

Labreport #1

Patrick Eickhoff, Alexander Timmermann

Aufgabe 1

1.1 Arbeiten mit der Shell

1.1.1 `man ls`

Mit dem Befehl `man ls` kann man sich die sog. *man page* des `ls`-Befehls ansehen. Dort werden alle Funktionen des Programmes erläutert und ausführlich dokumentiert.

1.1.2 `ls --help`

Mit dem Befehl `ls --help` kann man sich eine Kurzreferenz des `ls`-Befehls anzeigen lassen. Dort werden die wichtigsten Informationen zusammengefasst und auf Hintergrundinfos verzichtet.

1.1.3 `script`

Der Befehl `script` kann ein Transkript einer Shell-Session speichern. Insbesondere für das Schreiben des Lab Reports ist dies als Notiz sehr hilfreich.

1.2 Benutzerkonten und -verwaltung

1. Mit dem Befehl `adduser labmate` legen wir den User *labmate* an. Mit einer interaktiven Abfrage wird das Passwort des Users gesetzt.
2. Mit den Befehlen `groups` oder `id` lassen sich die Gruppen des Users anzeigen. Zu Beginn befindet sich der User lediglich in einer automatisch erstellten Gruppe, die nach dem Username benannt ist.
3. Mit dem Befehl `addgroup labortests` legen wir die Gruppe *labortests* an.
4. Mit dem Befehl `usermod -a -G labortests labmate` fügen wir den Benutzer *labmate* zur Gruppe *labortests* hinzu.
5. Damit der neue Benutzer den *sudo*-Befehl benutzen darf, muss er zu einer Gruppe hinzugefügt werden, die in der Datei */etc/sudoers* konfiguriert ist. Auf dem Test-System ist dies *admin*. Dies bewerkstelligen wir mit dem Befehl `usermod -a -G admin labmate`.

1.3 Datei- und Rechteverwaltung

1. Mit dem Befehl `sudo -su labmate` öffnen wir eine neue *shell* als Nutzer *labmate*.
2. Mit dem Befehl `cd ~` wechseln wir in das home-Verzeichnis des aktuellen Nutzers. Der Befehl `pwd` zeigt den aktuellen Verzeichnispfad `/home/labmate` an.
3. Mit dem Befehl `mkdir labreports` legen wir das Verzeichnis `/labreports` an.
- 4/5/6. Nachdem wir die Datei `bericht1.txt` mit den Befehlen `touch bericht1.txt` und `pico bericht1.txt` bearbeitet und gespeichert haben, setzen wir die Gruppe dieser Datei mittels `chgrp labortests bericht01.txt` auf *labortests*. Im Anschluss setzen wir mittels `chmod 0660` die Rechte für Besitzer, sowie Teilnehmer der Gruppe auf *rw* (read+write). Dieser Befehl setzt sich aus 4 Oktalzahlen zusammen, die die Rechte für bestimmte User-Gruppen spezifizieren: Einem Special-Attributes-Bit (für unsere Anwendungen irrelevant) und jeweils einem Bit für (v.l.n.r) den Owner, die Group und alle anderen Benutzer. Jedes Bit ist dabei die Summe von Zahlen, die die einzelnen Rechte darstellen: read (4), write (2) und execute (1). Diese Zahlen sind so gewählt, dass jede Summe dieser Zahlen eine eindeutige Rechtekombination ergibt.
8. Nachdem wir mittels `chmod 0660 labortests` die Rechte für dieses Verzeichnis verändert haben, versuchen wir mit dem Befehl `cd labortests` in das Verzeichnis zu wechseln. Dies ist jedoch nicht möglich, da wir keine execute-Rechte für das Verzeichnis besitzen.
9. Das selbe Verhalten beobachten wir, wenn wir versuchen in das Verzeichnis `/root` zu wechseln, da wir kein root-user sind.
10. Zuerst wechseln wir in das Verzeichnis `/opt` und konfigurieren mit folgenden Befehlen das neue Verzeichnis: `mkdir test && chown labmate test && chgrp user labmate`. Mit dem Befehl `chmod 0770 test` geben wir, wie oben beschrieben, dem Besitzer, sowie der Gruppe von *test* Lese-,Schreib- und Ausführungsrechte (*rw*x).
11. Mit dem Befehl `cp /home/labmate/labreports/index.html /opt/test` kopieren wir die Datei `index.html` in das Verzeichnis `/opt/test`.
12. Zuerst setzen wir die Gruppe der Datei `index.html` mit dem Befehl `chgrp user index.html` auf die Gruppe, der nur der Nutzer *user* angehört. Da *labmate* bereits Besitzer der Datei ist, können wir die Rechte einfach mit folgendem Befehl verteilen: `chmod 0750 index.html`.
- 14/15. Mittels `cat` lässt sich als User nun die Datei `index.html` auslesen. Wenn wir jedoch versuchen die Datei mit *nano* zu bearbeiten und zu speichern wird uns der Zugriff verweigert, da wir keine Schreibrechte als *user* haben.
- 16/17. Der Owner der neuen, kopierten Datei `userindex.html` ist jetzt *user*. Also lässt sich diese nun auch als *user* modifizieren, da die Zugriffsrechte der Ursprungsdatei beibehalten werden.

18. Die Datei *index.html* lässt sich als *user* nicht löschen. *user* muss entweder Besitzer der Datei sein oder Lese- und Schreibzugriff auf das übergeordnete Verzeichnis besitzen.

1.4 Administration und Aktualisierung

1. Zuerst sollte man mit dem Befehl `sudo apt-get update` die Indexdateien aller installierten Packages aktualisieren. Auf diese Weise ist das System über neue und geupdatete Packages informiert. Der Befehl `sudo apt-get upgrade` aktualisiert, dann alle Packages auf die neuste Version. Hierbei werden jedoch keine Packages gelöscht. Dies sollte per `sudo apt-get autoremove` vorgenommen werden (siehe `man apt-get`).
2. Das Programm *cowsay* gibt mittels `cowsay [words]` eine Kuh mit Sprechblase und den angegebenen Wörtern auf der Konsole zurück.

1.5 Prozesse und Prozessverwaltung

1. `ps` gibt eine Momentaufnahme der per Optionen spezifizierten Prozesse auf der Konsole aus. `top` hingegen gibt eine regelmäßig aktualisierende Übersicht über alle Prozesse, sowie wie weitere Informationen über CPU-Auslastung, Speicherbedarf, etc., auf der Konsole aus.
3. Beim Ausführen von `cat /dev/urandom` ist in `top` der Prozess *cat* deutlich zu sehen, da dieser viel Speicher verbraucht.
4. Der Prozess *cat* lässt sich nicht als *user* beenden, da der Prozess von *labmate* gestartet wurde. Wir benötigen Super- oder Rootuserrechte um den Prozess mit `kill [Prozess-ID]` beenden zu können.
5. Das System lässt sich nur als sich nur als Superuser neu starten. Deshalb benötigen wir den Befehl `sudo reboot`, um das System als *labmate* neuzustarten.
6. Mit der Software `crontab` lassen sich sogenannte *cronjobs* also Systemaufgaben anlegen. *cronjobs* bestehen aus einer Zeitangabe und den Kommandos, die ausgeführt werden sollen. Die Zeitangabe gliedert sich hierbei in Minuten, Stunden, Tag des Monats, Monat und Tag der Woche auf. Die Liste aller Cronjobs für einen Nutzer lässt sich mittel `crontab -e` editieren. Um alle 5 Minuten einen Zeitstempel an die Datei `/home/labmate/zeitstempel.txt` anzufügen, fügen wir folgenden Befehl zur Liste aller Cronjobs hinzu: `*/5 * * * * date >> /home/labmate/zeitstempel.txt` `*/5` steht für alle 5 Minuten und der Befehl `date >> /home/labmate/zeitstempel.txt` fügt den aktuellen Zeitstempel an die Datei an.

1.6 VMware-Tools installieren

5. Das Installationsskript lässt sich nicht als unprivilegierter Nutzer starten, da wir keine Schreibberechtigungen für relevante Ordner besitzen. Es ist eine Ausführung

mittels `sudo perl vmware-install.pl` nötig.

1.7 Bedienung von VMware

3. Beim Pausieren der VM ist zu beobachten, dass das komplette emulierte System pausiert wird und damit auch alle Prozesse die in unserer VM laufen. Wenn die VM weiterläuft sieht man auch wieder, dass `top` sich aktualisiert.

1.7.1 Snapshots

Snapshots sind Momentaufnahmen der gesamten momentanen VM. Mit diesen Snapshots lässt sich der Zustand der VM zum Zeitpunkt der Aufnahme wiederherstellen. Im Snapshotmanager sind die Snapshots in chronologischer Reihenfolge angeordnet.