

Aufgabe 3:

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) 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.

b) 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)**).

c) 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.

d) 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.

Allgemeine Hinweise

- Abgabe bis 26.06.2009 14:00