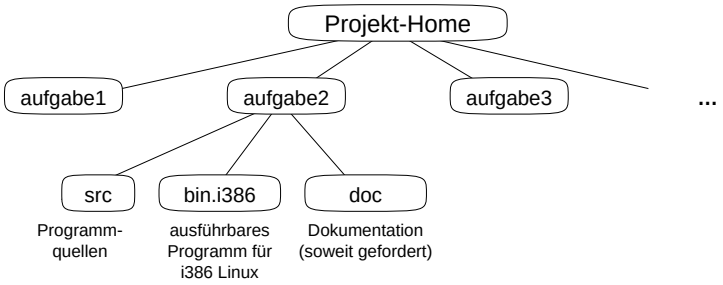
Allgemeine Hinweise zu den Übungen SOS I

- Die Aufgaben werden in der Tafelübungen besprochen, daher besteht Anwesenheitspflicht. Kann jemand seine Lösung auf Anforderung nicht erläutern, wird für sie/ihn die Aufgabe als nicht abgegeben bewertet (im Zweifelsfall kann hierzu ein Gespräch außerhalb der Tafelübung stattfinden). Die abgegebenen Programme werden automatisch auf Ähnlichkeit mit anderen Programmen desselben Semesters oder früherer Semester überprüft. Werden hierbei starke Übereinstimmungen festgestellt, wird die Aufgabe ebenfalls als nicht abgegeben bewertet.
- 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 zu finden.
- Die Aufgaben sind bis spätestens zum Abgabetermin durch Aufruf des Programms
  ~i4sos/pub/abgabe aufgabe***X*** ***X*** =1 ... n
  abzugeben. Dieses Programm überprüft die Directory-Struktur und die Namen der Dateien, die nach 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 vorgenommen Aufruf des Abgabeprogramms.

Bei Fragen zur Aufgabenstellung bitte Mail an *i4sos@informatik.uni-erlangen.de*

Aufgabe 1: dtsh

Entwerfen und programmieren Sie ein Programm **dtsh** (**d**uct-**t**aping **sh**ell), 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) Processerzeugung

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 beendender 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 vom ersten Programm soll als Eingabe für das zweite Programm dienen. 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) Pipe und Signale

Beim Arbeiten mit Pipes treten neue Signale auf. Beschreiben Sie in einer Datei *pipe.txt* wann und welches Signal auftritt. Behandel Sie die Signale in Ihrem Programm sinnvoll, Ihre 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 2006: http://www4.informatik.uni-erlangen.de/Lehre/SS06/V_SOS1/Uebung/
- Beachten Sie bitte auch die Informationen zur Abgabe und den Abgabezeitpunkt. Information dazu findens Sie unter http://www4.informatik.uni-erlangen.de/Lehre/WS06/V_SOS1/