

Übungsblatt 4: Shellskripte 2

7.11.2013

Allgemeine Hinweise

- Abgabetermin für die Lösungen ist
 - **Dienstag, 12.11., 10:00** für die Übungsgruppen am Donnerstag und Freitag
 - **Mittwoch, 13.11., 10:00** für die Übungsgruppen am Montag und Dienstag
- Schickt die Lösungen bitte per Email an Euren Tutor.

Aufgabe 4.1: Simulationen und Checkpoints (10 Punkte)

Die einzige Aufgabe auf diesem Blatt ist die Lösung eines Problems aus dem „wahren Leben“. Es geht darum, ein Shellskript zu programmieren, welches tatsächlich nützlich für Wissenschaftler wäre, die Simulationen durchführen.

In `/group/cgl/2013/04/mdrun` liegt ein Programm (tatsächlich ist es selbst ein Shellskript, aber das ist für die Aufgabe egal), das das echte Simulationsprogramm GROMACS¹ simuliert, sowie eine Parameterdatei dafür namens `/group/cgl/2013/04/params.mdp`. Kopiert Euch beide Dateien in ein Unterverzeichnis Eures Heimatverzeichnisses.

Die Simulationssimulation führt man wie folgt aus:

```
./mdrun params.mdp
```

Schaut Euch die Parameterdatei an; sie legt die verschiedenen Parameter einer Molekulardynamiksimulation fest. Insbesondere enthält sie den Parameter `nsteps`, der festlegt, wieviele Schritte der Simulation durchgeführt werden sollen. Nach Ablauf der Simulation erzeugt sie eine Datei namens `result.dat`, die die Ergebnisse der Simulation enthält.

In der Realität kann eine solche Simulation zwischen mehreren Stunden bis zu mehreren Wochen dauern, und ein Wissenschaftler muss möglicherweise hunderte solcher Simulationen laufen lassen. Dabei kann es leicht vorkommen, dass eine Simulation abgebrochen wird, bevor sie fertig ist, beispielsweise, weil der Computer neu gestartet wird, auf dem die Simulation läuft, oder weil ein Programm immer nur einige Stunden am Stück laufen darf, bevor wieder ein anderes Programm die Rechnerzeit in Anspruch nehmen darf. Damit die bis dahin verbrauchte Rechenzeit nicht umsonst war, schreiben Simulationsprogramme in regelmäßigen Abständen sogenannte *Checkpoints*, die dazu verwendet werden können, um eine Simulation neu zu starten. Der Parameter `nstcheckpoint` gibt an, nach jeweils wievielen Simulationsschritten ein solcher Checkpoint erzeugt wird. Wenn man die Simulation dann von einem Checkpoint aus neu starten will, dann muss man den Parameter `init_step` auf den gewünschten Wert setzen.

Ein Lauf unserer Simulationssimulation dauert glücklicherweise nur wenige Sekunden, dennoch bricht auch sie häufig vor dem Ende ab. Dafür erzeugt sie jedoch auch Checkpoints der folgenden Form: `restart_12000.config`. Probiert das Verhalten des Programmes einfach aus. Dummerweise testet `mdrun` nicht, ob bereits ein Checkpoint vorliegt, oder ob die Simulation sogar womöglich bereits fertig

¹<http://www.gromacs.org/>

gelaufen ist. Wenn man `mdrun` einfach nochmal startet, fängt die Simulation wieder von vorne an, solange man nicht den Parameter `init_step` vorher entsprechend ändert. Dies ist ausgesprochen lästig, wenn man viele Simulationsläufe macht, und dann für jeden Lauf schauen muss, ob er bereits fertig ist, und eventuell die Parameterdatei von Hand anpassen muss.

Eure Aufgabe ist es, ein Shellskript zu schreiben, das dem Wissenschaftler hilft. Wenn der Wissenschaftler das Skript mit der Parameterdatei als Argument startet, dann soll es die Simulation starten, falls sie noch nicht gelaufen ist, es soll sie vom letzten Checkpoint aus neu starten, wenn sie bereits gelaufen ist, aber abgebrochen wurde, und es soll nichts tun, wenn die Simulation bereits fertig gelaufen ist.

Hinweise:

- Überlegt Euch erst, was genau das Skript alles tun muss, bevor Ihr anfangt, zu programmieren.
- Das Skript soll als Argument die Parameterdatei übergeben bekommen.
- Das Skript soll mögliche Fehler abfangen, beispielsweise wenn die Parameterdatei nicht existiert, oder wenn keine Parameterdatei übergeben wurde.
- Das Skript soll dem Benutzer mitteilen, was es tut.
- Um einen Parameter aus der Parameterdatei in einem Shellskript zu lesen, könnt Ihr folgendes Codestück verwenden:

```
# grep the line with the parameter from the file
init_step=$(grep init_step $params)
# remove everything up to the space before the number
init_step=${init_step##* }
```

Nach diesem Stück sollte die Variable nur noch den Wert des Parameters enthalten.

- Um in der Shell zwei Zahlen zu vergleichen kann man den Befehl `test` wie folgt verwenden:

```
if test $step -gt $max_step; then
    echo "Whatever"
fi
```

`-gt` steht dabei für „greater than“.