Aufgabe 3:

06.11.2008

smash (12 Punkte) Bearbeitung in Zweier-Gruppen

Implementieren Sie ein Programm smash (**sma**ll **sh**ell), das Programme (im Weiteren als Kommandos bezeichnet) ausführt.

smash soll als Promptsymbol das aktuelle Arbeitsverzeichnis (**getcwd(3)**) gefolgt von einem ":"-Zeichen ausgeben und anschließend eine Zeile von der Standardeingabe einlesen. Die eingelesene Zeile soll in Kommandonamen und Argumente zerlegt werden, als Trennzeichen sollen Leerzeichen und Tabulatoren dienen (**strtok(3)**). Das Kommando soll dann in einem neu erzeugten Prozess (**fork(2)**) mit korrekt übergebenen Argumenten ausgeführt werden (**exec(2)**).

a) Vordergrundprozesse

smash soll anschließend auf das Terminieren der Kommandoausführung warten (**waitpid(2)**) und den Exitstatus ausgeben. Bei der Ausgabe soll unterschieden werden, ob der Prozess sich aktiv selbst beendet hat, oder ob der Prozess durch ein Signal beendet wurde. Die Ausgabe soll wie folgt aussehen:

1. Fall: Prozess beendet sich selbst:

```
/proj/i4sp/mike: ls -l
...
Exitstatus [ls -l] = 0
```

2. Fall: Prozess wird durch ein Signal beendet:

```
/proj/i4sp/mike: ./sp_wait
...
Signal [./sp_wait] = 2
```

Nach der Ausgabe des Exitstatus soll die Shell wieder eine neue Eingabe entgegennehmen. smash soll terminieren, wenn es beim Lesen vom Standardeingabekanal ein End-of-File (CTRL-D) erhält.

b) Hintergrundprozesse

Endet eine Kommandozeile mit dem *Token* '&', so soll das Kommando in einem Hintergrundprozess ausgeführt werden. In diesem Fall soll die Shell nicht auf die Beendigung des Prozesses warten, sondern sofort einen neuen Prompt zur Entgegennahme weiterer Kommandos anzeigen. Jeweils vor Anzeige eines neuen Prompts soll die Shell nun alle zu diesem Zeitpunkt terminierten Hintergrundprozesse aufsammeln und deren Exitstatus wie bei den Vordergrundprozessen ausgeben. Merken Sie sich hierfür beim Erzeugen eines Hintergrundprozesses dessen PID und Kommandozeile in einer verketteten Liste, die in einem Modul **plist.c** die im vorgegebenen Header */proj/i4sp/pub/aufgabe3/plist.h* beschriebene Schnittstelle implementiert. Sie können ggf. Ihre **slist** aus Aufgabe 1 anpassen und nicht mehr benötigte Funktionen entfernen.

c) Verzeichniswechsel

Implementieren Sie nun noch einen Verzeichniswechsel (**chdir(2)**). Wird als Kommando "cd" eingegeben, so soll der smash-Prozess sein Arbeitsverzeichnis auf den im nachfolgenden Argument angegebenen Pfad setzen.

d) Makefile

Erstellen Sie ein zur Aufgabe passendes Makefile.

Abgabe: bis spätestens Donnerstag, 20.11.2008, 14:00 Uhr

Hinweise zur Lösung dieser Aufgabe:

- In */proj/i4sp/pub/aufgabe3* finden Sie die Programme *smash* und *sp_wait* zum Vergleichen bzw. Testen.
- Die Modulschnittstelle von **plist** ist vorgeschrieben und darf nicht verändert werden. Hilfsfunktionen dürfen nicht Teil der Schnittstelle werden (*static*). Ihre Implementierung sollte auch mit unserer Implementierung von **plist** funktionieren, welche wir in */proj/i4sp/pub/aufgabe3/plist.o* zum Testen bereitstellen.