Universität Hamburg, Fachbereich Informatik
Arbeitsbereich Wissenschaftliches Rechnen
Prof. Dr. T. Ludwig, Anna Fuchs, Jannek Squar
Jakob Sachs, Kjell Ehlers, Lieven Petersen

Übungsblatt 1 zur Vorlesung
Hochleistungsrechnen
im WiSe 2022/2023
Abgabe: 22.10.2022, 23:59

Dieses Übungsblatt ist als Einführung in die Benutzung des Clusters und der Programmiersprache C zu verstehen. Im Folgenden sollen Sie sich auf dem Cluster einloggen, das Navigieren in einer Shell üben und erste Programmieraufgaben bearbeiten.

Sollten Probleme auftauchen, wenden Sie sich bitte an die Mailingliste der Vorlesung:

`hr-2223@wr.informatik.uni-hamburg.de`

1 Cluster-Kennung (30 Punkte)

Zur Bearbeitung vieler Übungsaufgaben benötigen Sie ein Konto auf unserem Cluster. Um ein solches zu erhalten, tragen Sie sich zwingend mit korrekten und vollständigen Daten (Vorname, Nachname, E-Mail-Adresse) in die Mailingliste ein:

`https://wr.informatik.uni-hamburg.de/listinfo/hr-2223`

Falls Sie schon angemeldet sind, stellen Sie sicher, dass Sie Ihren Vor- und Nachnamen angegeben haben. Melden Sie sich auf dem Cluster an und machen Sie sich mit den grundlegenden Befehlen vertraut. Informationen dazu finden Sie auf unserer Webseite:

`https://wr.informatik.uni-hamburg.de/teaching/ressourcen/beginners_guide`

Die folgenden konkreten Aufgaben haben Sie zu bewältigen:

1. *Einloggen*

   Loggen Sie sich auf dem Cluster mit ihrem Benutzer und Passwort ein.

2. *Bewegen im CLI (Command Line Interface)*

   a) Machen Sie sich mit der Verwendung von Manual-Pages vertraut:
   `$ man man`

   b) Lassen Sie sich den Namen des aktuellen Arbeitsverzeichnisses anzeigen:
   `$ man pwd`

   c) Lassen Sie sich den Inhalt Ihres Homeverzeichnisses anzeigen:
   `$ man ls`

   d) Erzeugen Sie ein neues Verzeichnis mit dem Namen `testdir`:
   `$ man mkdir`

   e) Ändern Sie das Arbeitsverzeichnis in das neue Verzeichnis:
   `$ cd testdir`

   f) Lassen Sie sich noch einmal das aktuelle Arbeitsverzeichnis anzeigen.

   g) Erzeugen Sie eine leere Datei mit dem Namen `testfile`:
   `$ man touch`

   h) Benennen Sie die neue Datei um in `testfile2`:
   `$ man mv`

i) Kopieren Sie die umbenannte Datei in `testfile3`:
`$ man cp`

j) Löschen Sie die Datei `testfile2`:
`$ man rm`

Frage: Mit `which` können Sie sich den Pfad einer Anwendung anzeigen lassen. Warum funktioniert das nicht für das Kommando `cd`? (Tipp: `man bash`)

3. *Packen eines Archivs*

a) Erstellen Sie ein Verzeichnis mit dem Namen `archiv`.

b) Erzeugen Sie darin eine Datei mit zufälligem Inhalt:
`$ dd if=/dev/urandom of=archiv/zufall bs=1k count=256`

c) Lassen Sie sich die Größe der Datei anzeigen:
`$ ls -lh archiv/zufall`

d) Lassen Sie sich die Größe des Verzeichnisses anzeigen:
`$ ls -ldh archiv`

e) Erzeugen Sie ein `tar`-Archiv, welches das Verzeichnis enthält:
`$ tar -cf archiv.tar archiv`

f) Lassen Sie sich die Größe des Archives `archiv.tar` ausgeben.

Frage: Was fällt Ihnen bezüglich der drei Größen auf?

g) Komprimieren Sie das Archiv:
`$ gzip archiv.tar`

Das Archiv ist nun erstellt. `gzip` hat das Archiv automatisch in `archiv.tar.gz` umbenannt.

h) Lassen Sie sich die Größe des gepackten Archives `archiv.tar.gz` ausgeben.

Frage: Ist es möglich, ein gepacktes Archiv (`.tar.gz`) mit einem Aufruf von `tar` zu erzeugen? Wie hätte dieser Aufruf lauten müssen?

i) Lassen Sie sich den Inhalt des gepackten Archives ausgeben.

2 C-Grundlagen (60 Punkte)

Die Aufgabe dient dem besseren Kennenlernen der Sprache C. Es sollen grundlegende Konstrukte genutzt und geübt werden. In den Materialien finden Sie die Datei `heatmap.c` mit dem dazugehörigen Makefile.

Es soll eine kleine Temperaturkarte der Größe 3×3 simuliert werden. Legen Sie dazu ein globales statisches 3×3-Array mit dem Namen `map` an.

Implementieren Sie die vorgegebene Funktion `set_temperature` so, dass an die übergebene Stelle mit Koordinaten `x` und `y` auf der Karte die entsprechende Temperatur gesetzt wird. Achten Sie dabei auf das adäquate Behandeln ungültiger Eingaben.
Implementieren Sie die Ausgabefunktion `show_map` mittels der Funktion `printf`. Die Ausgabe soll wie folgt aussehen:

```
1 12.35    100.00    20.00
2 300.50   10.02     100.00
3 12.35    420.69    72.82
```

Implementieren Sie die vorgegebene Funktion `set_average_value` so, dass der Durchschnitt der 8 um die übergebenen Koordinaten liegenden Werte berechnet wird. Der Wert an den übergebenen Koordinaten soll nicht mit in diese Berechnung einbezogen werden. Anschließend soll das Ergebnis an die übergebene Stelle mit Koordinaten `x` und `y` auf der Karte geschrieben werden.

3 C-Zeiger (100 Punkte)

In dieser Aufgabe sollen Sie sich mit dem grundlegenden Konzept der Zeiger in C vertraut machen. Dazu laden Sie sich die benötigten Materialien von der Webseite herunter. In der Datei `pointer.c` finden sie einige Funktionen. An einigen Stellen verrät die Ausgabe mittels `printf` das erwartete Ergebnis. An anderen Stellen verraten die Variablennamen oder Kommentare, was gemeint ist. Ihre Aufgabe ist es, die fehlenden Einträge zu vervollständigen, sodass die beschriebene Ausgabe korrekt erfolgt. Beachten Sie: Es darf nichts anderes am Programm geändert werden, außer die mit `TODO` gekennzeichneten Stellen. Das Programm muss am Ende fehler- und warnungsfrei kompilieren und eine semantisch korrekte Ausgabe produzieren.

Abgabe

Notieren Sie in der Datei `antworten.txt` Ihre Antworten zu den drei Fragen. Packen Sie ein komprimiertes Archiv (`.tar.gz`) aus dem sauberen Verzeichnis (ohne Binärdateien) und senden Sie das Archiv an `hr-abgabe@wr.informatik.uni-hamburg.de`.

Um das Archiv zum Versenden auf ihren Rechner zu kopieren, können sie `scp` verwenden. Näheres dazu finden Sie im Beginners' Guide.