

Aufgabenblatt 2

Allgemeine Hinweise:

- Für die Aufgaben auf diesem Übungsblatt müssen Sie am 5.10. votieren. Aufgabe 1 wird hierbei gesammelt angekreuzt, 2a) und b) separat. Aufgabe 2b) ist optional und gibt Bonus-Punkte.
-

Aufgabe 1

Nachdem Sie sich auf dem letzten Übungsblatt mit grundlegenden Funktionen der Shell vertraut gemacht haben, geht es heute um neue Befehle und etwas fortgeschrittenere Features.

Hinweis: Denken Sie daran, dass Sie bei neuen Befehlen die manpage lesen können oder Hilfe zum Befehl im Internet suchen können. Das Finden und Verstehen derartiger Informationen ist eine wichtige Fähigkeit, die Sie trainieren sollten!

- (a) Benutzen Sie die Kommandozeile, um die Datei https://conan.iwr.uni-heidelberg.de/teaching/ipk_ws2018/exampleprogram.tar.xz herunterzuladen. Hierfür können Sie den Befehl

```
1 curl -LO <url>
```

verwenden, wobei Sie `<url>` durch die herunterzuladende Adresse ersetzen.

- (b) Die heruntergeladene Datei ist ein komprimiertes *TAR*-Archiv, ähnlich einer ZIP-Datei unter Windows. Manchmal haben diese Dateien auch `.gz` oder `.bz2` als Endung, dies hängt davon ab, wie die Datei komprimiert wurde.

Überprüfen Sie, ob die Datei wirklich ein *TAR*-Archiv ist. Hierfür können Sie den Befehl

```
1 file <dateiname>
```

verwenden, der die Datei untersucht und das Ergebnis auf der Konsole ausgibt.

- (c) Entpacken Sie die Datei mit dem Befehl

```
1 tar xf <dateiname>
```

Schauen Sie sich an, welche Verzeichnisse und Dateien durch das Entpacken angelegt wurden.

- (d) Kompilieren Sie die im Archiv enthaltene C++-Datei `quadrat.cc` und wählen Sie die Compiler-Optionen so, dass das entstehende Programm `quadrat` heisst. Führen Sie das Programm aus und finden Sie heraus, was es macht.

- (e) Ersetzen Sie `exampleprogram.tar.xz` durch ein neues Archiv, das zusätzlich zu `quadrat.cc` auch das ausführbare Programm enthält. Hierfür wechseln Sie zurück in das Verzeichnis, das das heruntergeladene Archiv enthält und verwenden den Befehl

```
1 tar cfJ <Archivname> <Datei/Verzeichnis>...
```

Dieser Befehl legt ein neues Archiv mit dem angegebenen Namen an und speichert darin alle aufgelisteten Dateien und Verzeichnisse. Letztere werden dabei mit allen enthaltenen Dateien und Unterverzeichnissen hinzugefügt.

- (f) Schauen Sie sich den Inhalt des Archivs mit

```
1 tar tf <Archivname>
```

an. Danach verschieben Sie die Archivdatei in ein neues Verzeichnis (legen Sie ein neues an, falls nötig) und entpacken Sie das verschobene Archiv wieder. Versuchen Sie, das enthaltene Programm `quadrat` auszuführen.

Aufgabe 2

Syntaxfehler und Compiler-Meldungen

In dieser Aufgabe arbeiten wir mit C++-Quellcode, aber Sie müssen diesen nicht wirklich verstehen. Es geht hier primär darum, die Fehlermeldungen des Compilers zu verstehen und die Tipps zum Beheben der Fehler richtig anzuwenden.

- (a) Erstellen Sie die Datei `errors.cc` mit folgendem (absichtlich fehlerhaften) Inhalt:

```
1  #include <iostrea>
2
3  int main(int argc, char** argv)
4      stf::cout << "Typing is difficult" << endl;
5      int ret = 0;
6      return retv
7  }
```

Versuchen Sie, das Programm zu kompilieren. Der Compiler wird diverse Fehlermeldungen ausgeben. Beheben Sie den oder die angezeigten Fehler und starten Sie den Compiler erneut. Dies kann unter Umständen dazu führen, dass der Compiler andere Fehler anzeigt. Machen Sie so lange weiter, bis das Programm übersetzt werden kann.

- (b) **(Fortgeschritten)** Erstellen Sie die Datei `legalbutwrong.cc` mit folgendem Inhalt:

```
1  #include <iostream>
2
3  int main(int argc, char** argv)
4  {
5      int n = 10;
6      // calculate the sum of all numbers from 1 to n
7      int i;
8      int sum = 0;
9      for (int j = 1 ; i <= n ; j = j+1)
10     {
11         sum = sum + j;
12     }
13     std::cout << sum << std::endl;
14     return 0;
15 }
```

Kompilieren Sie das Programm und führen Sie es aus. Falls das Programm “hängt”, können Sie es mit der Tastenkombination “CTRL-C” beenden.

Das Programm ist zwar syntaktisch korrekt, aber es hat einen Fehler. Der Compiler kann Ihnen oft helfen, solche Probleme aufzuspüren. Hierzu müssen Sie ihn anweisen, Ihnen nicht nur Fehler, sondern auch Warnungen anzuzeigen. Sowohl GCC als auch clang benötigen hierfür die Option “-Wall” (warn all). Bei `ipkc++` ist diese Option bereits standardmässig eingeschaltet.

Kompilieren Sie das Programm erneut mit der zusätzlichen Option “-Wall” und beheben Sie die vom Compiler gemeldeten Probleme. Das Programm sollte nun die richtige Lösung (55) ausgeben.