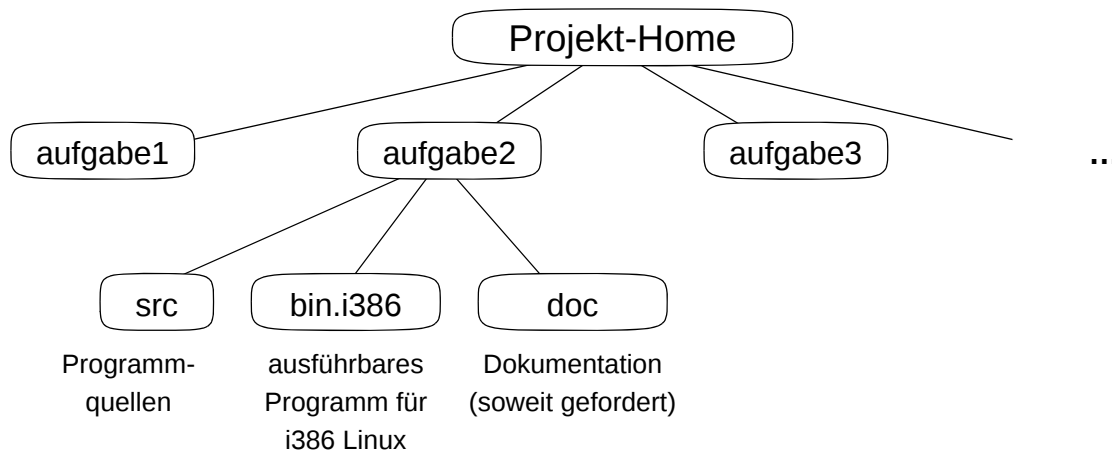


Allgemeine Hinweise zu den Übungen SOS I

- Der Übungsbetrieb der Wiederholerübungen findet in einer Mischform aus Tafel- und Rechnerübung statt, in der je nach Bedarf vom Übungsbetreuer der Stoff erläutert wird oder die Teilnehmer Lösungen am Rechner erarbeiten. Bei Bedarf können die Teilnehmer dazu aufgefordert werden, ihre Lösung an der Tafel zu präsentieren und zu erläutern. Falls ein Teilnehmer seine Abgabe nicht erläutern kann, so wird die Aufgabe als nicht abgegeben bewertet. Zusätzlich werden durch einen Plagiattest die Abgaben mit Abgaben anderer Teilnehmer, auch aus früheren Semestern, verglichen. Bei starker Übereinstimmung wird die Aufgabe ebenfalls als nicht abgegeben gewertet. Hierzu kann im Zweifelsfall ein persönliches Gespräch außerhalb der Übung zur Klärung geführt werden.
- Der Directory-Baum für die Aufgaben ist folgendermaßen aufzubauen:



- Die Teilbäume mit den Übungsaufgaben dürfen nicht für andere Benutzer zugreifbar sein. Die Zugriffsrechte werden automatisch überprüft, bei falschen Zugriffsrechten wird die Abgabe nicht gewertet. Das **Projekt-Home** ist im Verzeichnis `/proj/i4sos/$LOGNAME` zu finden.
- Die Aufgaben sind bis spätestens zum Abgabetermin durch Aufruf des Programms
`/proj/i4sos/pub/abgabe aufgabeX` $X=1 \dots n$
abzugeben. Dieses Programm überprüft die Directory-Struktur und die Namen der Dateien, die laut der Aufgabenstellung vorhanden sein müssen, und erzeugt dann ein Archiv der abgegebenen Dateien. Bis zum Abgabetermin kann ein Programm beliebig oft abgegeben werden — es gilt der letzte (vor dem Abgabetermin vorgenommene) Aufruf des Abgabeprogramms.
- Die Aufgaben der Wiederholerübungen sind alleine und nicht in Gruppen zu bearbeiten.
- In den Wiederholerübungen finden keine Miniklausuren statt.

Aufgabe 1: dtsh (12 Punkte)

Entwerfen und programmieren Sie ein Programm **dtsh** (duct-taped shell), das ähnlich einer einfachen UNIX-Shell zum Starten von Programmen verwendet werden kann.

a) Makefile

Erstellen Sie ein einfaches Makefile. Durch den Aufruf von “make” soll Ihr Program übersetzt werden, mit “make install” das übersetzte Programm aus dem Verzeichnis *src* in das Verzeichnis *bin.i386* kopiert werden und mit “make clean” sollen alle durch den Aufruf von “make” erzeugten Dateien im Verzeichnis *src* gelöscht werden.

b) Prozesserzeugung

Ihre Shell soll dazu das Prompt **dtsh>** ausgeben und auf die Eingabe durch den Benutzer warten. Sobald dieser seine Eingabe mit **Return** bestätigt, wird das angegebene Kommando mit allen Argumenten durch Ihre Shell gestartet (**exec(2)**, **fork(2)**). Die Shell wartet (**wait(2)**) dabei üblicherweise bis das Kommando beendet ist und fragt erst anschließend nach dem nächsten Kommando.

c) Hintergrundprozesse und Signale

Durch die Eingabe von Ctrl-Z (Stop-Signal, SIGTSTP) von der Tastatur soll der gerade im Vordergrund laufende Prozess in den Hintergrund geschickt werden und die Shell mit dem Promptsymbol ein neues Kommando anfordern (**sigaction(2)**). Alternativ dazu kann ein Kommando auch sofort als Hintergrundprozess gestartet werden, indem in der Eingabezeile als letztes Zeichen ein **&**-Zeichen angegeben wird. Ein beendeter Hintergrundprozess (SIGCHLD) soll von der Shell *sofort* aufgeräumt werden (**waitpid(2)**).

d) Umleiten von stdout auf stdin

Nun soll es möglich sein, zwei Programme gleichzeitig zu starten und die Ausgabe *stdout* des ersten Programms als Eingabe *stdin* des zweiten Programms zu verwenden. Die Programme werden dafür durch ein **|**-Zeichen getrennt angegeben. Enthält die Eingabe kein Trennsymbol, so soll die Shell sich wie gewohnt verhalten. Zum Weiterleiten von Daten zwischen zwei Prozessen gibt es unter Unix neben den Sockets noch die einfacheren "Pipes" (**pipe(2)**). Die Dateideskriptoren können mit dem **dup**-Systemaufruf (**dup/dup2(2)**) entsprechend verändert werden.

e) Pipes und Signale

Beim Arbeiten mit Pipes treten neue Signale auf. Beschreiben Sie in einer Datei *pipe.txt* wann welche Signale auftreten. Behandeln Sie diese Signale in Ihrem Programm sinnvoll; die Shell sollte sich nicht beenden.

Hinweise:

- Informationen zu den verwendeten Kommandos bzw. Systemaufrufen finden Sie im Unix-Manual, abrufbar mit dem Kommando **man**, oder in den Folien des Sommersemesters 2007: http://www4.informatik.uni-erlangen.de/Lehre/SS07/V_SOS1/Uebung/
- Beachten Sie bitte auch die Informationen zur Abgabe und den Abgabezeitpunkt. Informationen dazu finden Sie unter http://www4.informatik.uni-erlangen.de/Lehre/WS07/V_SOS1/