1 Quest

1.1 Aufgabe

Eine *Shell* ist eine Kommandosprache, die ein Interface bietet um mit dem Betriebssystem zu kommunizieren.
Einige Shells sind: `bash, sh, csh, tcsh, ksh`

1.2 Aufgabe

Im Folgenden eine kurze Übersicht von einigen Befehlen. Man b eachte, dass die Befehle im Allgemeinen auch andere und/oder mehr Parameter als angegeben akzeptieren. Durch Übergabe mehrerer Parameter, etwa Optionen, können die Funktionalitäten der Befehle auch erweitert oder eingeschränkt werden. Ausführliche Informationen zu den einzelnen Befehlen erhält man normalerweise mit dem Befehl `man [Befehl]`, insbesondere erhält man mit m̂an man eine Anleitung fur den Befehl `man`.

- `man [Befehl]`: Anzeigen von Manuals zu einem Befehl
- `cd [Verzeichnis]`: Wechseln in bestimmte Verzeichnisse
  Bei dem Aufruf von `cd` ohne Parameter oder dem Parameter $\sim$ wechselt man in sein Home-Verzeichnis. Beim Aufruf `cd [Benutzername]` wechselt man in das Home-Verzeichnis des angegebenen Benutzers (sofern dieser Benutzer existiert und man die Rechte besitzt in dieses Verzeichnis zu wechseln). Der Verzeichnispfad kann absolut (`cd /a/b/c` - der einführende Slash ist entscheidend) oder relativ zum aktuellen Verzeichnis (`cd b/c`) angegeben werden.
- `cat [Datei]`: Zusammenfügen von Dateien und Schreiben in die Standardausgabe
- `chmod [Rechte] [Datei]`: Ändern der Rechte von Dateien und Verzeichnissen
  Man kann Rechte zum Beispiel wie folgt zuweisen:
  `chmod xyz`, wobei die erste Ziffer `x` für die Rechte des Besitzers, die zweite `y` fur die der Gruppe, die dritte `z` für die aller anderen Besitzer steht. Die Ziffern werden wie folgt bestimmt:
  - lesen +4 (`w` write)
  - schreiben +2 (`r` read)
  - ausführren +1 (`x` execute)

  Also: Rechte fur Lesen und Ausfuhren etwa: 4+1=5
  Man kann die aktuellen Rechte von Dateien und Verzeichnissen durch den Befehl `ls -l` einsehen. Mögliche Rechteverteilung:

  -rw-r–r- datei.txt
  drwxr-xr-x verzeichnis

  Das erste Feld gibt an ob es sich um eine Datei (`-`) oder ein Verzeichnis (`d`) handelt. Die nächsten drei Felder geben die Rechte des Benutzers an, darauf folgen die Rechte der Gruppe, abschlieend die Rechte aller Benutzer, die weder Benutzer noch Gruppenmitglied sind.
  Beispiel: `chmod 754 test.txt` führt zu der folgenden Rechteverteilung:
  - Besitzer: Lese-, Schreib- und Ausführrechte
  - Gruppe: Lese- und Ausführrechte

  - Andere: Leserecht

  Man beachte: Man erhält immer die Rechte von derjenigen Partei, die am engsten gefasst ist. Beispiel: `----r-xrwx test.txt` → Der Besitzer der Datei hat weder Schreib- noch Lese- noch Ausfuhrrechte. Benutzer, die nicht einmal in der Gruppe der Datei sind, hingegen haben alle Rechte. (Diese Rechteverteilung wird im Normalfall jedoch nicht vorkommen.)

- `cp [Ausgangsdatei] [Zieldatei]`: Kopieren von Dateien und Verzeichnissen
- `ls`: Anzeige aller Dateien und Verzeichnissen im aktuellen Verzeichnis
  Man kann ebenfalls das Verzeichnis bestimmen, von dem man den Inhalt aufgelistet bekommen möchte (`ls [Verzeichnispfad]`. Ohne Parameter werden des weiteren standardmässig nur diejenigen Dateien und Verzeichnisse gezeigt, die nicht mit einem Punkt `.` beginnen. Möchte man diese ebenfalls angezeigt bekommen, so kann man den Befehl `ls -a` verwenden.
- `mkdir [Dateiname]`: Anlegen eines Verzeichnisses
- `more [Datei]`: Ausgabe des Inhalts einer Datei in die Konsole:
  Dabei wird nicht der ganze Text auf einmal ausgegeben, sondern nur so viel, bis der Screen voll ist. Durch Benutzereingabe wird der weitere Text fortlaufend angezeigt.
- `mv [Ausgangsdatei] [Zieldatei]`: Verschieben von Dateien und Verzeichnissen
- `rm [Datei]`: Löschen von Dateien und Verzeichnissen
  Nützlich ist vor allem die Option `-r` (rekursiv), die dazu führt, dass auch Unterverzeichnisse samt Inhalt gelöscht werden.
- `rmdir [Verzeichnis]`: Löschen von (leeren) Verzeichnissen
- `finger`: Anzeigen von Informationen über angemeldete Nutzer
- `date`: Ausgabe des aktuellen Datums (inklusive Uhrzeit)
- `pwd`: Ausgabe des aktuellen Verzeichnisses (von `/` ausgehend)
- `who`: Ausgabe der aktuell eingeloggten Benutzer (ähnlich zu `finger`)
- `find [regulärer Ausdruck]`: Suche von Dateien und Ordnern, die eine bestimmte Zeichenkette enthalten
  Insbesondere gibt es ein Zeichen `*` (asterisc), das durch beliebig viele variable Zeichen ersetzt werden darf. Beispiel: `find *t*.ps` findet alle Dateien, die mit `.ps` enden und vor der Dateiendung ein `t` enthalten, etwa `test.ps, data.ps, script.ps`.
- `grep [Zeichenkette] [Dateien]`: Filtern einer Eingabe durch Angabe einer Zeichenkette, die in den Ausgabeelementen enthalten sein muss
  Insbesondere kann man Zeichenketten innerhalb von Dateien suchen.
- `ps`: Anzeige der momentan laufenden Prozesse
- `kill [Signal] [Prozessnummer]`: Schicken eines Terminieren-Signals an einen Prozess
- `ssh [userhost]`: Secure-Shell
  Programm für Remote Logins (Einloggen in fremde Computer)

1.3 Aufgabe

`touch me [ENTER]`	Anlegen der Datei `me`
`vi me [ENTER]`	Öffnen von `me` mit dem Editor `vi`
`i`	In `vi` in den Eingabemodus wechseln
`Ach haett ich doch ...`	Eingabe des Textes
`[ESC]`	In `vi` in den Befehlsmodus wechseln
`:wq [ENTER]`	Speichern der Datei und Verlassen von `vi`

1.4 Aufgabe

- Login etwa mit `ssh`.
- `cd /opt/SUNWhpc/HPC5.0/examples/mpi`
- `less prime.cc`
- Das Suffix `cc` deutet an, dass es sich um eine Datei handelt, die ein C^{++}-Programm enthält.
- Die Funktion `primeset` hat den Rückgabewert **int**.
- `exit`

1.5 Aufgabe

`|`: Pipeline (Umleitung der Ausgabe eines Jobs in ide Eingabe des nächsten Jobs)
`>`: Umleiten der Ausgabe (in eine Datei, an einen Drucker etc.)

1.6 Aufgabe

C-Programme kann man etwa mit `cc [Dateiname]) oder cc [Dateiname] compilieren.`