



Aufgabenblatt 1

letzte Aktualisierung: 23. Oktober, 08:49 Uhr

(47167f2625ea1af4e9252f19263fdd337daaa0af)

Ausgabe: Dienstag, 17.10.2017

Abgabe: Donnerstag, 19.10.2017, 21:59

Thema: Erste Schritte mit C und SVN

1 Abgabemodalitäten

1. Die Aufgaben des C-Kurses bauen aufeinander auf. Versuche daher bitte Deine Lösung noch am gleichen Tag zu bearbeiten und abzugeben.
2. Alle abzugebenden Quelltexte müssen ohne Warnungen und Fehler auf den Rechnern des tubIT/IRB mittels `gcc -std=c99 -Wall` kompilieren.
3. Die Abgabe erfolgt ausschließlich über unser SVN im Abgaben-Ordner.
4. Du kannst bis zur Abgabefrist beliebig oft neue Versionen abgeben.
5. Die Abgabe erfolgt in folgendem Unterordner:
`ckurs-wise1718/Studierende/<L>/<tubIT-Login>/Abgaben/Blatt0<X>`
wobei `<L>` durch den ersten Buchstabe des TUBIT-Logins und `<X>` durch die Nummer des Aufgabenblattes zu ersetzen sind. Die Ordner werden automatisch angelegt sobald die Abgabe freigeschaltet wird.
6. Benutze für alle Abgaben soweit nicht anders angegeben das folgende Namensschema:
`ckurs_blatt0<X>_aufgabe0<Y>.c` wobei `<X>` und `<Y>` entsprechend zu ersetzen sind. Gebe für jede Unteraufgabe genau eine Quellcodedatei ab.
7. Du darfst den Abgabeordner für das Blatt nicht selbst erstellen, das machen wir jeden Morgen kurz nach 8 Uhr!
8. Du musst aber den Befehl `svn up` auf der obersten Verzeichnisebene des Repositories (also in `ckurs-wise1718`) ausführen um alle Änderungen vom Server abzuholen.
9. Im Abgaben-Ordner gelten einige restriktive Regeln. Dort ist nur das Einchecken von Dateien mit den in der Aufgabe vorgegebenen Namen erlaubt, ausserdem werden die Abgabefristen vom Server überwacht. Beachte eventuelle Fehlermeldungen beim SVN-Commit. Lade nur Dateien hoch, die Du selbst bearbeiten sollst, insbesondere also keine Vorgaben.
10. Es gibt einen Ordner 'Arbeitsverzeichnis', in dem Du Dateien für Dich ablegen kannst.
11. Die Ergebnisse der automatischen Tests kannst Du auf OSIRIS einsehen:
<https://teaching.inet.tu-berlin.de/services/osiris-wise1718/>

Registrierungen

Falls Du dich noch nicht bei folgenden Systemen/Stellen registriert hast, solltest Du dies jetzt tun, um am Übungsbetrieb und den Kursen teilnehmen zu können.

tubIT-Account

Adresse: <http://www.tubit.tu-berlin.de/>

Der tubIT-Account ist das zentrale Benutzerkonto an der TU-Berlin. Du benötigst ihn für alle weiteren Schritte bis hin zur Prüfungsanmeldung.

Moses

Adresse: <https://www.moseskonto.tu-berlin.de/>

In Moses registriert Du dich für die Tutorien in den verschiedenen Kursen und kannst Präferenzen für die Zeiten angeben. Du solltest dich umgehend anmelden und Deine Präferenzen angeben, da in der Regel am ersten Mittwoch des neuen Semesters die Registrierung beendet ist.

Achtung: Stelle sicher, dass Du dich nicht versehentlich für die falsche Veranstaltung anmeldest, es gibt mehrere ähnlich klingende Module: Einführung in die *Informatik*, Einführung in die Programmierung *mit Java*, ...

Der korrekte Name lautet **Einführung in die Programmierung (RUE)**.

Die durch Moses vergebenen Tutorien starten erst nach dem C-Kurs, während des C-Kurses vergeben wir Tutorienplätze täglich im Anschluss an die Vorlesung.

ISIS

Adresse: <https://isis.tu-berlin.de/course/view.php?id=10807>

Isis ist die zentrale E-Learning Plattform an der TU-Berlin, hier werden aktuelle Informationen zu unserem Kurs im Nachrichtenforum veröffentlicht und die aktuellen Folien und Arbeitsblätter veröffentlicht. Dieser Kurs ist unter der ID 10807 erreichbar.

OSIRIS

Adresse: <https://teaching.inet.tu-berlin.de/services/osiris-wise1718/>

OSIRIS dient uns zur Koordinierung des Übungsbetriebs dieser Veranstaltung.

Vor allem bekommt ihr hier Feedback zu den Hausaufgaben und könnt die Ergebnisse der automatischen Tests der Abgaben überprüfen. Später im Semester werden hier Arbeitsgruppen gebildet, Tutorien getauscht und hier könnt ihr auch die Ergebnisse der schriftlichen Tests einsehen.

Wichtig: Auch wenn Du dich in OSIRIS für unsere Prüfungsleistungen registriert, ersetzt dies nicht die Anmeldung im Prüfungsamt über Qispos! Auch eine Abmeldung in OSIRIS ist keine Modulabmeldung!

Melde Dich mit Deiner tubIT-Kennung bei der oben genannten Adresse in OSIRIS an und registriere Dich mit vollem Namen und Matrikelnummer (falls Du noch keine Matrikelnummer hast, trage hier "0000" ein). Solltest

du eine temporäre tubIT-Kennung erhalten haben, trage in das Feld für die Matrikelnummer *”vorläufig”* ein. Du kannst diese Eingabe später auf deine *’richtige’* Matrikelnummer ändern.

Schreibe Dich dann in den Kurs *’Einführung in die Programmierung’* ein.

Danach musst Du dich in OSIRIS noch für die Prüfungsleistung *”C-Kurs”* anmelden, um auf das SVN-Repository zugreifen zu können. Was genau das bedeutet erklären wir später.

Nach Ende des C-Kurses werden wir in OSIRIS die Anmeldung für die Hausaufgaben freischalten, die genauso funktioniert wie die Anmeldung zum C-Kurs.

Es fehlt dann noch die eigentliche Prüfungsanmeldung in Form der Modulanmeldung. Diese wird vom Prüfungsamt verwaltet und ist erst später im Semester nötig, wir werden dann darauf hinweisen.

Die Modulanmeldung musst Du beim Prüfungsamt entweder im elektronischen System Qispos oder unter Benutzung von Papierformularen (je nach Studiengang unterschiedlich) durchführen. Falls der Kurs in Deinem Qispos Account nicht zur Anmeldung aufgelistet ist, musst Du vermutlich ein Formular beim Prüfungsamt ausfüllen.

Du kannst in OSIRIS aber unter dem Punkt *’Modulanmeldung’* nachvollziehen, ob wir die Daten von Qispos bzw. den *’gelben Formularen’* übernommen haben und uns Deine Modulanmeldung bekannt ist.

Umgang mit SVN (Rechnerübung)

Die folgenden Aufgaben sollst Du dich mit Subversion (SVN), einem Werkzeug zur Versionsverwaltung und Kollaboration vertraut machen.

In der Softwareentwicklung hat man häufig die Situation, dass Probleme nicht von Einzelnen sondern im Team gelöst werden. Dabei ist es nicht ungewöhnlich, dass die Mitglieder des Teams über mehrere Standorte verteilt gleichzeitig an denselben Projekten arbeiten. Dies kann zum Beispiel große OpenSource Projekte, deren Entwickler über die ganze Welt verteilt arbeiten, vor einige Herausforderungen stellen:

- Wie schafft man es den Überblick zu behalten, wer welche Teile geändert hat?
- Wie vermeidet man Konflikte, wenn zwei oder mehr Personen dieselbe Datei ändern?
- Wer führt diese Änderungen zusammen?
- Wie können fehlerhafte Änderungen rückgängig gemacht werden?
- ...

Viele dieser Probleme lassen sich mit Software zur Versionsverwaltung lösen: Sie protokolliert welche Änderungen wann von wem an welcher Datei erfolgt sind, erlaubt es einfache Konflikte zu lösen und Änderungen zusammenzuführen. Dadurch, dass jede Änderung einzeln protokolliert wird, lässt sich das Projekt ggfs. auf einen vorherigen Status zurücksetzen. Es gibt mittlerweile sehr viele Systeme zur Versionsverwaltung: CVS, Subversion, Git, Mercurial, Bazaar und weitere.

Für diesen Kurs haben wir uns entschieden SVN zu benutzen, da es eine sehr feinteilige Rechteverwaltung ermöglicht.

Diese Aufgabe erklärt die Basisoperationen von SVN. Auf dem nächsten Aufgabenblatt werden wir weitere Funktionen von SVN vorstellen.

Wenn ihr im Rechnerpool des eeCSIT arbeitet, achtet beim Login darauf, dass ihr als Session immer EECS Ubuntu 16 - Desktop Session beim Anmelden auswählt.

Falls ihr auf einem anderen Rechner arbeiten wollt, solltet ihr die Version des SVN-Clients prüfen, da ältere Versionen zu Problemen führen können. Gebt dazu auf der Kommandozeile `svn --version --quiet` ein. Wird dort eine Versionsnummer angezeigt die mit 1.8 oder 1.9 beginnt ist alles in Ordnung.

0.1. (Code) Voraussetzungen Stelle zunächst sicher, dass Du dich in Osiris im Kurs *”Einführung in die Programmierung”* eingeschrieben hast und für die Prüfungsleistung C-Kurs registriert bist. Es kann wenige Minuten dauern, bis die Anmeldung übernommen wird und Du Zugriff auf den SVN-Server hast.

Desweiteren gehen wir davon aus, dass Du mit grundlegenden Unix-Befehlen, wie sie in der *’Rechnereinführung’*¹ vorgestellt wurden, vertraut bist, insbesondere was das Anlegen und Bearbeiten von Textdateien mit einem Texteditor wie Kate angeht.

0.2. (Code) Repository auschecken Wir geben im Folgenden alle Schritte als Befehle auf der Kommandozeile an, so wie sie zum Beispiel auf den Linux-Terminalservern des eeCSIT (früher IRB) (auch mittels SSH-Client von anderen Computern aus) funktionieren sollten.

Wechsel in das Verzeichnis, in dem Du im kommenden Semester die Hausaufgaben bearbeiten willst oder lege eines an. Führe dann folgendes Kommando (in einer Zeile mit Adresse eingeben) aus:

```
svn --username <TUBITLOGIN> checkout  
https://teaching.inet.tu-berlin.de/services/svn/playground-wis1718
```

Wenn Du in einem TU-Pool arbeitest und Dich bereits mit dem tubit-Account angemeldet hast, kannst Du auch kürzer schreiben:

```
svn checkout  
https://teaching.inet.tu-berlin.de/services/svn/playground-wis1718
```

Da Du erstmalig ein Repository von unserem Server auscheckst, bekommst Du möglicherweise folgende Abfrage:

```
Fehler bei der Validierung des Serverzertifikats für  
>>https://teaching.inet.tu-berlin.de:443<<:  
- Das Zertifikat ist nicht von einer vertrauenswürdigen Instanz ausgestellt  
Überprüfen Sie den Fingerabdruck, um das Zertifikat zu validieren!  
Zertifikats-Informationen:  
- Hostname: teaching.inet.tu-berlin.de  
- Gültig: von Oct 13 12:05:24 2015 GMT bis Jan 9 12:05:24 2019 GMT  
- Aussteller: Technische Universitaet Berlin, DE(ca@TU-Berlin.DE)  
- Fingerabdruck: 20:80:94:4A:DC:14:2F:72:39:10:BA:9F:0F:85:02:6F:3C:7D:CB:13  
Ve(r)werfen, (t)emporär akzeptieren oder (p)ermanent akzeptieren?
```

Hier akzeptierst Du das Zertifikat permanent (p eingeben), damit Du nicht bei jedem ein- und auschecken die Zertifikatsanfrage neu beantworten musst. Hinweis: Um den Fingerabdruck zu validieren, vergleiche den hier abgedruckten Fingerabdruck mit dem, der in der Ausgabe erscheint.

Danach wirst Du nach Deinem tubIT Passwort gefragt.

Nun sollst Du eine Liste von Dateien sehen, die vom Server übertragen werden.

0.3. (Code) Repository benutzen Wechsel nun per `cd playground-wis1718` in die lokale Arbeitskopie und lass Dir mit `ls -l` den Inhalt anzeigen.

In dem ausgecheckten Repository findest Du mittlerweile vermutlich eine ganze Reihe unterschiedlicher Dateien und Ordner, da die anderen Studierenden hier seit Beginn des Kurses bereits gearbeitet haben.

Lege in diesem Ordner per `mkdir <Ordnername>` einen neuen Ordner an. Füge den neuen Ordner mittels folgendem Befehl zum SVN hinzu:

```
svn add <Ordnername>
```

¹http://www.eecs.tu-berlin.de/menue/studium_und_lehre/start_ins_studium/1_semester/rechnereinfuehrung/

Achte darauf, dass Du nicht versehentlich Dein Home-Directory oder andere vertrauliche Daten in das Repository einchecken, da alle Dateien für alle Teilnehmerinnen und Teilnehmer lesbar sind! Vermeide Sonderzeichen und Umlaute in Dateinamen, da diese bei anderen Kursteilnehmern möglicherweise zu Problemen führen können. Gib den Ordernamen ohne die <> Zeichen an.

Lege in dem Ordner zwei Dateien an, füge jedoch nur eine von ihnen zur Versionskontrolle hinzu. Schau Dir den Status der beiden Dateien bezüglich der Versionskontrolle an, indem Du in das neue Verzeichnis wechselst und folgendes Kommando ausführst:

```
svn status
```

Sollte das Hinzufügen erfolgreich gewesen sein, solltest Du sehen, dass eine Datei mit einem A (wie Added) gelistet wird, die andere mit einem ?, da sie der Versionskontrolle nicht bekannt ist.

Solltest Du versehentlich beide Dateien mit `svn add` hinzugefügt haben, kannst Du dies mittels `svn revert <Dateiname>` wieder rückgängig machen. `svn revert` wird normalerweise dazu verwendet, alle Änderungen seit dem letzten Commit weg zu werfen. Dateien, die per Add hinzugefügt, aber noch nie committed wurden, werden so zurückgestellt.

Jetzt wird es Zeit, die Datei in das Repository zu übertragen, damit andere Nutzer des Repositories die Datei sehen können. Hierzu dient der Befehl

```
svn commit
```

Unsere Repositories verlangen das Angeben einer Commit Message, die später im Commit Log erscheint und anderen Benutzern (und Dir selbst) erklären soll, was Du in dem Commit getan haben. Die Commit Message gibst Du mit dem Parameter `-m '<text>'` an. Ein Beispielauftrag könnte lauten

```
svn commit -m 'fixed the year 2k bug'
```

Hinweis: Solltest Du keine Commit Message über den `-m` Parameter angeben, wird SVN abhängig von der Konfiguration Deiner Installation einen Editor öffnen. Eine Kurzanleitung, wie man den Standard-Editor auf den eecsIT-Rechnern bedient, findest Du unter².

Erst wenn Du dich für den C-Kurs angemeldet hast, wirst Du für das eigentliche Repository³ freigeschaltet. Dies kann nach der Registrierung bis zu 10-15 Minuten dauern.

Hinweis: Um zu prüfen, ob Du für das C-Kurs Repository freigeschaltet bist, kannst Du mit einem Webbrowser auf die Adresse des Repositories zugreifen und nachsehen, ob Du ein Studierende-Verzeichnis mit Unterverzeichnissen sehen kannst. Findest Du dort ein Verzeichnis mit den Namen Deines tubIT-Logins innerhalb folgender Ordnerhierarchie

```
/Studierende/<ERSTER_BUCHSTABE_TUBIT-LOGIN>/<tuBIT-Login>
```

war die Freischaltung erfolgreich.

Detaillierte Informationen zu den Abgabemodalitäten sowie der SVN Struktur findest Du auf der ISIS Seite dieses Kurses⁴.

In der Zwischenzeit arbeiten wir aber mit dem Playground weiter:

Nun wollen wir eine Liste der Änderungen an dem Playground Repository ansehen. Rufe hierfür

```
svn log
```

auf. Je nachdem wieviele Änderungen es bereits gegeben hat, kann dieses Log sehr lang werden. Durch die ungünstige Sortierung (die ältesten Änderungen sind ganz unten, die neusten Änderungen sind oben) kann es sogar dazu kommen, dass Du Deine eigene Änderung nicht ohne weiteres sehen kannst. Mit `svn log -r<nummer>:<nummer2>` kannst Du nur die Änderungen zwischen zwei Revisionen betrachten. Die aktuelle Revision kann dabei auch über HEAD angesprochen werden. Folglich listet `svn log -r1:HEAD` das Log andersherum auf, sodass aktuelle Änderungen unten angezeigt werden.

Um das Repository zu aktualisieren, kannst Du den Befehl

`svn update` oder die Kurzform `svn up`

verwenden. Insbesondere wenn Du nach dem initialen Auschecken des Repositories ein paar Tage vergehen lässt, solltest Du neue Dateien der anderen Studierenden finden können, da das "playground" Repository allen Studierenden zur Verfügung steht.

Sollten einmal Unklarheiten aufkommen wie SVN benutzt werden kann, kannst Du die Hilfe von SVN öffnen indem Du

```
svn help
```

aufrufst, oder sogar spezielle Hilfe zu einzelnen svn Befehlen erhalten, indem Du

```
svn help <Befehl> benutzt, z.B.: svn help checkout
```

1. Aufgabe: Hallo OSIRIS (Abgabe, unbewertet)

Diese Aufgabe soll Dir helfen, mit OSIRIS⁵ vertraut zu werden und die Ausgaben der Tests zu interpretieren. Die Abgabe Eurer Aufgaben im SVN hat den Vorteil, dass wir einige Tests automatisch ausführen können und Ihr im C-Kurs sofort nach der Abgabe erfahrt ob die Lösung korrekt ist.

Im letzten Abschnitt haben wir den SVN-Playground: `playground-wise1718` benutzt, in dem jeder Teilnehmer und jede Teilnehmerin des Kurses Schreibrechte hatte - und auch alle Dateien lesen konnte. Für die Abgabe von Hausaufgaben ist dies eher weniger geeignet, daher gibt es für den C-Kurs ein eigenes Repository: `ckurs-wise1718`.

Lege also wie in der letzten Aufgabe beschrieben einen neuen Ordner an und checke das Repository dorthin aus. Die Adresse des C-Kurs Repositories lautet:

```
https://teaching.inet.tu-berlin.de/services/svn/ckurs-wise1718
```

Achtung! Der Ordner darf nicht innerhalb des Playground Ordners liegen!

Wenn alles funktioniert hat solltest Du einen neuen Ordner `ckurs-wise1718` finden können. Dort findest Du dann auch Deinen persönlichen Ordner unter:

```
ckurs-wise1718/Studierende/<L>/<tubIT-Login>/
```

Hier musst Du `<L>` durch den Anfangsbuchstaben deines tubIT Login und `<tubIT-Login>` durch deinen ganzen tubIT Login ersetzen (z.B. 'h' und 'hans99'). In diesem Ordner findest Du ein Arbeitsverzeichnis `Arbeitsverzeichnis/` und ein Abgabeverzeichnis `Abgaben/`. Für jedes Blatt legen wir automatisch einen Ordner im Abgabeverzeichnis `Abgaben/Blatt0<X>` an (`<X>` bezeichnet die Blattnummer). Das Arbeitsverzeichnis kannst Du Dir selbst einrichten und zum Ablegen von Dateien für den C-Kurs benutzen. Zusätzlich gibt es noch einen Ordner `ckurs-wise1718/Aufgaben` in dem Du täglich die aktuellen Aufgabenblätter finden wirst.

Wenn Du das SVN erfolgreich auf deinen Computer geladen hast, geht es mit folgenden Schritten weiter:

1. Erzeuge im Unterordner `Blatt01` des Abgabeordners eine Textdatei mit dem Dateinamen `ckurs_blatt01_hallo_osiris.txt`
2. Schreibe in diese Datei einen kleinen Text mit beliebigem Inhalt und füge ihn via commit in in Deinen Abgabeordner im SVN ein.
3. Warte einen Augenblick und besuche die OSIRIS-Seite. Dort findest Du einen Menüpunkt "Automatische Tests". Hier sollte Deine Abgabe erscheinen. Je nach Auslastung des Servers wirst Du entweder sehen, dass die Tests noch nicht gestartet wurden oder wie das Ergebnis des Tests lautet. Für jede Änderung an der Datei die Du eincheckst (d.h. via commit in das SVN einfügst), wird ein neuer Test ausgelöst. Probiere es ruhig aus und beobachte was passiert wenn Du die Übersichtsseite mit den Tests neu lädst.

⁵<https://teaching.inet.tu-berlin.de/services/osiris-wise1718/>

²<http://www.friedhoff.org/survivingvim.html>

³<https://teaching.inet.tu-berlin.de/services/svn/ckurs-wise1718>

⁴<https://isis.tu-berlin.de/course/view.php?id=10807>

Auf der Übersichtsseite findest Du auch einen Link, der Dir nicht nur eine Liste der letzten Tests gibt, sondern eine Übersicht über alle Aufgaben des C-Kurses mit den Ergebnissen der Tests. Wenn Du auf ein Ergebnis klickst, siehst Du auch hier das Protokoll der Tests.

4. Zu jedem der Testläufe gibt es ein Textfeld mit einem kleinen Protokoll. Dort erfährst Du auch, was Du tun musst, damit diese erste Aufgabe bestanden wird. Viel Spaß!

2. Aufgabe: Hallo Welt! (1 Punkt)

Schreibe ein C Programm, welches auf dem Terminal mittels `printf` den Text "Hallo, <Name>!" ausgibt. Ersetze dabei <Name> durch Deinen tubIT-Login. Die Ausgabe soll auf einer separaten Zeile erfolgen.

Eine beispielhafte Benutzung des Programms liefе ab wie in Listing 1 gezeigt:

Listing 1: Kompilieren und Aufruf des Programm

```
> gcc -std=c99 -Wall ckurs_blatt01_aufgabe02.c -o ckurs_blatt01_aufgabe02
> ./ckurs_blatt01_aufgabe02
Hallo, tubITName!
```

Gehe sicher, dass Deine Aufgabe dabei die folgenden Bedingungen erfüllt:

- Die Ein- und Ausgabebibliothek `stdio.h` wird geladen.
- Der Dateiname ist `ckurs_blatt01_aufgabe02.c`.
- Die Funktion `main` ist definiert.
- `printf` wird zur Ausgabe von Text benutzt
- Die Datei kompiliert ohne Fehler und Warnungen beim Aufruf von:
`gcc -std=c99 -Wall ckurs_blatt01_aufgabe02.c -o ckurs_blatt01_aufgabe02`

Checke das Blatt im SVN ein, wie unter "Abgabemodalitäten" beschrieben.