Universität Hamburg
Fachbereich Informatik
Wissenschaftliches Rechnen
Prof. T. Ludwig, A. Fuchs, M. Kuhn, J. Weging

Übungsblatt 1 zur Vorlesung
Hochleistungsrechnen
im WiSe 2013/2014
Abgabe: 19.10.2013

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 eine erste Abgabe einer Programmieraufgabe vorbereiten.

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

`hr-1314@wr.informatik.uni-hamburg.de`

Sie können sich unter folgender Adresse in die Mailingliste eintragen:

`http://wr.informatik.uni-hamburg.de/listinfo/hr-1314`

1 Cluster-Kennung (60 Punkte)

Zur Bearbeitung vieler Übungsaufgaben benötigen Sie ein Konto auf unserem Cluster. Um ein solches zu beantragen, schicken Sie eine E-Mail mit Ihrem vollen Namen und Ihrer E-Mail-Adresse in folgendem Format an `michael.kuhn@informatik.uni-hamburg.de`:

Max Mustermann <max.mustermann@domain.tld>

Bitte melden Sie sich auf dem Cluster an und machen Sie sich ein wenig mit den grundlegenden Linux-Befehlen vertraut. Informationen dazu finden Sie auf unserer Webseite im Beginners' Guide:

`http://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: Warum gibt es keine Manual-Page zum Kommando `cd`? (Tipp: `man bash`)

3. *Packen eines Archivs*

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

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

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

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

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

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

Frage: Was fällt Ihnen auf?

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

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

h) Lassen Sie sich die Größe des gepackten Archives `testarchiv.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-Zeiger (80 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 sie Ausgabe mittels `printf`, was herauskommen soll. An anderen Stellen verraten der 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.

3 Makefile (20 Punkte)

Machen Sie sich schlau, was ein Makefile ist. (Tipp: The GNU Make Manual)
Erstellen Sie ein Verzeichnis mit einem Namen wie auf Übungsblatt 0 beschrieben und schreiben Sie in diesem Verzeichnis ein kleines C-Programm (z. B. `helloworld.c`). Fügen Sie in dieses Verzeichnis die Dateien `pointer.c` und `Makefile`. Modifizieren sich das Makefile mittels Pattern-Regeln so, dass durch den Aufruf von `make` in diesem Verzeichnis beide Programme kompiliert werden (Tipp: `make`, `gcc`). Durch den Aufruf von `make clean` soll das Verzeichnis

wieder in den Zustand von vor dem Kompilieren versetzt werden. Ein Tutorial zu Makefiles finden Sie z. B. unter:

`http://www.ijon.de/comp/tutorials/makefile.html`

Abgabe

Erstellen Sie in dem Verzeichnis mit ihren C-Programmen eine Datei `antworten.txt` mit Ihren Antworten zu den drei Fragen. Packen Sie ein komprimiertes Archiv (`.tar.gz`) aus dem sauberen Verzeichnis (ohne Binärdateien).

Senden Sie das Archiv per E-Mail 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 auf unserer Webseite.