Betriebssysteme

Minishell

Praktikum 4

Fachhochschule Bielefeld
Campus Minden
Studiengang Informatik

Beteiligte Personen:

Name	Matrikelnummer
Peter Dick	1050185

---

11. Mai 2016

Inhaltsverzeichnis

Aufgabe 4 - Implementierung Shell

1 Aufgabenstellung

Implementieren Sie eine ”Minden-Mini-Shelläls 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() aud das Ende des Kindprozesses.

Randbedingungen:

- Die Kommandos exit, cd und set sind keine externen Programme, sondern werden direkt von der Shell bearbeitet (Built-in-Kommandos). Warum ist dies notwendig?

- Kommandos können auch Parameter enthalten, Sie müssen also einen rudimentären Parser implementieren

- Umgebungsvariablen in der Parameterliste sollen aufgelöst werden

- Beachten Sie auch, dass Umgebungsvariablen mehrfach in einen Kommando auftreten können

2 Aufgabe 4.1 + Randbedingung 2

2.1 Vorbereitung

C-Projekt anlegen. Makefile schreiben.

2.2 Durchführung

Code schreiben und dann testen bzw debuggen.

2.3 Fazit

Zuerst wird eine Endlosschleife erstellt. Dann wird mit "user = getenv("USER");" der User ermittelt. Der aktuelle Pfad wird genau so wie eben nur mit der Umgebungsvariable PWD ermittelt. Danach wird geprft ob der User und der Pfad ermittelt wurde. Mit hilfe von printf wird der Promt dann ausgegeben. Dann wird die Methode readline aufgerufen. Die Methode readline liest die Zeile mit hilfe der Methode getline ein. Um die Zeile zu parsen wird die Methode parseline Aufgerufen. Die Methode parseline erstellt Tokens mit hilfe der Methode strtok und vordefinierter Seperatoren.

3 Aufgabe 4.2 - 4.4 + Randbedingung 1

3.1 Vorbereitung

keine

3.2 Durchführung

Code schreiben und dann testen bzw debuggen.

3.3 Fazit

Zuerst wird die Methode execute aufgerufen. Dann wird der Prozess mit fork dupliziert. Beim Kind-Prozess ("if (pid == 0)") wird das neue Programm mit execvp geladen. Wenn die pid keiner als null ist dann heist das das Forken fehlgeschlagen ist. Der Eltern-Prozess wated mit waitpid so lange bis der Prozess beendet oder Gekillt wurde. Der Rückgabe wert ist eins wenn nicht der exit-befehl angegeben wurde.

Die befehle exit, cd und werden vor dem Fork abgefangen und als Bulit-in-Komanndos aus gefürht, weil diese wichtig fr die Shell sind. exit beenndet die Shell. cd und set verändern die die Umgebungsvariablen die für diesen Prozess zugeordnet sind.