

Aufgabe 7: lash (12 Punkte) Bearbeitung in Zweier-Gruppen

Programmieren Sie basierend auf der smash von Aufgabe 3 eine um Signalbehandlung, Job-Verwaltung und E/A-Umleitung erweiterte Shell: **lash** (**large shell**). Verwenden Sie als Ausgangsbasis dafür bitte unbedingt die Vorgaben in `/proj/i4sp/pub/aufgabe7/`. Die unten gestellten Fragen sind in der Dokumentation (**lash.txt**) zu erläutern.

a) Sofortiges Aufsammeln von Zombieprozessen (RCS Release # 1)

Die smash sammelt angefallene Zombieprozesse jeweils vor der Ausgabe eines Promptsymbols auf. Ändern Sie das Verhalten der Shell so, dass Zombieprozesse direkt nach ihrem Entstehen aufgesammelt werden (**sigaction(2)**). Auch die Ausgabe des Exitstatus und der Kommandozeile soll unmittelbar nach Aufsammeln des Zombieprozesses erfolgen.

Legen Sie ein **Unterverzeichnis für RCS** an und checken Sie diese Version von **lash.c** als Version 1 ein (**ci -u lash.c**). Am Ende jeder weiteren Teilaufgabe soll nun das in Klammern angegebene Release eingchecked werden. Pro Release können beliebig viele Level eingchecked werden - der jeweils aktuellste Level jedes Release repräsentiert die Abgabe der jeweiligen Teilaufgabe.

b) Signalhandler (RCS Release # 2)

Der shell-Prozess soll nun das Interrupt-Signal vom Terminal abfangen (**CTRL+C**). In der Signalbehandlungsfunktion soll nur die Meldung "Interrupt!" auf dem Standardfehlerkanal ausgegeben werden. Was passiert, wenn Ihr lash-Prozess ein Interrupt-Signal erhält und nur ein Vordergrundprozess läuft bzw. wenn auch Hintergrundprozesse laufen (☞ Dokumentation)? Ändern Sie das Programm nun so, dass die Hintergrundprozesse das Signal **SIGINT** ignorieren. Was hat sich dadurch am Verhalten bei einem **CTRL+C** an das Terminal geändert (☞ Dokumentation)?

c) Jobverwaltung (RCS Release # 3)

Implementieren Sie in der Shell ein Kommando **jobs**, das die Kommandozeilen und Prozess-Ids aller laufenden Hintergrundprozesse ausgibt. Geben Sie beim Terminieren eines Hintergrundprozesses auch dessen Exit-Status zusammen mit der Kommandozeile aus. Zur Verwaltung der Hintergrundprozesse (Jobs) verwenden Sie bitte die Implementierung aus `/proj/i4sp/pub/aufgabe7/plist.[ch]` (Makefile anpassen nicht vergessen!).

d) Nebenläufigkeitsprobleme (RCS Release #3 oder bereits in früheren Releases enthalten)

Durch die potentiell nebenläufige Arbeit auf der Jobliste durch Signalbehandlungen und den eigentlichen Programmablauf kann es zu sog. *Race Conditions* kommen. Identifizieren Sie die möglichen Probleme und beschreiben Sie diese zusammen mit Ihren Lösungen zur Vermeidung dieser Probleme in der Dokumentation (**sigprocmask(2)**).

e) Umleitung der Standard Ein- und Ausgabe für Kindprozesse (RCS Release # 3 oder # 4)

Zuletzt soll die Shell noch das Umleiten der Standardein- und -ausgabe erlauben (**dup2(2)**). Die Standardeingabe wird aus einer Datei *file* gelesen, wenn das Token `<file` in der Kommandozeile auftritt, die Standardausgabe wird in eine Datei *file* umgeleitet, wenn das Token `>file` auftritt, wobei *file* angelegt wird, falls dieses noch nicht existiert, und überschrieben wird, wenn es bereits existiert. Die Umleitzeichen `<` bzw. `>` sind hierbei immer mit dem Dateinamen verbunden. Die Token sollen - soweit verwendet - immer am Ende der Kommandozeile in der Reihenfolge `<`, `>` und `&` auftreten.

Hinweis: Wenn Sie die sigaction-Option **SA_RESTART** verwenden möchten, müssen Sie die zusätzliche Compileroption **-D_XOPEN_SOURCE=500** verwenden.

Abgabe: bis spätestens Montag, 22.12.2008, 16:00 Uhr