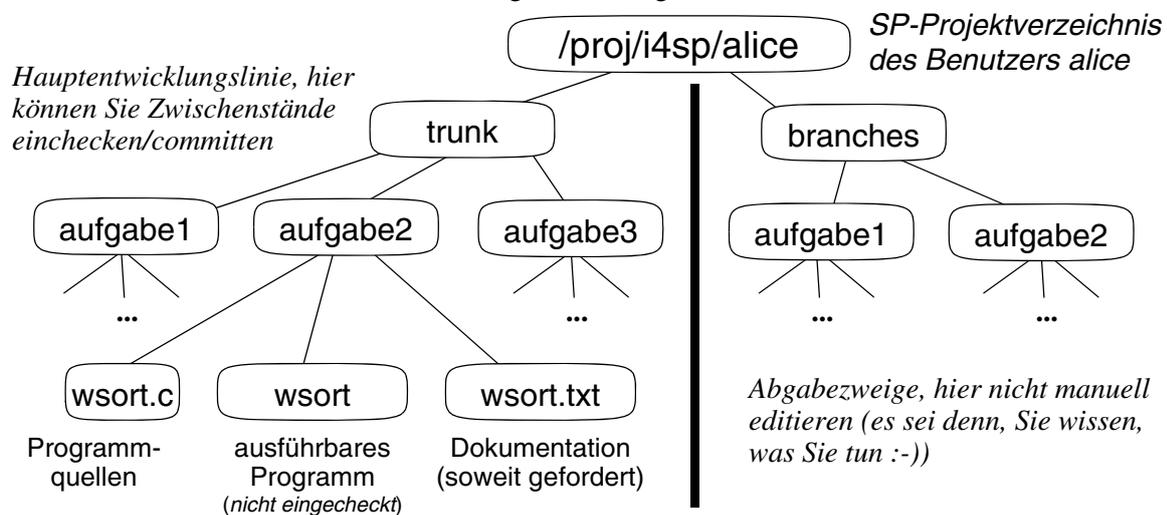


Allgemeine Hinweise zu den SP-Übungen

- Die Aufgaben sind teils alleine, teils in Zweiergruppen zu bearbeiten (siehe Aufgabenstellung). Bei Gruppenarbeit sind Lösungsweg und Programmierung gemeinsam zu erarbeiten.
- Zur Versionsverwaltung wird jedem Teilnehmer ein Subversion-Repository zur Verfügung gestellt, welches auch zur Abgabe verwendet wird. Zum Zugriff auf das Repository muss jeder Teilnehmer zunächst ein Subversion-Passwort durch Aufruf des Kommandos `/proj/i4sp/bin/change-password` setzen.
- Bei Gruppenarbeiten erfolgt die Arbeit im Repository eines Gruppenmitglieds. Dieses muss dem Partner durch Aufruf des Kommandos `/proj/i4sp/bin/set-partner` Zugriff auf den entsprechenden Teil des Repositories ermöglichen. Details hierzu werden in der Übung bekanntgegeben.
- Die Übungsaufgaben müssen spätestens bis zum jeweiligen Abgabetermin abgegeben werden. In darauffolgenden Tafelübungen werden einzelne abgegebene Lösungen besprochen — jeder Übungsteilnehmer muss dabei in der Lage sein, die gesamte Lösung seiner Gruppe zu erläutern. Kann jemand seine Lösung auf Anforderung nicht erläutern, wird für ihn die Aufgabe als nicht abgegeben bewertet (im Zweifelsfall kann hierzu ein Gespräch außerhalb der Tafelübung stattfinden). In den Tafelübungen gilt zur Ermöglichung dieser Maßnahme **Anwesenheitspflicht**.
- Die abgegebenen Programme werden automatisch auf Ähnlichkeit mit anderen Programmen desselben Semesters und früherer Semester überprüft. Hierzu wird ein Dienst der Uni Karlsruhe verwendet, an welchen die abgegebenen Dateien übermittelt werden. Die Dateien sollten daher keine persönlichen Daten enthalten. Werden im Test starke Übereinstimmungen festgestellt, wird die Aufgabe als nicht abgegeben bewertet.
- Kann ein Übungstermin nicht wahrgenommen werden, kann vorher mit den Übungsleitern ein Ersatztermin in einer anderen Übungsgruppe vereinbart werden.
- Jeder Benutzer erhält für SP ein spezielles Projektverzeichnis mit dem Namen `/proj/i4sp/`, gefolgt von dem eigenen Login-Namen. Die Projektverzeichnisse werden für alle im WAFFEL zu einer Tafelübung angemeldeten Teilnehmer nächtlich erstellt. Eine Anmeldung ist zur Übungsteilnahme daher zwingend erforderlich. Der Verzeichnis-Baum für die Aufgaben ist folgendermaßen aufzubauen:



- Unterhalb von `branches` sollten Sie keine manuellen Änderungen durchführen, da dies dazu führen kann, dass das Abgabeprogramm nicht mehr korrekt funktioniert.
- Die Aufgaben sind bis spätestens zum Abgabetermin durch Aufruf des Programms `/proj/i4sp/bin/submit aufgabeX` $X=1 \dots 9$ abzugeben. Dieses Programm kapselt die SVN-Kommandos `copy` und `merge` und überträgt den aktuellen Entwicklungsstand der Aufgabe in den Abgabeweige. Eine Aufgabe kann beliebig oft abgegeben werden; es gilt, soweit nicht anders vereinbart, die letzte rechtzeitige Abgabe.
- Sollten Sie aus triftigem Grund eine fristgerechte Abgabe versäumen, so ist eine verspätete Abgabe durch Aufruf des Abgabeprogramms möglich. Damit eine solche Abgabe bei der Wertung berücksichtigt wird, ist jedoch in jedem Fall eine Rücksprache mit Ihrem Übungsleiter erforderlich. Eine vorhandene rechtzeitige Abgabe wird durch eine verspätete Abgabe nicht gelöscht und im Zweifelsfall gewertet.

Aufgabe 1: lilo (4 Punkte) Bearbeitung einzeln

25.10.2010

Implementieren Sie eine verkettete Liste, welche nicht-negative Ganzzahlen verwaltet. Auf die Liste soll mit den folgenden Funktionen zugegriffen werden:

- `int insertElement(int value)`: Fügt einen Wert in die Liste ein, wenn dieser noch nicht vorhanden ist. Im Erfolgsfall gibt die Funktion den eingefügten Wert zurück, ansonsten den Wert `-1`.
- `int removeElement()`: Entnimmt den ältesten Wert aus der Liste und gibt diesen zurück. Ist der Wert in der Liste nicht vorhanden, wird `-1` zurückgeliefert.

Das Hauptprogramm soll einige Werte in die Liste einfügen und wieder entnehmen. Die folgende Beispiel-Codesequenz

```
printf("insert 47: %d\n", insertElement(47));
printf("insert 11: %d\n", insertElement(11));
printf("insert 23: %d\n", insertElement(23));
printf("insert 11: %d\n", insertElement(11));

printf("remove: %d\n", removeElement());
printf("remove: %d\n", removeElement());
```

soll folgende Ausgabe erzeugen:

```
insert 47: 47
insert 11: 11
insert 23: 23
insert 11: -1
remove: 47
remove: 11
```

Hinweise:

Das C-Programm ist in der Datei **lilo.c** abzulegen. Das Programm muss dem ANSI-C99-Standard entsprechen und mit dem GNU-C-Compiler auf den Linux-Rechnern im CIP-Pool kompilieren. Dazu ist der Compiler mit folgenden Parametern aufzurufen.

```
gcc -std=c99 -pedantic -D_XOPEN_SOURCE=600 -Wall -Werror -o lilo lilo.c
```

Unterprogramme und globale Variablendefinitionen sind ausreichend zu kommentieren. Achten Sie bitte außerdem auf saubere Gliederung des Quellcodes!

Abgabe: bis spätestens Montag, 08.11.2010, 16:00 Uhr