

Aufgabenblatt 4

Aufgabe - Implementierung Shell

Im Praktikum haben wir bisher die Shell als Benutzerschnittstelle von UNIX verwendet. Der Nutzer kann Kommandos eingeben, die vom Betriebssystem ausgeführt werden. Die Shell hat somit die Funktion eines Kommando-Interpreters. Die verschiedenen Shell-Varianten in UNIX übernehmen dazu weitere Funktionen: Starten und Stoppen von Prozessen, Handhabung von Platzhaltern (Wildcards), Handhabung von Umgebungs-Variablen, Verkettung von Prozessen, Ausführung von Skripten bestehend aus Shell-Kommandos.

Ein (kleiner) Teil dieser Aufgaben soll im Rahmen des Praktikums realisiert werden.

Aufgabenstellung:

Implementieren Sie eine ‚Mini-Shell‘ als C-Programm, das gewisse Aufgaben der Shell übernimmt. Die von Ihnen implementierte Shell soll mindestens die folgenden Anforderungen erfüllen:

1. Das Programm gibt einen Prompt bestehend aus User-Name und aktuellem Directory aus und liest eine Eingabezeile von der Standardeingabe (**stdin**).
2. Das Programm erzeugt per **fork()** einen neuen Kindprozess.
3. Das Programm startet per Funktion **execlp()** oder per **execvp()** das aus der Benutzereingabe extrahierte Programm im Kindprozess.
4. Der Elternprozess wartet in der Zwischenzeit mit **waitpid()** auf das Ende des Kindprozesses.

Randbedingungen:

- Die Kommandos **exit**, **cd** und sowie **set** und **export ...="..."** (Abfragen und Setzen von *Umgebungsvariablen* analog zur Bash) sind keine externen Programme, sondern sollen direkt von der Shell bearbeitet (sog. *Built-in-Kommandos*) werden. Warum ist dies notwendig? Die Built-in-Kommandos müssen keine Optionen unterstützen.
- Kommandos werden in der Regel mit Parametern aufgerufen. Sie müssen also einen rudimentären Parser implementieren.
- (Umgebungs-)Variablen in der Parameterliste sollen aufgelöst werden. Beachten Sie auch, dass Variablen mehrfach in einem Kommando auftreten können.
- Kommandos können teilweise auch Variablen verändern.
- Freiwillige Zusatzaufgabe: implementieren Sie die Verwaltung von *lokalen Variablen*, d.h. die Zuweisung von Variablen mit **...="..."** und den Zugriff auf diese lokalen Variablen in Kommandos.

Hinweise:

- Für die Behandlung von Dateinamen als Argumente ist die Funktion **realpath()** nützlich.
- Die Auflösung von Umgebungsvariablen kann entweder mit dem dritten Parameter, der **main** übergeben wird, oder über die Funktion **getenv()** erfolgen. Zum Setzen von Umgebungsvariablen kann **setenv()** verwendet werden.
- Zum Einlesen und Auswerten der Eingaben benötigen Sie die Standard-C-Funktionen zur Verarbeitung von Zeichenketten (z.B. **sscanf()**, **strtok()**, ...).
- Ihre Shell wird in späteren Aufgaben noch weiterentwickelt! Achten Sie auf ein sauberes Design (Verwendung von Funktionen), so dass Erweiterungen möglich sind. Achten Sie auch auf das Abfangen von Fehlern speziell bei der Prozesserzeugung, da es ansonsten zu nur schwer nachvollziehbaren Phänomenen kommen kann.

Für das Testat ist ein Ausdruck (inkl. Namen der Gruppenmitglieder) des erstellten Skriptes abzugeben.

Testierung: 5.11.2019