

Systemmanagement und Sicherheit

1. Übung

Aufgabe 1 (FVWM Umgebung)

Sie sehen

- einen Start-Button unten links, dieser ist nützlich
 - fürs Ausloggen
 - für das Experimentieren mit Ihrer Oberfläche
- eine Button-Leiste darüber, Sie haben Buttons für
 - ein Terminal
 - das Einloggen auf die Rechner isl-g-01, isl-s-04 und stl-s-stud
 - den Firefox-Browser
 - einen Editor
 - einen PDF-Viewer
 - einen Taschenrechner
- einen Pager, der Ihnen erlaubt, 6 virtuelle Bildschirme zu managen
- einen kill-a-window-Button, um nicht reagierende Anwendungen zu beenden
- auf der leeren blauen Fläche ein Kontextmenü mit
 - linker Maustaste (Starten von Anwendungen)
 - rechter Maustaste (Zugriff auf die Fenster)

Erzeugen Sie

- eine Proxykonfiguration für Firefox (Rechner:www-proxy, Port 3128)
- Proxykonfiguration für lynx
 - dies ist ein terminalbasierter Webbrowser

- `man lynx` zeigt die Manualpage
- Sie müssen die Umgebungsvariable `http_proxy` setzen
- Sie sollten den entsprechenden Befehl in der Datei `.bashrc` eintragen

Schließlich sollten Sie noch eine Datei `${HOME}/.bash_profile` mit dem Inhalt `source ${HOME}/.bashrc` anlegen, um sicherzustellen, dass Ihre Umgebungsvariablen nach jedem Einloggen gelesen werden.

Aufgabe 2 (Shell Kommandozeile)

Wir werden in der Kommandozeile mit der Bourne Again Shell arbeiten (`bash`). Falls Sie unter `stl-s-stud` die Kornshell eingetragen haben, haben Sie auf den ISL-Rechnern ebenfalls die Kornshell (`ksh`). In diesem Falle geben Sie nach dem Öffnen eines Terminalfensters einfach (`bash`) ein. Sie können das automatisieren, indem Sie das Kommando (`bash`) in

`${HOME}/.kshrc`

eintragen.

Die `bash` Shell hält einige Features parat, um das Editieren der Kommandozeile zu vereinfachen. Finden Sie diese in den folgenden Aufgaben heraus und ordnen Sie die richtigen Tastenkombinationen zu.

| | | | | | | | |
|----------------|----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| (1) [Alt]+f | (2) [Alt]+b | (3) [Strg]+l | (4) [Strg]+w | (5) [Strg]+k | (6) [Strg]+e | (7) [Strg]+a | (8) [Strg]+d |
|----------------|----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|

- Cursor an den Anfang der Zeile
- Cursor ans Ende der Zeile
- Löschen des Zeichens unter dem Cursor
- Löschen des Rests der Zeile
- Wortweises Weiterbewegen des Cursors
- Wortweises Rückwärtsbewegen des Cursors
- Lösche bis zum Anfang des Wortes
- Lösche Bildschirm

Legen Sie die Datei `.inputrc` in ihrem Homeverzeichnis an mit dem Inhalt:

`C-p: history-search-backward`

Dann können Sie nach dem nächsten bash-Start die letzte Befehlszeile mit einem vorgegebenen Anfang sehr leicht durch das Drücken der Tastenkombination `[Strg]+p` finden.

Beispiel:

```
$ cc -O2 -Wall -pedantic -o test test.c
$ cc[[Strg]+p]
```

ergibt die komplette cc-Zeile. Weitere Features, die das Tippen erleichtern, finden sich in der Manualpage der **bash** unter *Commands for Moving*, *Commands for Manipulating the History*, *Commands for Changing Text*, *Killing and Yanking*, *Completing*.

Eine ähnliche Funktion hat das Ausrufezeichen.

```
$ !cc
```

führt das letzte Kommando aus, das mit `cc` beginnt.

Aufgabe 3 (C Programm)

Das Kommando `date` gibt das aktuelle Datum mit Uhrzeit aus.

Schreiben Sie ein C-Programm, das die gleiche Ausgabe erzeugt.

- a) mit Hilfe von `time()`, gefolgt von `ctime()`
- b) mit Hilfe von `time()`, gefolgt von `localtime()` und `strftime()`

Es sind zwei Compiler auf dem System vorhanden (`cc` und `gcc`).

Überprüfen Sie, dass Ihr Programm mit beiden Compilern funktioniert.

Aufgabe 4 (Kryptoschlüssel erzeugen)

Erzeugen Sie sich einen RSA-Kryptoschlüssel mit 2048 Bit.

```
$ ssh-keygen -t rsa -b 2048
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/export/home_pm/dweber/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

```
Your identification has been saved in /export/home_pm/dweber/.ssh/id_rsa.
```

Your public key has been saved in /export/home_pm/dweber/.ssh/id_rsa.pub.

The key fingerprint is:

3d:96:a1:ab:cf:9a:ff:d6:f2:de:e6:10:d5:60:e5:d4 dweber@stl-s-studwork

Fügen Sie diesen Key zu der Datei

`${HOME}/.ssh/authorized_keys`

hinzu.

Jetzt können Sie sich ohne Passworteingabe zwischen isl-Rechnern und stl-Rechnern einloggen.

Hinweis: Vorgang zum Einloggen in das ISL-Netz von zu Hause aus:

```
$ ssh -l stl-login-name stl-s-stud.htwsaar.de
The authenticity of host 'stl-s-stud.htwsaar.de (134.96.216.212)' can't
be established.
RSA key fingerprint is 00:c3:5b:21:6a:c0:ad:3f:03:37:1c:e0:88:bf:82:7b.
No matching host key fingerprint found in DNS.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'stl-s-stud.htwsaar.de' (RSA) to the list of
known hosts.
Password: *****
Last login: Fri Apr 17 11:03:02 2015 from pd9e08fd4.dip0.
$ ssh isl-c-01
The authenticity of host 'isl-c-01 (134.96.216.81)' can't be
established.
RSA key fingerprint is 04:5d:22:aa:dc:d6:67:27:8e:a7:db:10:2e:03:7e:5e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'isl-c-01,134.96.216.81' (RSA) to the list
of known hosts.
Password: *****
Last login: Fri Apr 17 16:55:49 2015 from :0 FreeBSD 10.1-STABLE
(ISL-C-07) #0 r281529: Tue Apr 14 18:35:24 CEST 2015
stl-login-name@isl-c-01(1)$
```