.

The Virtual Unix Lab was created to solve this problem.

2 The Virtual Unix Lab

The Virtual Unix Laboratory was started in the “Praktikum Unix Cluster Setup” project in the “Hochschul- und Wissenschaftsprogramm (HWP)” initiative at the University of Applied Sciences of Regensburg (FH Regensburg). It was designed as an interactive course system for system administration training in general, and with a focus on training installation and configuration of the Network File System (NFS) and Network Information System (NIS) on Unix-based systems in particular.

The Virtual Unix Lab fulfills this purpose today. After sign-up, machines are installed on demand, and students can do their assignments with full ”root”-access. Users can book exercises for a certain time, and all machines are setup identically. Exclusive access to the lab machines during exercises is guaranteed, with access to the lab being realized via the Internet via a web-browser as well as standard Unix clients (ssh, telnet, ftp). At the end of the assignment, the system checks if/which parts were done correctly, and gives a report containing feedback on the success of the exercise to the student.

After that, machines are re-installed from scratch for next user and exercises.

3 A Tour trough the Virtual Unix Lab

The tour through the Virtual Unix Lab covers both the parts that the users will see while using the system as well as a few areas which cover administrative actions plus an overview of the steps for updating and creating a new exercise.

3.1 User Area

This tour through the user area of the Virtual Unix Lab covers the following areas:

- Login and account creation
- List of exercises
- Booking an exercise
- Taking an exercise
- Retrieving feedback afterwards

The tour itself will consist of a number of screenshots displaying the various parts of the web based user interface that the Virtual Unix Lab presents.

1. Access to the user interface of the Virtual Unix Lab is through a web browser, which allows accessing all facilities provided, except performing exercises themselves (see

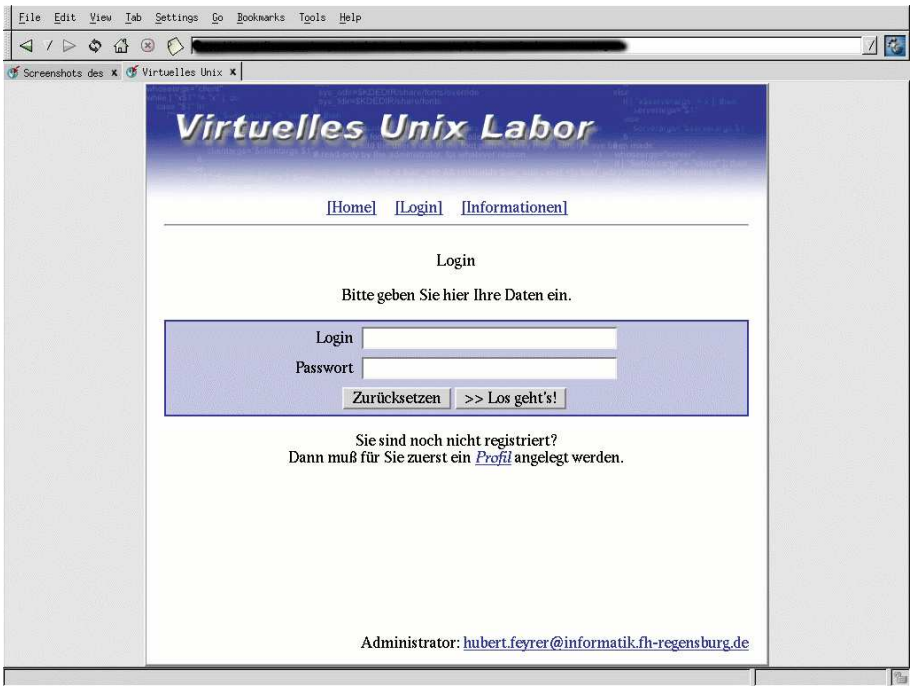


Figure 1: Logging into the Virtual Unix Lab

below). Language of the user interface is German (only) right now – internationalisation is on the list of items to do in the future.

When accessing the webpage, the first thing students encounter is a mask to login as displayed in figure 1.

2. If a student doesn't have a login yet, he can create a new login ("Profile") using the form displayed in figure 2.

   The student will have to give his student ID number ("Matrikel-Nummer"), first and last name, an email address where he can be reached and a password (twice). Upon registering, an email will be sent to the given email address, which contains an authentication token that the user has to enter to permanently enable his account. Accounts not enabled that way will be deleted after 7 days. This allows instant access to the lab, but ensures that people provide at least a valid email address if they want to keep using the lab.

3. After successful login into the Virtual Unix Lab, the welcome screen shown in figure 3 is displayed and users can choose from several actions they want to do: Update their user settings ("Benutzerdaten"), get a list of available exercises ("Übungen auflisten"), book an exercise for a certain time & date ("Buchung vornehmen"), get a list of past and future exercises, delete future exercises and retrieve feedback on past ones ("Buchungen einsehen") as well as logout of the web site:

4. Figure 4 shows a list of exercises available in the Virtual Unix Lab, including the exercise name ("Übung"), a one-line description of the exercise ("Bezeichnung") and duration of the exercise for the user ("Dauer"):

5. Each of the exercises in the list can be clicked on to retrieve the exercise text as shown when actually taking the exercises, see figure 5 for an example.

   This allows preparing the exercises, and learning all the items necessary to successfully perform the exercise in the Virtual Unix Lab.

6. After these preparations, an exercise can be booked by selecting the "übung buchen"

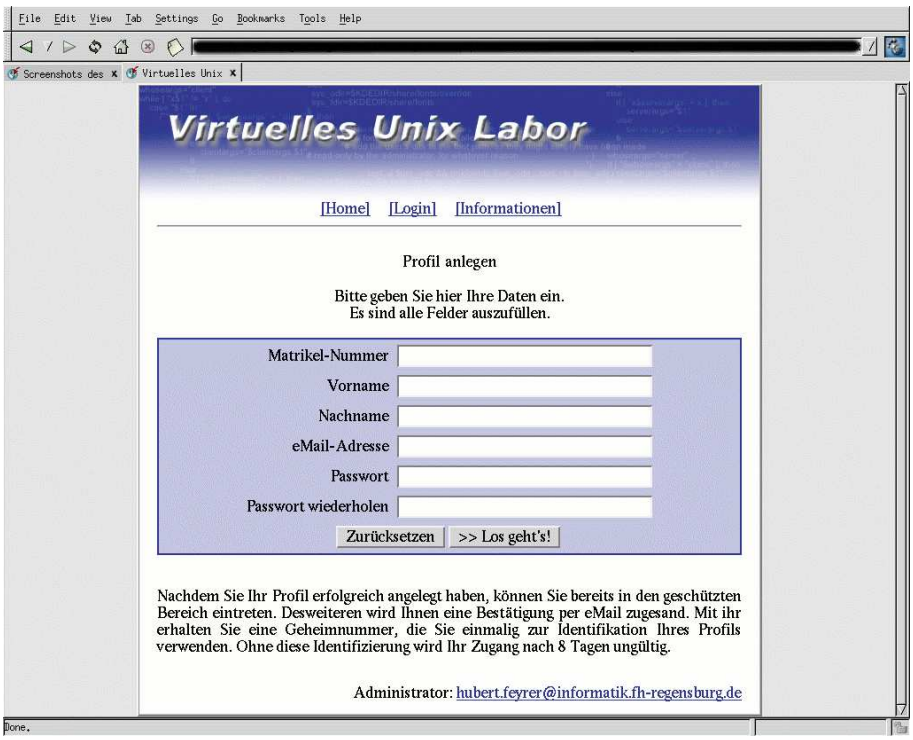


Figure 2: Entering data for a new login

menu item. The first step in booking an exercise consists of deciding at which date and time to take the exercise, which is displayed in figure 6.

Exercises are available in three-hour intervals (1.5 hours for the exercise, plus about one hour for preparation of the lab machines and some time for postprocessing), slots already booked by other users are not displayed as available. In the above screenshot, the exercises at 0am, 3am, 6am, 9am and 12am are not available because of that.

7. After deciding on the date and time for the exercise to take place, the next step is to choose which actual exercise to take, from the list of available exercises. As in the previous list of available exercises, the exercise name, description and duration are displayed, and the user has to decide for one as displayed in figure 7.
8. After selecting date & time and which course to take, a final confirmation shown in figure 8 has to be made before the exercise is booked.
9. The exercise is booked now, and the system will know when to prepare the lab machines for the exercises, using an at(1) job.

   The student can walk away, prepare for the exercise or whatever, and as for a real test, he should come back to the lab a few minutes before the selected time of the exercises, logging in again, see figure 9.
10. After the user has logged in, the system will tell him that an exercise was prepared for him (and which one), and that he can already start preparing the exercise by following the provided link ("bitte hier klicken" in the red text) as displayed in figure 10.
11. Before starting the exercise, the student has to enter the IP address of the machine from which he wants to access the lab machines. This process, shown in figure 11, is needed to restrict access so other students cannot disturb the exercise.

    What happens when the user has entered his IP number here is that appropriate rules will be injected into the firewall covering the lab machines to allow access to the lab machines only from the given hostname when the exercise starts.

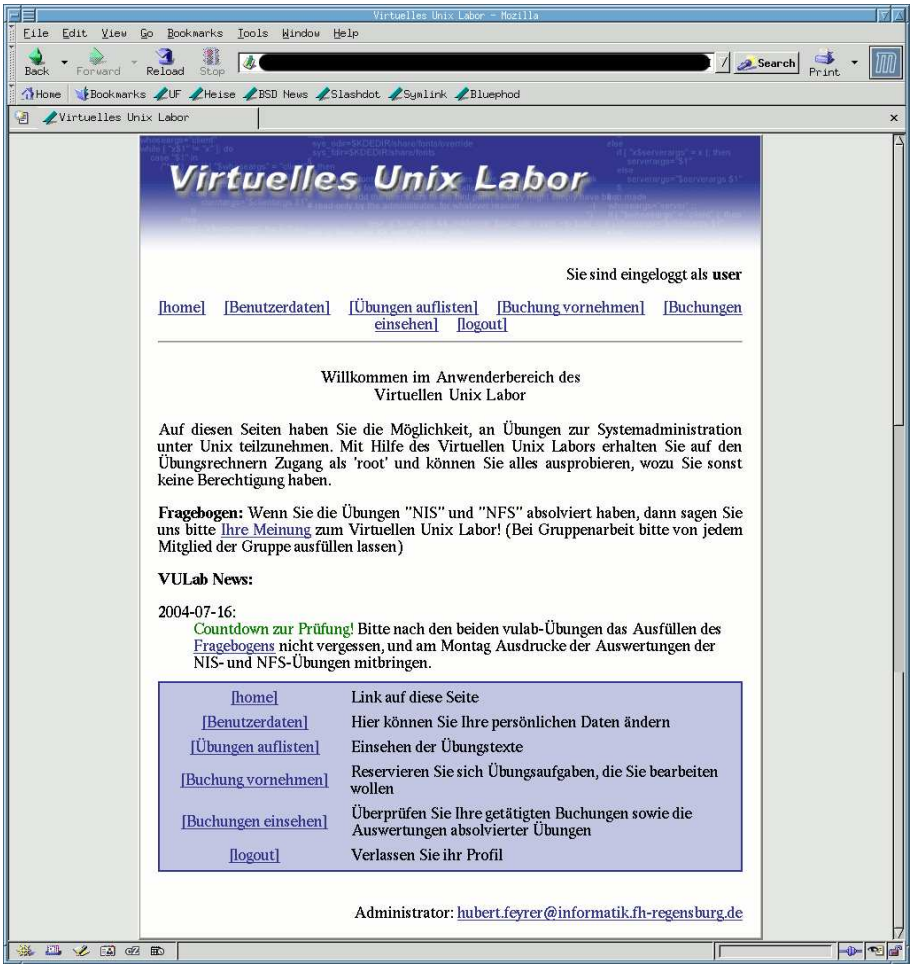


Figure 3: Welcome to the Virtual Unix Lab

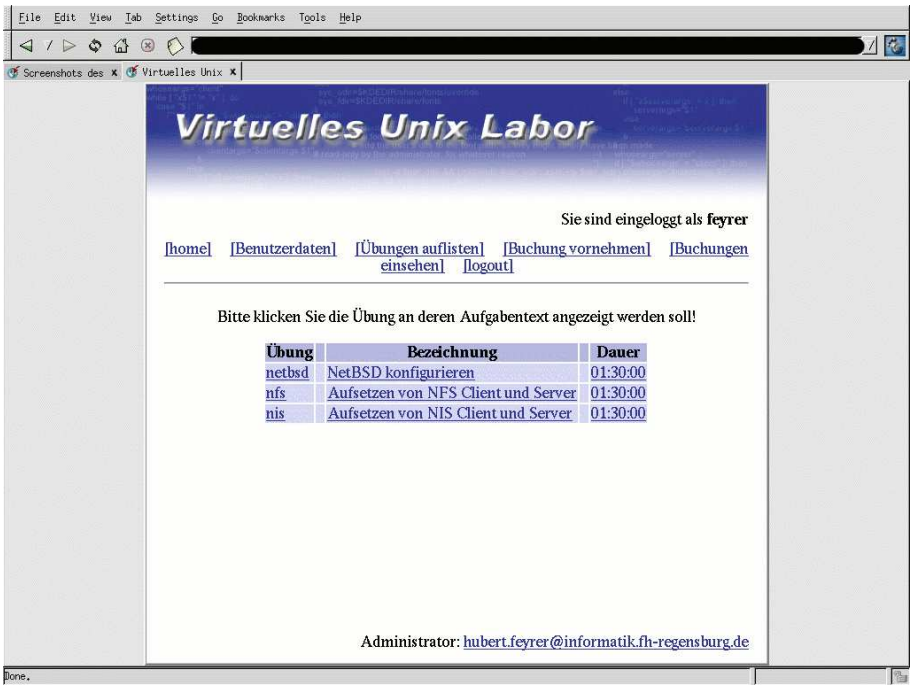


Figure 4: List of available exercises

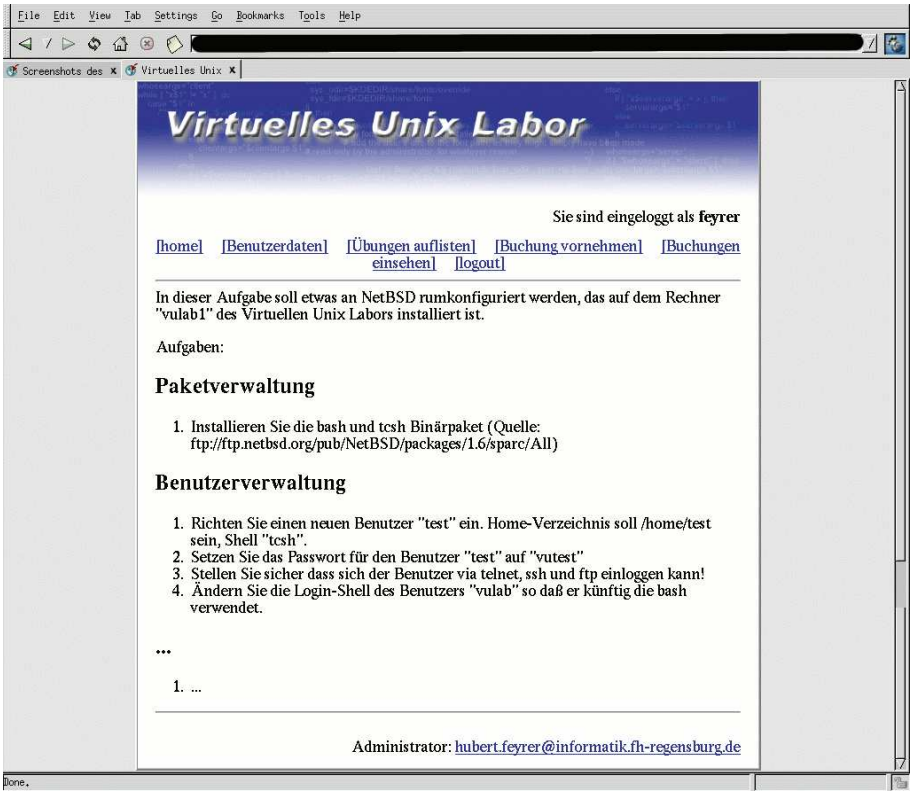


Figure 5: Exercise text preview

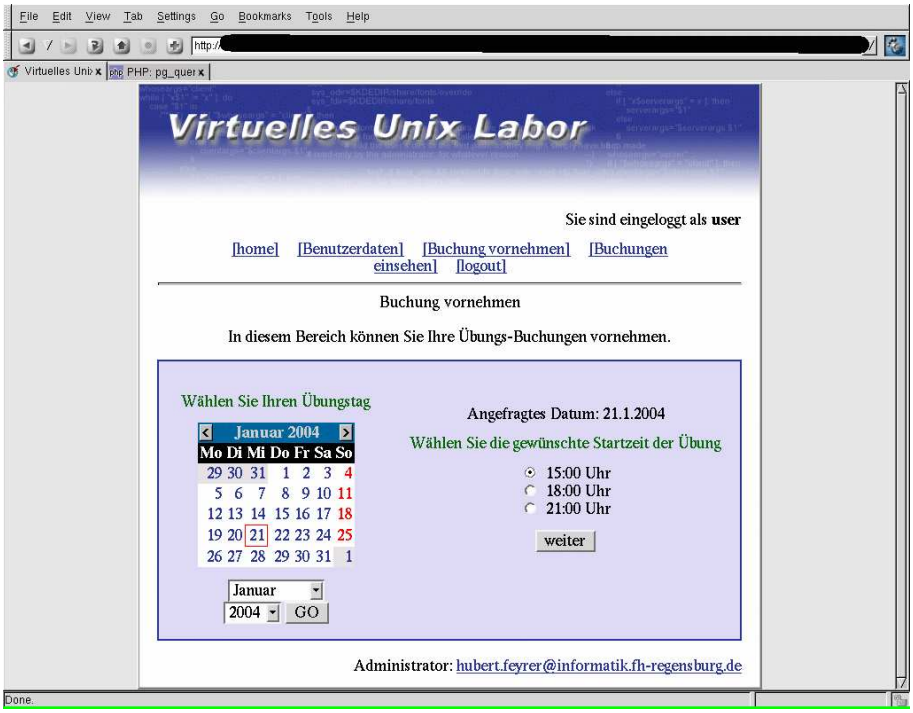


Figure 6: Booking an exercise: selecting date & time

12. Just as in a real test, the student can come into the lab and sit down, but the test won't start until a certain time. In a real lab test, this would be when the teacher passes out sheets of paper with the exercises printed on them. In the Virtual Unix Lab, the student has to wait for the start of the exercise too, as displayed in figure 12.

13. When exercise time is reached, the firewall protecting the lab systems will be opened to allow access to the lab systems, and the exercise will be displayed as e.g. in figure 13.

    The text displayed here is the same as was available for looking at before, so students were able to prepare properly, with a few small additions. First, a link with help for accessing the lab systems is placed under the exercise text, so students not familiar with the lab (yet) can make themselves familiar how to access the lab machines, giving proper syntax for telnet, ftp and ssh. Below this link, the time remaining for the exercise is printed on the lower left ("Verbleibende Zeit"), and if the user decides she has finished the exercise before the time runs out, this can be stated by pressing the "Fertig!"-button.

14. For accessing the lab machines, separate terminal windows have to be opened to access the lab machines and perform the tasks needed to successfully solve the tasks given in the exercise text as shown in figure 14.

    Figure 14 shows access to a Solaris/sparc (left xterm) and NetBSD/sparc (right xterm) system. For each system, the user knows a "normal" user account and password (without any system privileges) as well as the system administrator (root) password, given in the instructions on how to access the lab machines.

    The student can the use any measures he seems appropriate to solve the tasks, using the full administrative privileges he has available. If one of the lab machines has to be rebooted, this can be done as with any normal remotely administrated machine (i.e.: there is no access to the console, right now).

15. After the exercise has ended – either because time ran out, or because the student

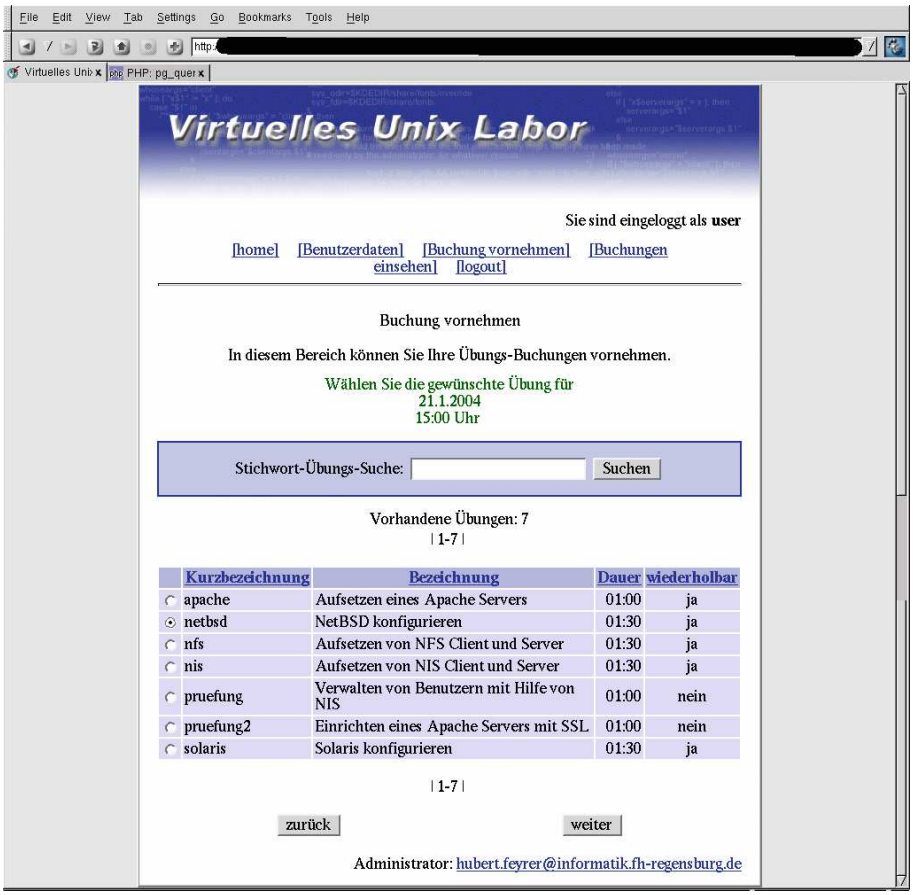


Figure 7: Booking an exercise: selecting the exercise

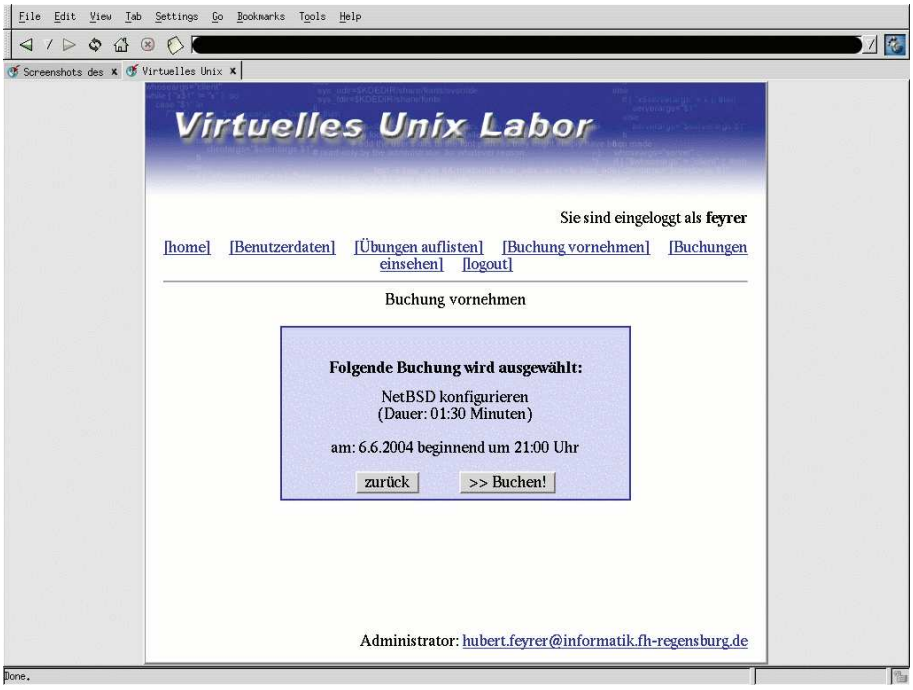


Figure 8: Booking an exercise: confi rmation

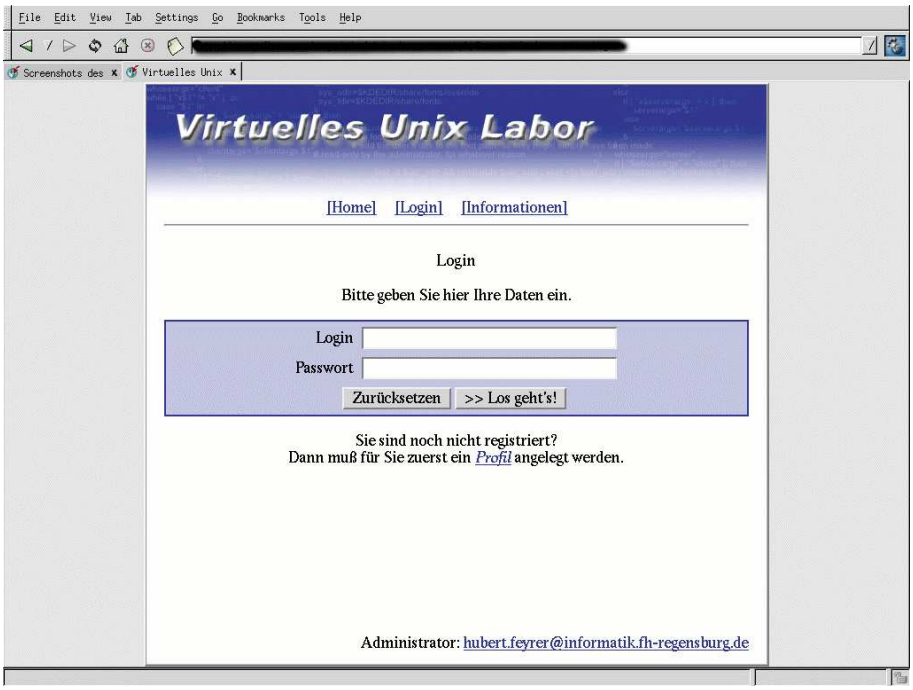


Figure 9: Logging in for a booked exercise

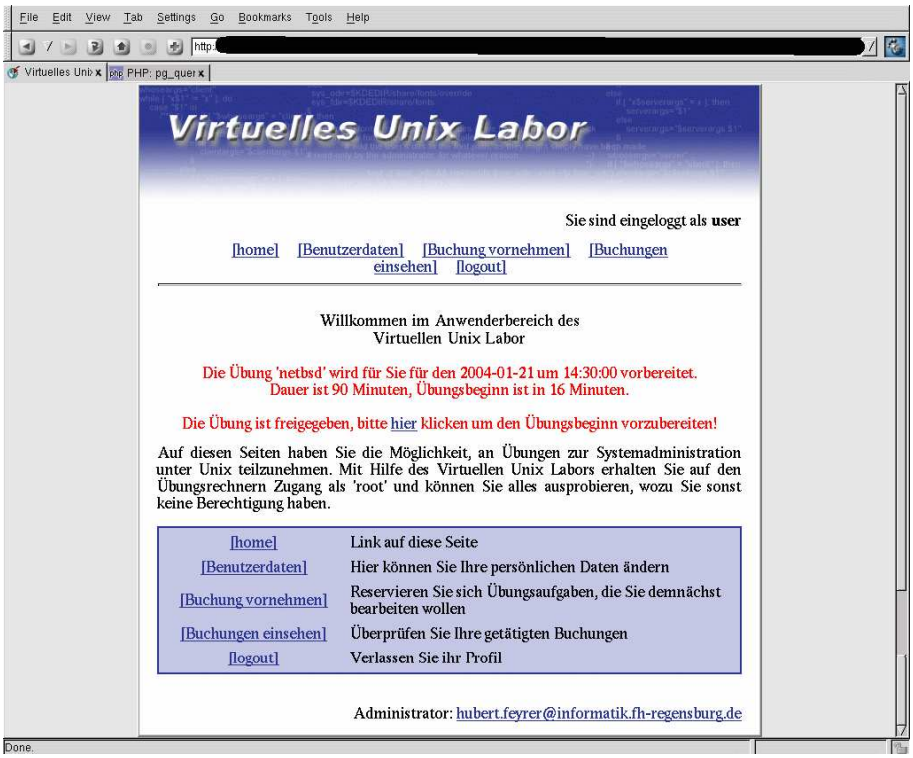


Figure 10: An exercise is prepared & waiting

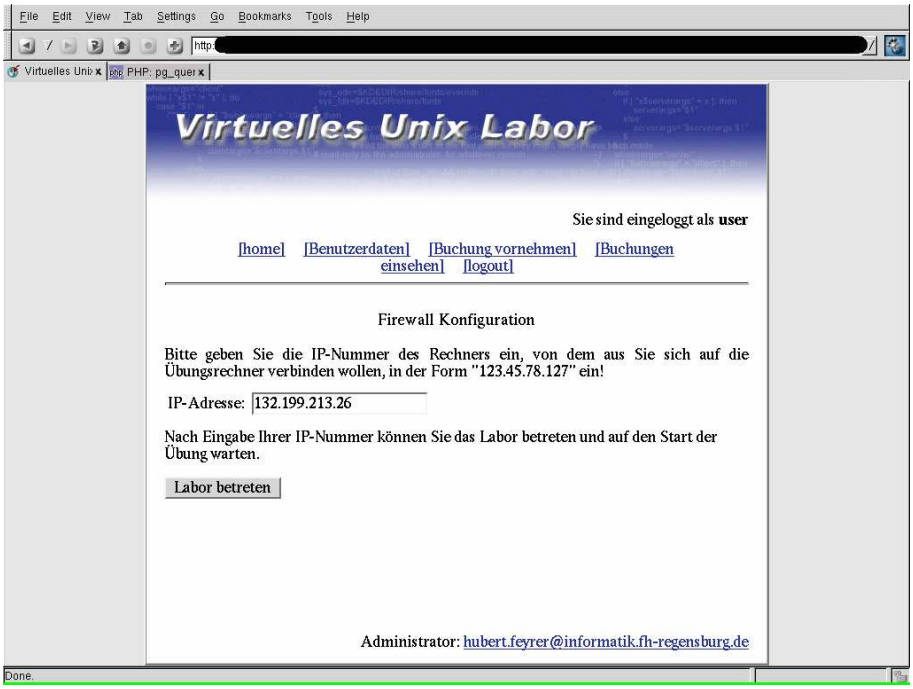


Figure 11: Confi guring access to the lab machines

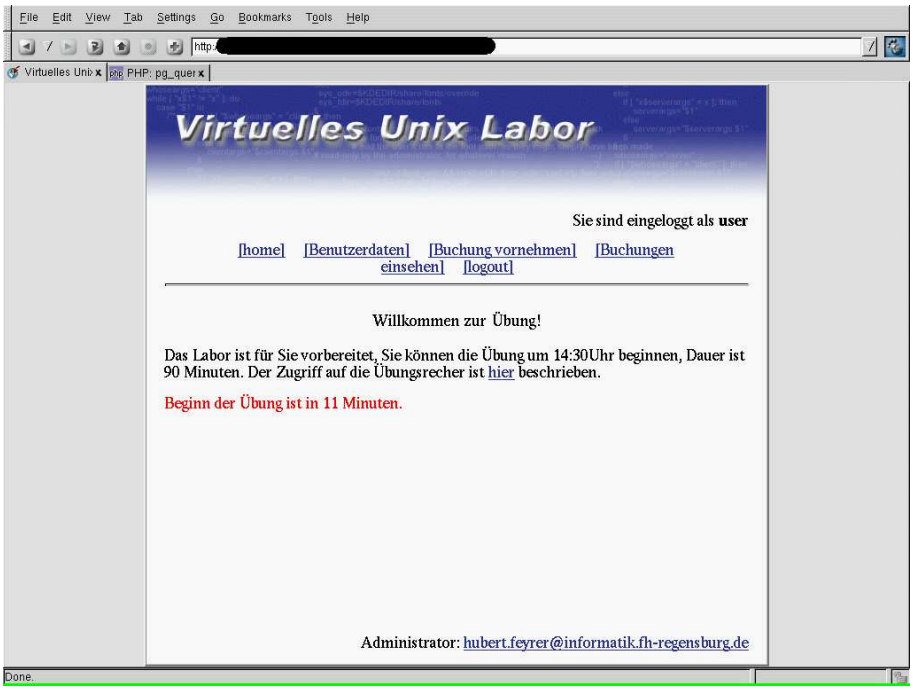


Figure 12: Waiting for start of exercise time

signalled he's done by pushing the "Fertig!"-button – the system will revoke access to the lab systems by re-enabling the firewall, and print a message that the exercise is over and that feedback on the exercise can be retrieved from the database within a few minutes as shown in figure 15.

At this point, the lab systems are analyzed in the background by a number of scripts, which know what configuration steps are necessary for successful performance of the exercise, and which will report their findings in the database for later retrieval (see below for more on this).

16. After an exercise has finished, students can retrieve feedback on an individual exercise via the main menu ("Buchungen einsehen"). They will see the exercise text again, with the various tasks containing comments on what checks were done (green text) , and if the particular check was done successfully ("OK") or not ("Nein"). See figure 16 for an example.

3.2 Admin Area

After a walkthrough of the functions provided by the Virtual Unix Lab to it's users and students, this section concentrates on some of the administrative actions that are available to administrators of the Virtual Unix Lab.

Again, here are the various impressions to show these aspects, in screenshot format.

1. Users with administrative privileges in the Virtual Unix Lab have to log into the system like "normal" users too, the system will know they have admin status, and display available actions as appropriate. Available actions are updating & changing user settings ("Benutzerdaten"), editing or creating new exercises ("Uebungs-Setup"), looking at past and future exercises booked by all users ("Buchungen") and seeing what users said in the survey ("Feedback", not covered here). Figure 17 shows the admin

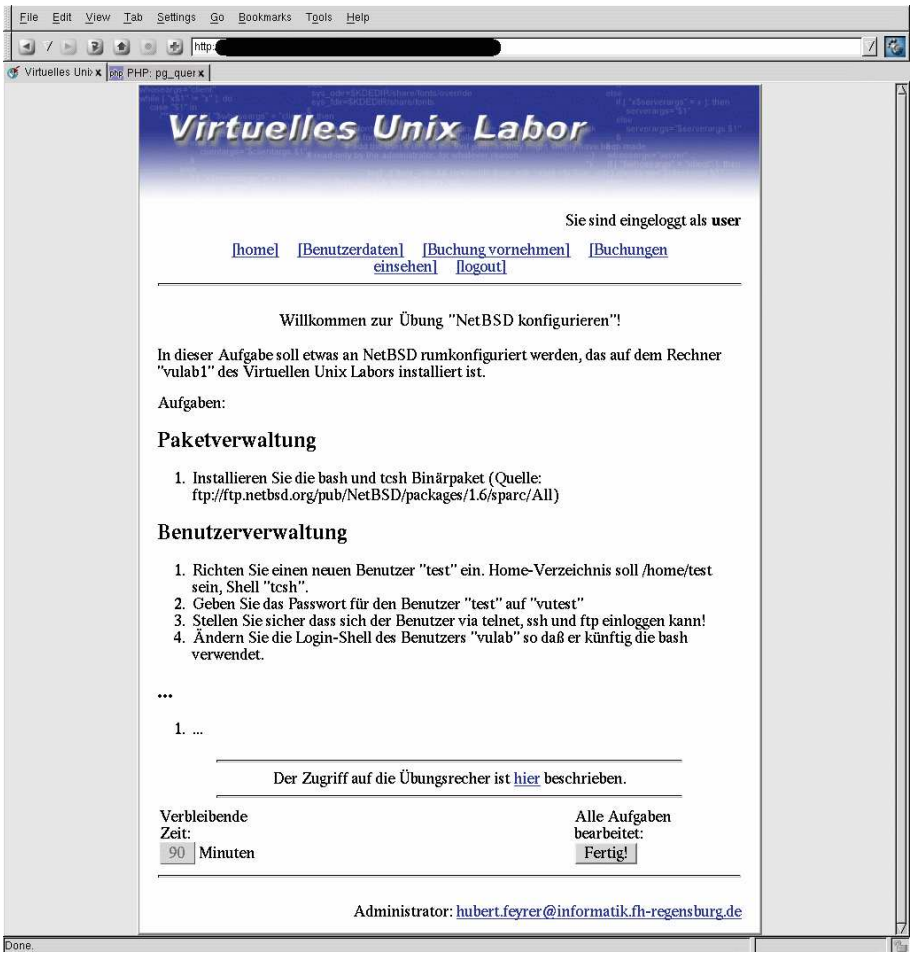


Figure 13: Display of the exercise text

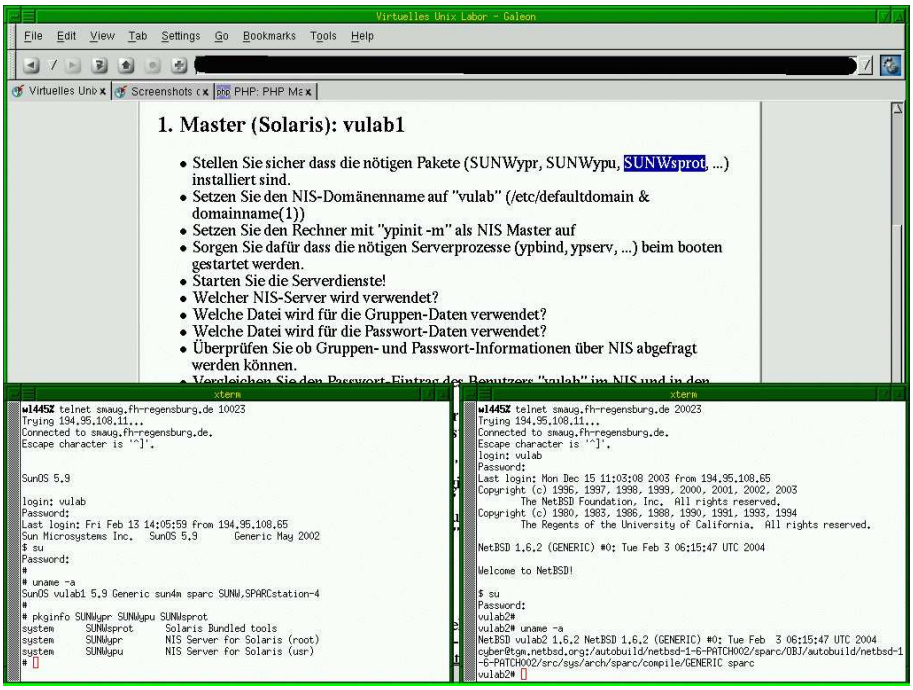


Figure 14: Logging into lab machines for the exercise

Figure 15: End of exercise

File Edit View Tab Settings Go Bookmarks Tools Help

http:[redacted]

Virtuelles Unix PHP: pg_quer

Virtuelles Unix Labor

Sie sind eingeloggt als **user**

[home] [Benutzerdaten] [Buchung vornehmen] [Buchungen einsehen] [logout]

Auswertung der Übung "NetBSD konfigurieren"!

Die Übung "NetBSD konfigurieren" (Buchungs-ID #77) fand am 2004-01-21 von 12:00:00 bis 13:23:37 statt und dauerte damit 83 von max. 90 Minuten. Die Übung wurde von der IP-Nummer 132.199.213.26 aus absolviert.

Es folgt die genaue Auswertung der einzelnen Teilaufgaben:

In dieser Aufgabe soll etwas an NetBSD rumkonfiguriert werden, das auf dem Rechner "vulab1" des Virtuellen Unix Labors installiert ist.

Aufgaben:

Paketverwaltung

1. Installieren Sie die bash und tcsh Binärpaket (Quelle: ftp://ftp.netbsd.org/pub/NetBSD/packages/1.6/sparc/All)

   Pakete installiert? (pkg_info -e) Nein

Benutzerverwaltung

1. Richten Sie einen neuen Benutzer "test" ein. Home-Verzeichnis soll /home/test sein, Shell "tcsh".

   "test" finger(1)bar? OK
   Korrektes Home-Verzeichnis? (finger, test -d) OK
   Shell richtig gesetzt? (finger) OK
   Eintrag in /etc/master.passwd? OK

2. Geben Sie das Passwort für den Benutzer "test" auf "vutest"

   Passwort richtig gesetzt? (getpwnam(3), crypt(3)) OK

3. Stellen Sie sicher dass sich der Benutzer via telnet, ssh und ftp einloggen kann!
4. Ändern Sie die Login-Shell des Benutzers "vulab" so daß er künftig die bash verwendet.

   Login-Shell vulab? (chfn/chsh, finger) Nein

...

1. ...

Anzahl Teilübungen: 7
Davon bestanden: 5 (72%)

[home]	Link auf diese Seite
[Benutzerdaten]	Hier können Sie Ihre persönlichen Daten ändern
[Buchung vornehmen]	Reservieren Sie sich Übungsaufgaben, die Sie demnächst bearbeiten wollen
[Buchungen einsehen]	Überprüfen Sie Ihre getätigten Buchungen
[logout]	Verlassen Sie ihr Profil

Administrator: hubert.feyrer@informatik.fh-regensburg.de

Done.

Figure 16: Feedback on an exercise taken

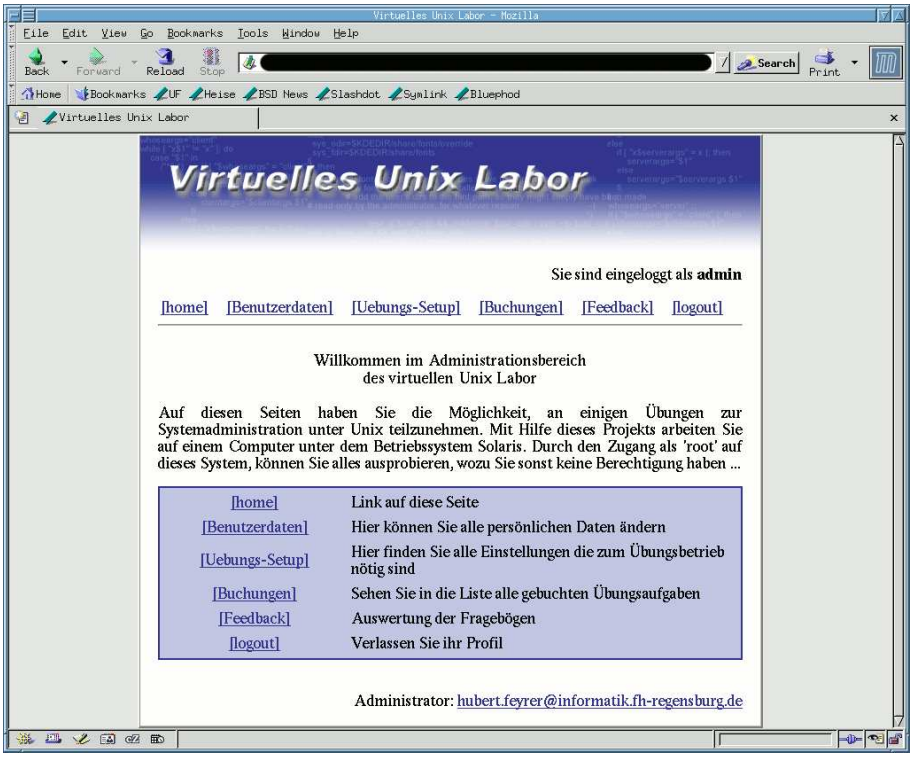


Figure 17: Administrator's login & menu

login screen.

All the menu items available in this menu will be presented in the following paragraphs.

2. Selecting "Benutzerdaten" displays a list of all users of the Virtual Unix Lab as shown in figure 18, including their student ID ("MatrikelNr"), last and first name, login (which defaults to the email address entered when signing up, but can be changed by users later) as well as the type of a user. Types are defined for users and administrators, with an additional type reserved for teachers of the system ("dozent") for future additions.

   Two buttons are available right to each user, the lower button to delete the user from the system (and all his associated data, like exercises he took), and the upper one to edit the data of a certain user.

3. When pressing the button to edit a user's data, the form displayed in figure 19 will be presented to change a users' data.

   Interesting items here are the user type ("Benutzer-Typ"), which is usually "User" for students, and "Admin" for administrators of the Virtual Unix Lab. If a new user's login has not been confirmed yet by entering the authentication token mailed to the user when registering, the "Anmeldung bestätigt?" field will be set to "Nein" to indicate the login hasn't been confirmed, with the authentication token being displayed for confirmation & verification.

   If a user can't remember his password, a new one can be filled in (twice, for confirmation) to set a new one. During tests with a group of 40 computer science students over the duration of one semester in summer 2004, it was amazing how many students managed to forget their password – "I have forgotten my password" apparently isn't a user-support myth (only)!

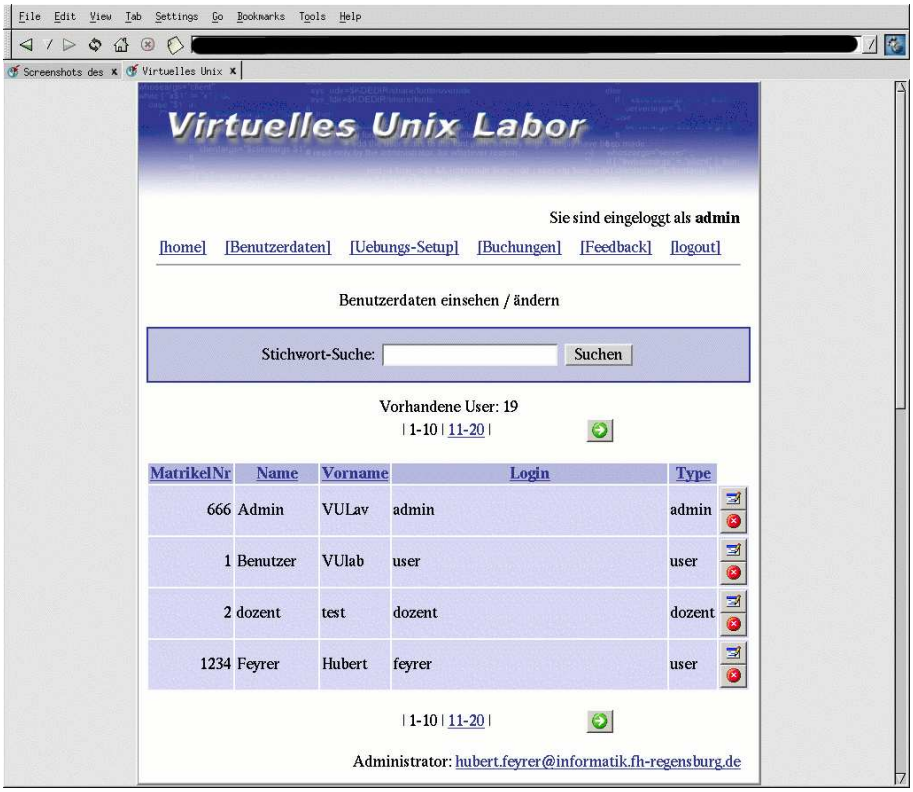


Figure 18: Managing users

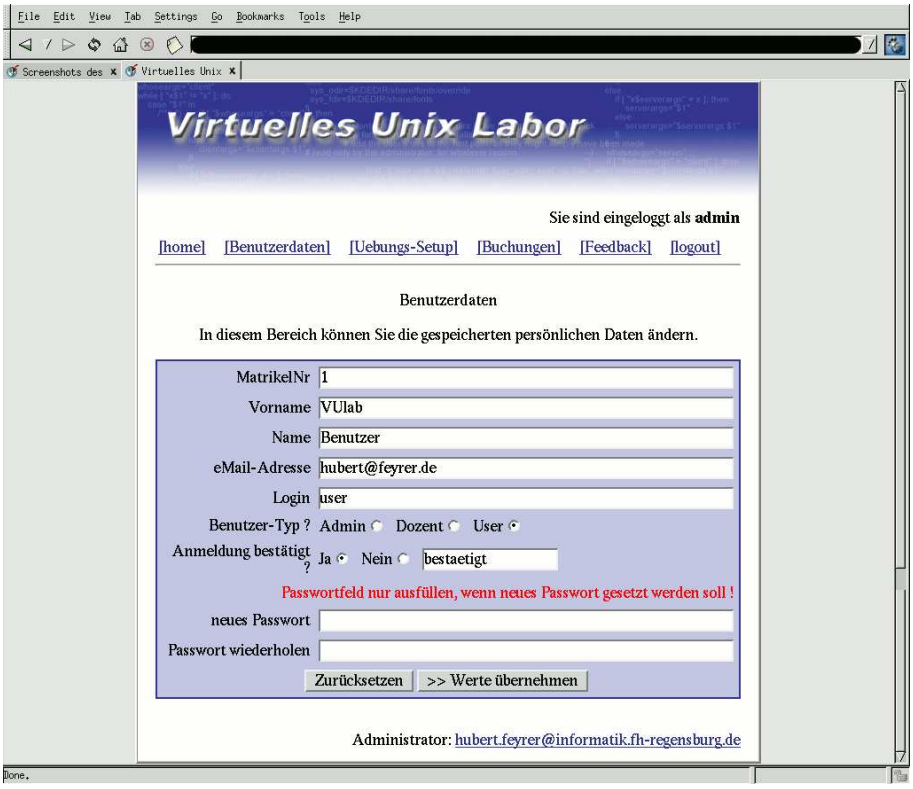


Figure 19: Editing user specifi c data

4. Listing all exercises booked by all users in the past and future can be achieved by selecting “Buchungen” from the menu. The list, displayed in figure 20, contains login name of the user and what exercise he booked, including date & time and also an indicator if the exercise is still to be taken (“freigegeben”) or already done.

   To the right of each exercises, two buttons are placed, just as in the for for editing user data. The lower button can be used to delete all traces of a past exercise (not available to normal users to prevent them from destroying evidence), the upper button can be used to retrieve feedback on a particular exercise for both admin and student users.

5. When an administrator selects feedback on a certain exercise, he will get similar feedback as normal users (see figure 21), containing data if single tasks of the exercise were solved successfully, and some hints what the system did to test (in green font).

   In addition to normal users, administrators will get an overview on how all students taking the same exercise performed, giving numbers on how many did (“Bestanden”) or did not (“Nicht bestanden”) manage to solve the task successfully, as well as an overall number of students who did the exercise. In addition to absolute numbers and percentage, bars of “o”s are printed to give sort of a graphical overview, making it more visible how the overall group performed.

6. When selecting the “Uebung-Setup” menu item, various settings for new and existing exercises displayed in figure 22 can be changed. First, a new menu row will appear which allows managing the list of lab machines available (“Rechner verwalten”), the list of harddisk images available for them (“Images verwalten”) as well as creating a new exercise (“neue Übung erstellen)”. Furthermore, a list of all existing exercises is given, with two buttons on the right. The rightmost button can be used to delete an exercise (including all data available on that exercise, esp. past exercises taken by students - use with care!). The other button can be used to change various settings of the exercise, using the same menus as when creating a new exercise, see below.

3.3 Creating New Exercises

1. When creating new exercises or editing existing ones, a list of lab machines needs to be known. This list can be edited by selecting “Rechner verwalten” in the previous menu, and using the user interface shown in figure 23.

   Buttons besides the entries are used to delete entries for lab machines, or edit properties, with hostnames being the only property right now.

2. A similar list can be retrieved for all harddisk images available, that can be deployed on the lab machines.

   As can be seen in figure 24, harddisk images are available for NetBSD 1.6.2/sparc and Solaris 9/sparc right now. Adding other harddisk images for other operating systems like Linux, or special setups like troubleshooting would be easily possible.

   At this point, only one kind of lab machine is available (two Sun SPARCstation 4), and thus no additional checks are needed if an image can be deployed on a certain machine. This may change in the future!

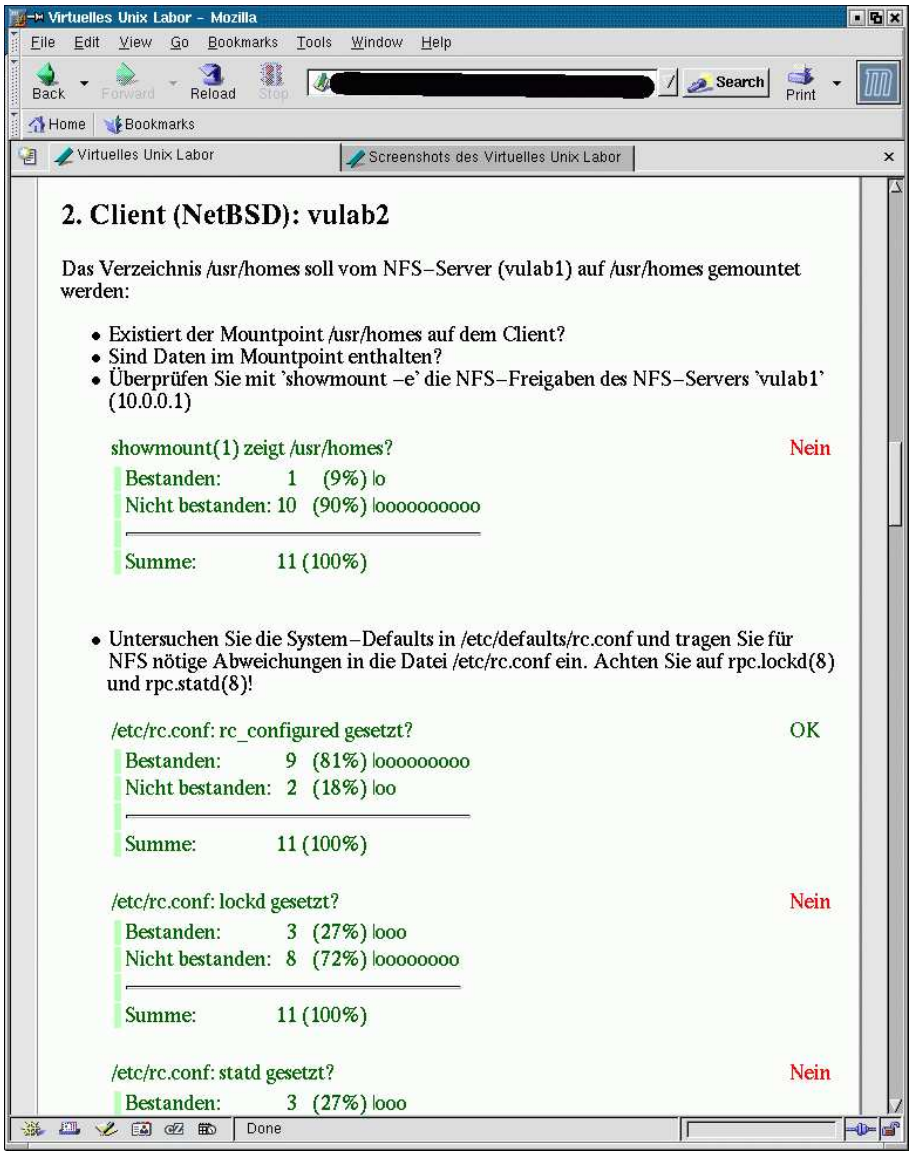
3. When choosing to create a new exercises, three screens have to be filled with data.

   The first one asking for general information on the new exercise, like a short description (“Kurzbezeichnung”), one-line description (“Bezeichnung”) and a username that should be allowed exclusive access to the exercises (“Nur für”, used for administrative exercises, see below). Time for preparing the lab machines (“Vorlauf”), duration of the exercise (“Dauer”) as well as time for analyzing the lab machines (“Nachlauf”) is needed next, each given in hours and minutes.

   If an exercise should not be repeatable like for a real test, not just a lab exercise, this can be done by setting the exercise to be not repeatable (“Wiederholbar?”). Next item needed is file placed in the filesystem of the Virtual Unix Lab which contains the exercise text (“Pfad auf die Textdatei”) and which needs to contain some special PHP calls to allow giving feedback (not covered here).

ID	Login	Kurzbez.	Bezeichnung	Datum	Startzeit	Dauer	freigegeben	at_id
114	feyrer	nfs	Aufsetzen von NFS Client und Server	03.05.2004	15:00 Uhr	01:30	nicht-mehr	234
115	feyrer	nis	Aufsetzen von NIS Client und Server	11.05.2004	21:00 Uhr	01:30	nicht-mehr	236
123	walter.kern@stud.fh-regensburg.de	nfs	Aufsetzen von NFS Client und Server	21.05.2004	15:00 Uhr	01:30	nicht-mehr	248
124	walter.kern@stud.fh-regensburg.de	nis	Aufsetzen von NIS Client und Server	21.05.2004	21:00 Uhr	01:30	nicht-mehr	249
122	trm35740	nfs	Aufsetzen von NFS Client und Server	22.05.2004	12:00 Uhr	01:30	nicht-mehr	246
120	walter.kern@stud.fh-regensburg.de	nfs	Aufsetzen von NFS Client und Server	22.05.2004	15:00 Uhr	01:30	nicht-mehr	244
121	walter.kern@stud.fh-regensburg.de	nis	Aufsetzen von NIS Client und Server	23.05.2004	12:00 Uhr	01:30	nicht-mehr	245
126	walter.kern@stud.fh-regensburg.de	nis	Aufsetzen von NIS Client und Server	23.05.2004	15:00 Uhr	01:30	nicht-mehr	256

Figure 20: Managing booked exercises, past & future



2. Client (NetBSD): vulab2

Das Verzeichnis /usr/homes soll vom NFS–Server (vulab1) auf /usr/homes gemountet werden:

- Existiert der Mountpoint /usr/homes auf dem Client?
- Sind Daten im Mountpoint enthalten?
- Überprüfen Sie mit 'showmount –e' die NFS–Freigaben des NFS–Servers 'vulab1' (10.0.0.1)

showmount(1) zeigt /usr/homes? Nein

Bestanden: 1 (9%) |o
Nicht bestanden: 10 (90%) |oooooooooo

Summe: 11 (100%)

- Untersuchen Sie die System–Defaults in /etc/defaults/rc.conf und tragen Sie für NFS nötige Abweichungen in die Datei /etc/rc.conf ein. Achten Sie auf rpc.lockd(8) und rpc.statd(8)!

/etc/rc.conf: rc_configured gesetzt? OK

Bestanden: 9 (81%) |ooooooooo
Nicht bestanden: 2 (18%) |oo

Summe: 11 (100%)

/etc/rc.conf: lockd gesetzt? Nein

Bestanden: 3 (27%) |ooo
Nicht bestanden: 8 (72%) |oooooooo

Summe: 11 (100%)

/etc/rc.conf: statd gesetzt? Nein

Bestanden: 3 (27%) |ooo

Figure 21: Retrieving feedback on a group's performance

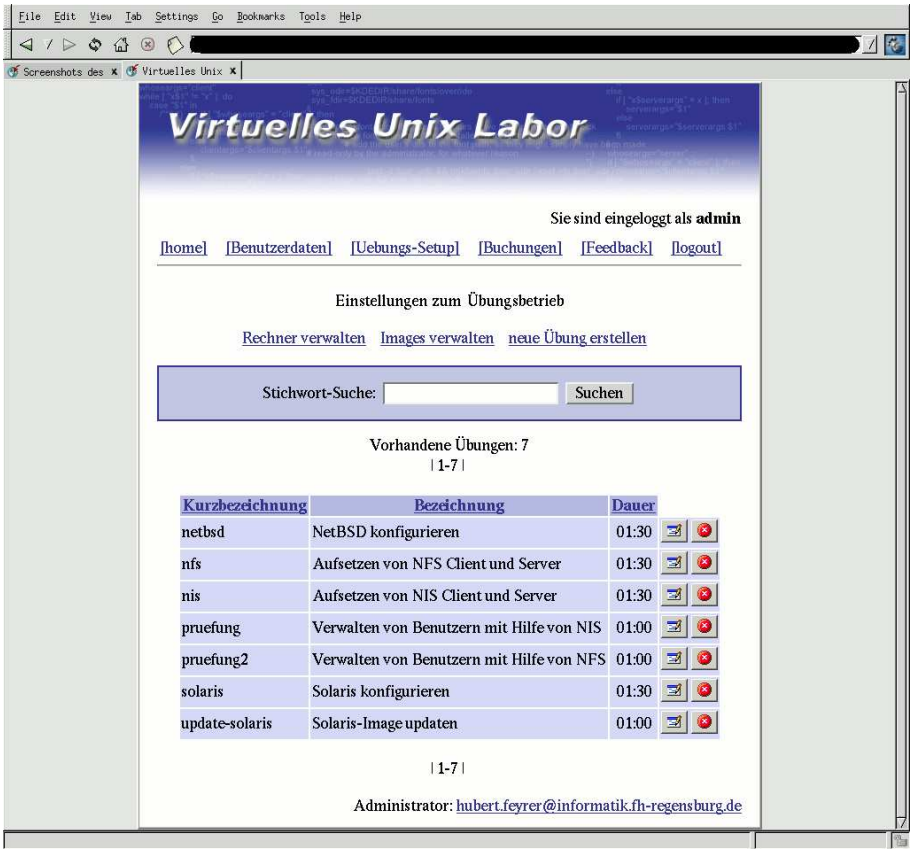


Figure 22: Editing various properties of an exercise

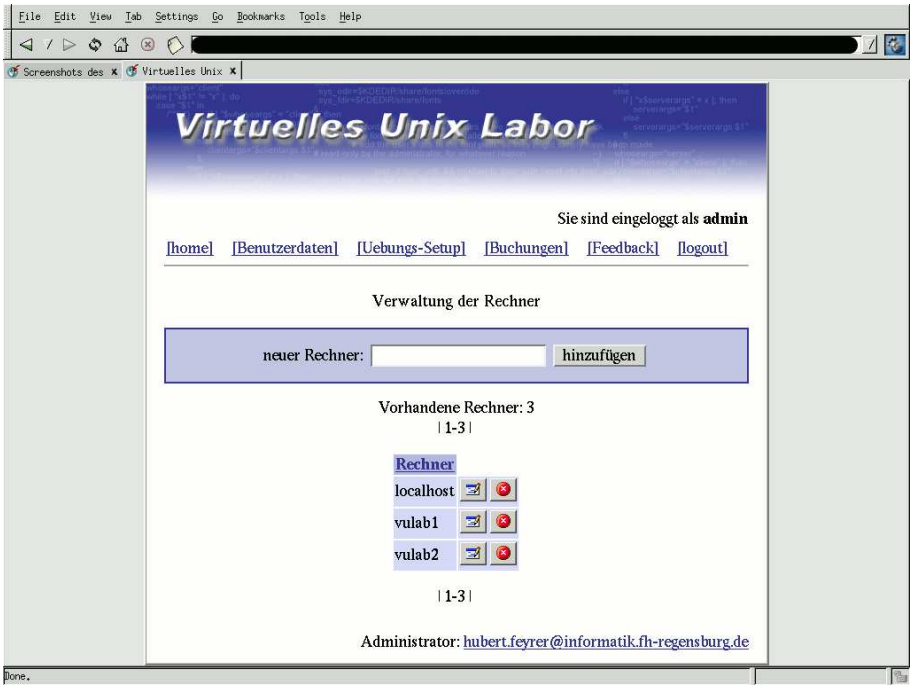


Figure 23: Managing lab machines

Figure 24: Managing disk images for lab machines

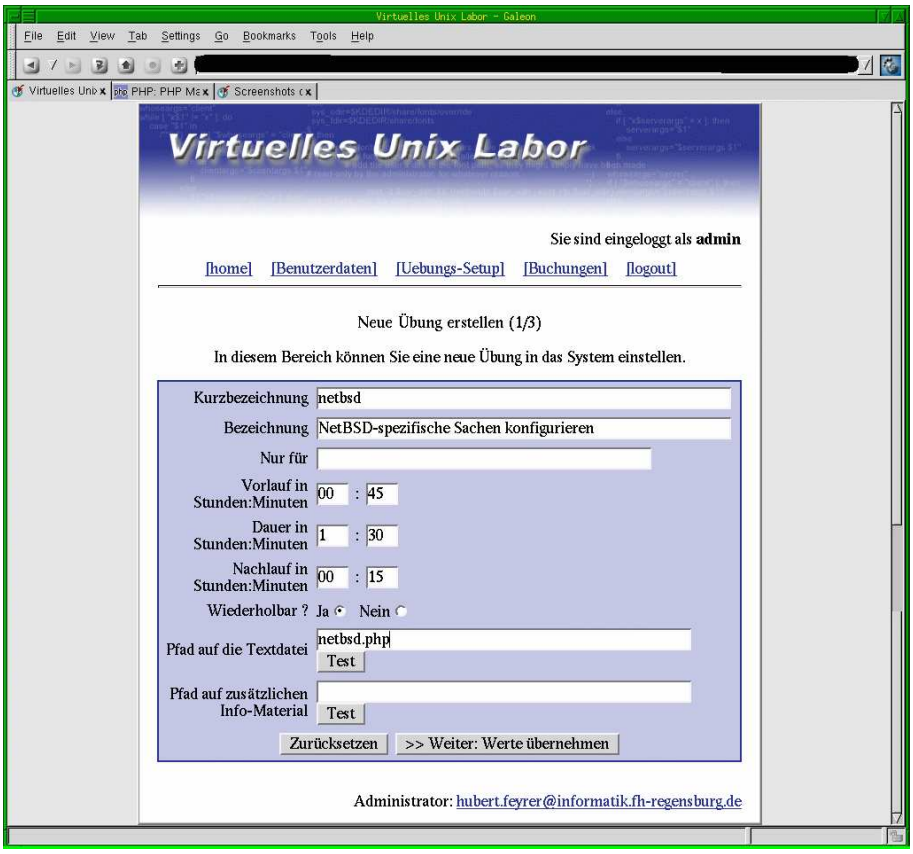


Figure 25: Entering basic data for a new exercise

Finally, design plans for the Virtual Unix Lab include a tutorial component which can give help on demand, which is what the filename with additional information ("Pfad auf zusätzliches Info-Material") is intended for, but not used right now.

Figure 25 displays the first screen asking for all this data.

4. The second part of creating a new exercise consists of deciding which lab machines are needed for the exercise ("verwendeter Rechner"), and which harddisk image they should get installed ("benötigtes Image"). Pressing the "Rechner-Konfiguration hinzufügen" button will add the machine/image-combination to the exercise. A list of machines & images already part of the image are printed below that, with buttons available to delete the machine/image combination or to edit it. To make sure several lab machines get setup for an exercise, they must be added with an appropriate exercise here.

   In the example screenshot displayed in figure 26, only one lab machine running NetBSD 1.6.2 would be prepared for the exercise (unless others were added).

5. The third and last step of creating a new exercise consists of defining which tests are made at the end of the exercise to determine if the (parts of) the exercise were performed successfully or not.

   Testing can be done via a number of so-called "Check-Scripts", which exist to test a number of aspects of a system. Parameters can be passed to the scripts to adjust what they do (see below). As an exercise can involve several lab machines, and as each lab machine can run a different configuration (client, server, ...), it's important to define on which machine a check script runs ("Läuft auf Rechner").

   The selected check script will be ran on the named machine with the given parameters at the end of of an exercise, and store the result of the script – success or failure –

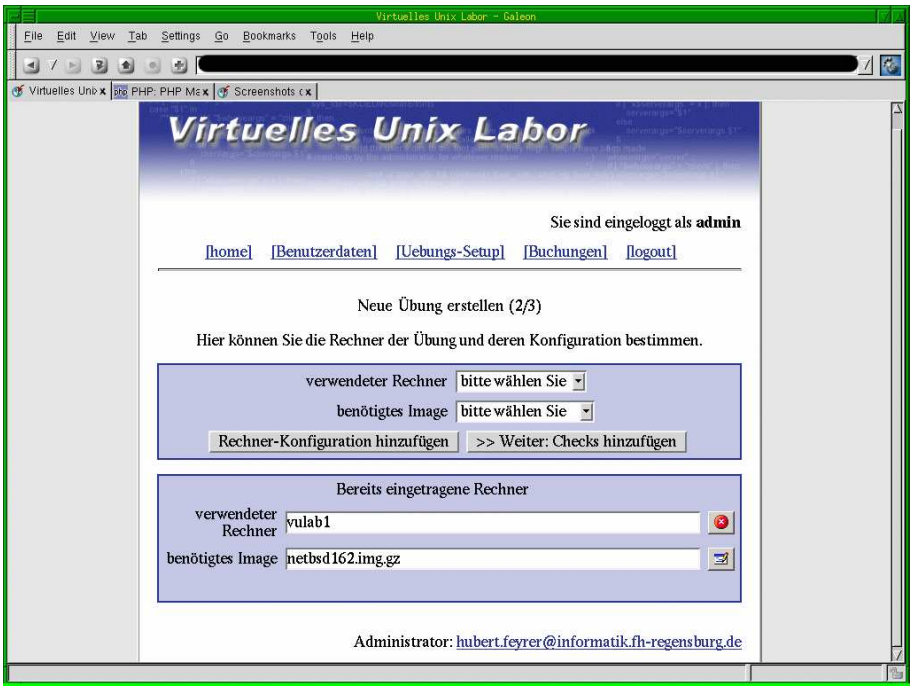


Figure 26: Defi ning machines & confi guration for a new exercise

in the database for giving feedback later. As a simple display of "success" or "you failed" may not be too helpful[1], an additional description of what exactly was tested and was either passed or failed can be given ("Bezeichnung für Auswertung").

As with the selection of lab machines used for an exercise, several checks can be added to an exercise ("Check hinzufügen"), completion of the new exercise can be indicated by pressing the "Fertig" button of the form displayed in figure 27.

6. All scripts available to perform checks on the lab machines are stored on the filesystem of the Virtual Unix Lab server. A list which allows easy selection of a check-script is made available through the PHP framework displayed in figure 28.

   Scripts are available to test various aspects of a system, either being independent of the operating system ("`check-*`"), work on all Unix systems ("`unix-check-*`") or only on a particular Unix flavour ("`netbsd-check-*`", "`solaris-check-*`"). If a certain subsystem allows various properties to be changed, that's also encoded in the check script's filename ("`unix-check-user-*`", "`check-file-*`"). Future incarnations of the Virtual Unix Lab may also include check scripts to run on Windows systems.

7. Figure 29 shows an exercise that has some check scripts defined to be ran upon completion of the exercise:.

   For each check, buttons are available to remove it from the list of checks to run, and to edit the data stored for the particular check.

8. When choosing to edit a particular check, all the parameters from the previous form can be changed using the form displayed in figure 30: script, parameters, description for feedback, and on which machine to run the script.

   In addition, information on what the script does is printed in addition to a list of parameters that the script takes, including name of the variable, the default value (if applicable), and a description of the parameters.

[1] [Schulmeister, 1997] p. 111

Figure 27: Adding checks for a new exercise

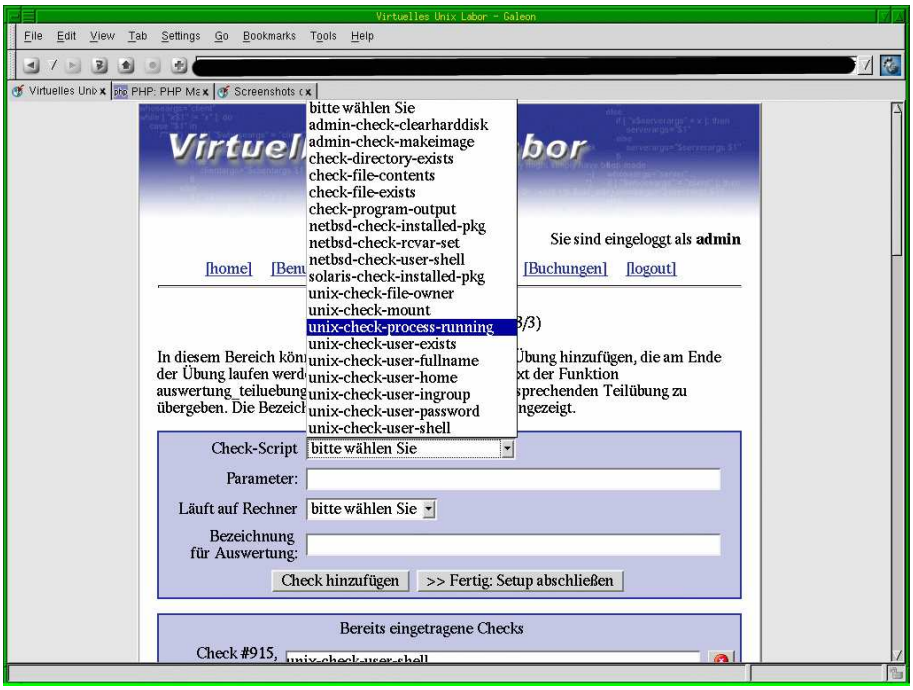


Figure 28: A list of available check-scripts

File Edit View Tab Settings Go Bookmarks Tools Help

Virtuelles Unix

Virtuelles Unix Labor

Sie sind eingeloggt als **admin**

[home] [Benutzerdaten] [Uebungs-Setup] [Buchungen] [Feedback] [logout]

Übung editieren (3/3)

In diesem Bereich können Sie neue Check-Scripte zur Übung hinzufügen, die am Ende der Übung laufen werden. Die Check-# ist im Übungstext der Funktion auswertung_teiluebungen() für die Auswertung der entsprechenden Teilübung zu übergeben. Die Bezeichnung wird bei der Auswertung angezeigt.

Check-Script: bitte wählen Sie

Parameter:

Läuft auf Rechner: bitte wählen Sie

Bezeichnung für Auswertung:

Check hinzufügen | >> Fertig: Setup abschließen

Bereits eingetragene Checks

Check #917, Script: admin-check-makeimage

Parameter: RECHNER=vulab1 IMGFILE=solaris29-2.img.gz

Läuft auf Rechner: localhost

Bezeichnung für Auswertung: Image von vulab1 in solaris29-2.img.gz ziehen

Check #916, Script: admin-check-clearharddisk

Parameter:

Läuft auf Rechner: vulab1

Bezeichnung für Auswertung: Festplatte von vulab1 von Muell saeubern (dd </dev/null >x)

Administrator: hubert.feyrer@informatik.fh-regensburg.de

Figure 29: An exercise with two checks defined

Figure 30: Editing data of a certain check

The information on purpose of a check-script and list of parameters including their description is retrieved from the check scripts – all check scripts are expected to allow querying them for these informations.

9. A typical exercise consists of about 30-40 single checks to run at the end. Entering all the data for all these checks via the web frontend is possible, but tiresome. To solve this problem, an alternative way to enter data on check scripts into the database had to be constructed[2].

   As the text of an exercise already contains calls to PHP functions to retrieve success of the particular check (using the check numbers shown in the previous screenshots), putting data near these calls was an obvious solution. So, instead of putting

```
Do task #1

<?php auswertung_teiluebungen( 916, 917);
 ?>

Do task #2
...
```

   into the exercise text's PHP file to describe task #1 and associate checks numbered 916 and 917 with them as e.g. defined in figure 29, the layout can first be a bit changed (no functional change):

```
Do task #1

<?php auswertung_teiluebungen(
          916,
          917
      );
 ?>
```

   and after this layout change, comments can be made in the PHP code to add the data for the checks as comments as shown in figure 31.

   With an appropriate preprocessor, the data can be extracted and stored in the database. In the above example, the check-numbers are not known when writing the exercise text, and left as "XXX". When running the preprocessor, it will extract data from the PHP comments, and store them into the database.

   Figure 32 shows running the preprocessor, which fills in the now-known check-numbers into the PHP calls for giving feedback as shown in figure 33.

   Using this framework, it is possible to keep all the data for an exercise – text, which check script, on which lab machine to run it, any possible parameters as well as text for feedback – in one file, which is a lot easier to maintain. The web interface can still be used to edit existing exercises.

   More words could be spent here on exercise verification, stereotypes and language design, system front-ends, domain specific languages and design patterns, but it would be beyond the introductional character of this paper and be published elsewhere.

4 Setup

4.1 Hardware

The following machine is used as server machine:

[2] [Spinellis, 2001] p. 96

```
feyrer@wl445:/home/feyrer/work/vulab/code/public_html/texte
<p>

Aufgaben:
<p>

<h2> Paketverwaltung </h2>
<ol>
<li> Installieren Sie die bash und tcsh Bin\xe4rpaket (Quelle:
     ftp://ftp.netbsd.org/pub/NetBSD/packages/1.6/sparc/All)

     <?php auswertung_teiluebungen(
                 XXX, // vulab1: netbsd-check-installed-pkg PKG=tcsh
                      //         tcsh installiert? (pkg_info -e tcsh)

                 XXX  // vulab1: netbsd-check-installed-pkg PKG=bash
                      //         bash installiert? (pkg_info -e bash)
        ); ?>
</ol>


<h2> Benutzerverwaltung </h2>
<ol>
<li> Richten Sie einen neuen Benutzer "test" ein, Home-Verzeichnis
```

Figure 31: Embedding check-data into exercise text comments

```
feyrer@wl445:/home/feyrer/work/vulab/code/public_html/texte
wl445% perl uebung2db -v netbsd netbsd.php n
check_id 908 inserted (1)
check_id 909 inserted (1)
check_id 910 inserted (1)
check_id 911 inserted (1)
check_id 912 inserted (1)
check_id 913 inserted (1)
check_id 914 inserted (1)
check_id 915 inserted (1)
old checks removed from database
wl445%
```

Figure 32: Extracting check data into the database

```
feyrer@wl445:/home/feyrer/work/vulab/code/public_html/texte
--- netbsd.php  Mon Feb 23 16:39:21 2004
+++ n   Mon Feb 23 16:37:58 2004
@@ -1,3 +1,4 @@
+<!-- DB updated by feyrer on Mon Feb 23 16:37:57 MET 2004 from netbsd.php -->
 <!-- $Id: netbsd.php,v 1.13 2004/02/19 10:55:52 feyrer Exp $ -->
 <?php auswertung_ueberschrift(); ?>
 <!-- ---------------------------------------------------------------- -->
@@ -15,10 +16,10 @@
      ftp://ftp.netbsd.org/pub/NetBSD/packages/1.6/sparc/All)

      <?php auswertung_teiluebungen(
-                XXX, // vulab1: netbsd-check-installed-pkg PKG=tcsh
+                908, // vulab1: netbsd-check-installed-pkg PKG=tcsh
                     //         tcsh installiert? (pkg_info -e tcsh)

-                XXX  // vulab1: netbsd-check-installed-pkg PKG=bash
+                909  // vulab1: netbsd-check-installed-pkg PKG=bash
                     //         bash installiert? (pkg_info -e bash)
        ); ?>
 </ol>
@@ -30,23 +31,23 @@
      soll /home/test sein, Shell "tcsh".

byte 886
```

Figure 33: The preprocessor has filled in the check-script numbers

- Sun SPARCstation 5, 85MHz
- 192 MB RAM
- 3* external SCSI disk
- additional SBus ethernet card
- Runs NetBSD 1.6.2/sparc

The following machines are used as lab clients:

- Two
- Sun SPARCstation 4, 110MHz
- 64 MB RAM
- 1 GB internal SCSI disk
- Run NetBSD 1.6.2/sparc or Solaris 9/sparc

The ultimate goal here is to use virtual machines instead of real hardware. When the Virtual Unix Lab project was started, no hardware for running virtual machines was available.

4.2 Lab Machine Installation

Installation of the lab machine is done by using the server to act as DHCP, RARP and NFS server to lab-internal network. When a new exercise is to be prepared, the lab clients are netbooted, so they are independent of any operating system (and it's possible damaged state). From the netbooted environment, a new harddisk image is written to the lab client's harddisk. The image-deployment techniques used here were developed in the g4u[3] project.

[3] [Feyrer, 2004]

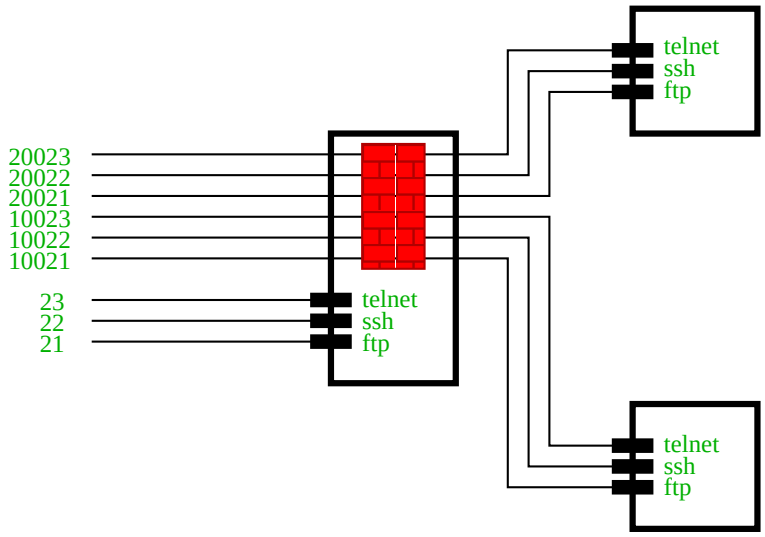


Figure 34: Accessing the lab clients

4.3 Restricting Access to Lab Machines

Protecting the lab clients from unauthorized access during exercises was one of the design goals of the Virtual Unix Lab, but it was also considered important to not hook up lab clients directly to the production network of the University of Applied Sciences of Regensburg (FH Regensburg) to prevent students having access to the lab machines abusing their admin privileges.

As a result, the lab clients were placed inside their own network, with only the lab machines and the Virtual Unix Lab server. The Virtual Unix Lab server acts as firewall, and access to the clients is realized by redirecting access to certain ports on the server to the client machines, as can be seen in figure 34.

The firewall is configured dynamically to allow access from a single client when an exercise starts (see figure 11), and access is disabled at the end of an exercise either when time runs out or the user indicates he's done, just before verification of the exercise results starts.

4.4 Software

The following software was used to create the Virtual Unix Lab:

Apache: Web server for the user interface and exercise text

Postgres: Database engine; MySQL didn't compile on NetBSD/sparc, and Postgres has worked very fine

IPfilter: Firewalling software by Darren Reed [Reed, 2004]

NetBSD: Operating system for the Virtual Unix Lab server and clients

Solaris: Fine operating system, for clients of the Virtual Unix Lab

PHP: Scripting engine for the web-based user interface

Perl: Scripting engine for result verification and some internals

Bourne shell: Scripting engine for result verification, client deployment and more internals

5 Current Status

Current status of the Virtual Unix Lab is that it works(!).

Two full-length exercises have been worked out and are available for students:

- Network Information System (NIS)
- Network File System (NFS)

The system was tested successfully in summer semester 2004 by 40 students during course "System Administration" at the department of computer science at the University of Applied Sciences (Fachhochschule) Regensburg.

An upgrade of the server hardware to a modern PC is pending.

6 Future Perspectives

Many ways are possible to improve the system on one end, but there are also a number of perspectives that the system in it's current incarnation can be used for. Here's an itemized list:

- **Funding** badly needed, for keeping the system running, and any of the following items
- Define **more exercises**:
  - Web- and Mail server, including spamfilters etc.
  - DNS, DHCP, LDAP, Samba
  - Database setup & tuning
  - System and network troubleshooting
  - Security post-morten analysis and prevention
- Add more options for **lab machines**:
  - Real hardware (PCs, Sun E15000)
  - Emulated (virtual) hardware: VMware, Xen, ...
  - More operating systems: Linux, Windows
- Internationalisation
- Implement a tutoring system
- Think about user modeling
- Do a lot of polishing and internal restructuring for the above items
- Funding! Very very badly needed!

References

[Feyrer, 2004] Feyrer, H. g4u - Harddisk Image Cloning for PCs [online]. (2004) [cited 2004-07-16]. Available from: `http://www.feyrer.de/g4u/`.

[Reed, 2004] Reed, D. IP Filter [online]. (2004) [cited 2004-09-14]. Available from: `http://coombs.anu.edu.au/~avalon/`.

[Schulmeister, 1997] Schulmeister, R. (1997). *Grundlagen hypermedialer Lernsysteme*. Oldenbourg Verlag, München, Germany.

[Spinellis, 2001] Spinellis, D. (2001). Notable design patterns for domain-specific languages. *The Journal of Systems and Software*, 56(1):91–99.