

Informatik für Mathematiker und Physiker **Serie 2** **HS 09**URL: http://www.ti.inf.ethz.ch/ew/courses/Info1_09/

Die erste Aufgabe dieser Serie soll sie mit der Unix resp. Linux Arbeitsumgebung vertraut machen, die wir zum Programmieren verwenden. Sie können die Rechner in einem der ETH Computerräume oder Ihren eigenen Rechner nutzen. Sollten Sie auf Ihrem eigenen Rechner kein Linux verwenden, dann stellen wir eine sogenannte VirtualBox zur Verfügung, mit der Sie Linux emulieren können. Die ausführlichen Anleitungen hierzu finden Sie auf der Kurshomepage. Wenn Sie an einem Linux Rechner (das schliesst die ETH Rechner mit ein) arbeiten wollen, dann konsultieren Sie bitte die Seite http://www.ti.inf.ethz.ch/ew/courses/Info1_09/LinuxTarball.html. Wenn Sie das System VirtualBox nutzen wollen, dann halten Sie sich an die Seite http://www.ti.inf.ethz.ch/ew/courses/Info1_09/VirtualBox.html.

Aufgabe 1 (8 Punkte)

Richten Sie sich ein Linux System ein. Laden Sie die Datei `hello_world.cpp` bei den Materialien zur zweiten Vorlesung herunter und speichern Sie sie in dem Ordner `progs` auf Ihrem Desktop. Wenn Sie die Datei doppelklicken, dann öffnet sich der Texteditor Emacs, und Sie können einfach F9 drücken, um die Datei zu übersetzen. Anschliessend können Sie die ausführbare Datei, die entstanden ist, im Terminal ausführen.

Ändern Sie die Datei `hello_world.cpp`, so dass sie den Anforderungen genügt, die wir an Übungsabgaben stellen (siehe das Merkblatt zur Vorlesung). Insbesondere muss am Anfang der Datei ein Kommentar der Form

```
// Informatik - Serie 2 - Aufgabe 1
// Programm: hello_world.cpp
// Autor: X. M. Plestudent (Gruppe X)
```

stehen. Für diese Aufgabe fügen Sie zusätzlich eine Kommentarzeile der Form

```
// ETH#: 09-123-456
```

ein, in der Sie Ihre ETH-Nummer angeben. Schicken Sie eine E-Mail mit der resultierenden Datei `hello_world.cpp` als *attachment* an Ihre Übungsleiterin oder Ihren Übungsleiter.

Aufgabe 2 (4 Punkte)

Die naivste Methode um herauszufinden, ob eine Zahl $n \geq 2$ prim ist, besteht darin, zu testen, ob n durch irgend eine ganze Zahl im Bereich $2, \dots, n-1$ teilbar ist. Können Sie eine Methode formulieren, die den Primtest schneller löst?

Aufgabe 3 (4 Punkte)

In der Vorlesung haben Sie gesehen, dass man alle Programme in eine eindeutige Reihenfolge bringen kann, (P_0, P_1, P_2, \dots) . Desweiteren haben Sie die Mengen $M(P_i)$ und DIAG kennen gelernt. Gegeben sei nun der Anfang der folgenden Tabelle.

	0	1	2	3	4	5	6	7	8	
M(P ₀)	1	1	0	1	0	1	1	1	1	...
M(P ₁)	0	0	1	1	0	0	1	1	1	
M(P ₂)	1	0	1	0	0	1	1	0	0	
M(P ₃)	1	0	1	1	0	0	0	1	1	
M(P ₄)	0	1	0	1	0	1	0	1	1	
M(P ₅)	0	0	0	0	0	1	1	0	0	
M(P ₆)	0	1	1	0	0	1	0	1	0	
M(P ₇)	1	0	0	0	0	0	0	0	0	
M(P ₈)	0	1	1	1	0	1	1	0	0	
⋮										⋱

Eine 1 (0) steht natürlich dafür, dass das Programm in dieser Zeile, die Zahl in dieser Spalte (nicht) akzeptiert. Geben Sie die Menge $\text{DIAG} \cap \{0, \dots, 8\}$ an. Können Sie noch eine andere Menge als DIAG angeben, die von keinem Programm akzeptiert wird?

Aufgabe 4 (Challenge) (8 Punkte)

Beschreiben Sie eine Methode, mit der die Kreiszahl π in endlicher Zeit bis zu einer beliebigen endlichen Genauigkeit angegeben werden kann.

Aufgabe 5 (Challenge) (8 Punkte)

Wir definieren

$$\mathbb{N}^3 := \{(i, j, k) \mid i, j, k \in \mathbb{N}\}$$

als die Menge aller Tripel (i, j, k) natürlicher Zahlen. Anstelle von i, j und k kann man jede natürliche Zahl einsetzen. Also könnte man sagen, dass $|\mathbb{N}^3| = |\mathbb{N}| \cdot |\mathbb{N}| \cdot |\mathbb{N}| = \infty \cdot \infty \cdot \infty = \infty^3$. Beweisen Sie, dass $|\mathbb{N}^3| = |\mathbb{N}|$, also auch $\infty = \infty^3$ gilt.

Programm: `hello_world.cpp` _____

```
// Program: hello_world.cpp
// Print out "Hello World!"

#include <iostream>

int main()
{
    // output some text
    std::cout << "Hello World!" << std::endl;

    return 0;
}
```

Abgabe: Bis 6. Oktober 2009, 15.15 Uhr.