

Systeme I: Betriebssysteme

Übungsblatt 5 - Praxis & Theorie

Auf diesem Blatt lernen Sie den praktischen Umgang mit der Steuerung von Prozessen und Beispiele für die Kommunikation zwischen Prozessen kennen.

Aufgabe 1 (0.5+0.5+1.5+1.5 Punkte)

Erstellen Sie zunächst eine Datei `counter.sh` mit folgendem Inhalt:

```
counter.sh
1  #!/bin/bash
2  let x=0                # setze Variable x auf 0
3  while true            # Wiederhole endlos
4  do
5      echo $x           # gib Variable x aus
6      let x=x+1         # Zähle x um 1 hoch
7      sleep 1           # Warte 1 Sekunde
8  done
```

Das Skript implementiert einen einfachen Zähler. Machen Sie das Skript mit `chmod u+x counter.sh` ausführbar und führen Sie es mit `./counter.sh` aus.

Öffnen Sie ein weiteres Terminal und bearbeiten Sie die Aufgabe, während `counter.sh` im ersten Terminal weiterläuft.

- a) Der Befehl `ps` zeigt Informationen über die derzeit laufenden Prozesse an. Ohne Optionen zeigt `ps` nur die Prozesse des aktuellen Benutzers an, die auf der aktuellen Textkonsole gestartet wurden.

Welche Optionen muss man beim Aufruf von `ps` benutzen, um eine komplette Liste der derzeit laufenden Prozesse zu erhalten?

- b) In Linux werden *Signale* verwendet, um Prozesse zu steuern und einfache Nachrichten („Daten stehen bereit“, „Fehler aufgetreten“ usw.) zwischen Prozessen auszutauschen. Dazu gehören unter anderem

- SIGSTOP

- SIGCONT
- SIGTERM
- SIGKILL

Wie können Sie den Befehl `kill` verwenden, um diese Signale an den Prozess von `counter.sh` zu senden?

Hinweis 1: Die Prozess-ID (PID) können Sie mit `pidof -x counter.sh` bestimmen oder in der Ausgabe von `ps` ablesen.

Hinweis 2: Um beispielsweise die Manpage `xyz(9)` aufzurufen, verwenden Sie den Befehl `man 9 xyz`.

- c) Beschreiben Sie, was jedes der vier Signale aus Teilaufgabe b) bewirkt. Erklären Sie auch den Unterschied zwischen `SIGTERM` und `SIGKILL`.
- d) Sie möchten `counter.sh` starten, sich vom Rechner abmelden, `counter.sh` aber trotzdem weiter laufen lassen und die Ausgabe nach dem nächsten Einloggen wieder aufrufen. Sobald das Terminal-Fenster geschlossen wird, wird jedoch auch der Prozess beendet. Eine mögliche Lösung besteht darin, das Programm *GNU Screen* zu verwenden. Es wird mit dem Befehl `screen` gestartet. Welche Kommandos und Eingaben werden benötigt, um diese Teilaufgabe mit *GNU Screen* zu lösen?

Hinweis 1: Die englischen Begriffe für das gesuchte Verfahren heißen „detaching“ und „reattaching“.

Hinweis 2: Sollte *GNU Screen* noch nicht auf Ihrem Rechner installiert sein, können Sie das Programm mit dem Befehl `sudo apt-get install screen` installieren. Hierfür benötigen Sie Administratorrechte.

Aufgabe 2 (0.5+2.5 Punkte)

Daten können zwischen Prozessen über sogenannte *Pipes* ausgetauscht werden. Auf der Konsole kann eine Pipe eingerichtet werden, indem zwei Kommandos mit „|“ verknüpft werden. Die Ausgabe des ersten Kommandos wird als Eingabe für das zweite übernommen. Beispiel: `ls -l | less`

Die (Standard-)Eingabe und Ausgabe eines Programms kann mit „<“ bzw. „>“ in eine Datei umgeleitet werden. Wenn die Datei, auf die die Ausgabe eines Programms umgeleitet wird, schon existiert, wird der alte Inhalt der Datei gelöscht und durch den neuen ersetzt.

Beispiel: `echo Hallo > hallo.txt`

Wenn Sie statt „>“ „>>“ verwenden, wird die Ausgabe des Programms an eine existierende Datei angehängt.

Beispiel: `echo Zweite Zeile >> hallo.txt`

- a) Entwerfen Sie eine Kommandozeile, die Ihnen die Anzahl der derzeit auf Ihrem Rechner ausgeführten Prozesse angibt. *Hinweis:* Die Befehle „ps“ aus Aufgabe 1a und „wc“ mit geeigneten Optionen könnten Ihnen von Nutzen sein.

b) Schreiben Sie ein Shell-Skript, das eine Datei `info.txt` nach dem Muster von Abbildung 1 mit folgenden Informationen erzeugt:

- Anzahl aller auf dem Rechner im Moment laufenden Prozesse
- Aktuelles Verzeichnis
- Erster Verzeichniseintrag (Datei oder Unterverzeichnis) im aktuellen Verzeichnis (verwenden Sie dazu den Befehl `head`)
- Auflistung der Dateiendungen aller Dateien im aktuellen Verzeichnis zusammen mit der Anzahl der Dateien mit dieser Endung. Sie können davon ausgehen, dass die Dateinamen keine zusätzlichen Punkte `.` enthalten; Dateinamen ohne Punkt sollen ignoriert werden. Verketteten Sie dazu `ls`, `cut`, `sort` und `uniq` mit jeweils passenden Optionen in dieser Reihenfolge in einer Zeile.

Die Datei soll das in Abbildung 1 gezeigte Layout haben. Bitte kommentieren Sie das Skript ausführlich.

```
Laufende Prozesse: 262
Aktuelles Verzeichnis: /home/osswald/data
Erster Verzeichniseintrag: anbot.pdf
Dateiendungen:
    2 doc
    1 html
    3 jpg
   12 ods
    1 odt
    5 pdf
```

Abbildung 1: Muster für die Datei `info.txt`

Hinweis: Ein Shell-Skript beginnt stets mit dem Kommentar in der ersten Zeile, der die verwendete Skriptsprache angibt, z.B. `#!/bin/bash` (siehe auch das Skript in Aufgabe 1). In den darauffolgenden Zeilen können Sie Befehle bzw. Befehlssequenzen angeben, wie Sie es auf der Shell machen würden. Sie können auch Kommentare in ein Shellskript einfügen; alles nach einem `#` wird bei der Ausführung ignoriert. Um die Datei ausführbar zu machen, setzen Sie mittels `chmod u+x` das Ausführungsrecht.

Aufgabe 3 (5 Punkte)

Über die Dreisam wird eine neue Brücke gebaut. Diese ist zur Zeit nur einspurig befahrbar. Die Brücke ist so kurz, dass sich immer höchstens ein Auto auf ihr befindet.

Sie wurden beauftragt, eine Verkehrsregelung für die Brücke zu entwerfen und überlegen sich folgende Möglichkeiten:

- a) Sie lassen auf jeder Seite der Brücke eine Ampel mit Zeitschaltung aufstellen. Die Schaltzeiten wählen Sie so, dass ein normal fahrendes Auto die Brücke noch vollständig überqueren kann, bevor der entgegenkommende Verkehr grünes Licht erhält.

- b) Sie behaupten, die Brücke sei noch nicht fertig und sperren sie für den Verkehr komplett.
- c) Sie stellen einen Brückenwärter ein, der die Autos über die Brücke fährt. Nur der Brückenwärter darf Autos über die Brücke fahren. Kommen aus beiden Richtungen Autos, so wechselt der Wärter nach jeder Überführung die Richtung ab. Ansonsten kann er auch bei Bedarf über die Brücke zurücklaufen.
- d) Sie stellen an den beiden Enden der Brücke die Schilder  bzw.  auf und legen damit eine Vorrangrichtung fest.
- e) Sie stellen keine Schilder auf und gehen davon aus, dass die Autofahrer vor der Brücke stehen bleiben, wenn sich bereits ein entgegenkommendes Auto auf der Brücke befindet. Fährt ein Auto auf der Brücke voraus, so schließen die nachfolgenden in gleicher Richtung fahrenden Autos auf und fahren unmittelbar danach auf die Brücke.

Nehmen Sie an, jedes Auto entspreche einem eigenen Prozess und die Brücke sei die kritische Region, um welche die Prozesse konkurrieren. In der Vorlesung haben Sie die folgenden vier Anforderungen für den wechselseitigen Ausschluss kennengelernt:

- 1) Keine zwei Prozesse dürfen gleichzeitig in ihren kritischen Regionen sein.
- 2) Es dürfen keine Annahmen über die Geschwindigkeit gemacht werden.
- 3) Kein Prozess darf blockiert werden, wenn alle anderen Prozesse außerhalb ihrer kritischen Regionen laufen.
- 4) Kein Prozess sollte unendlich lange darauf warten müssen, in seine kritische Region einzutreten.

Geben Sie für jeden der Lösungsvorschläge a)–e) an, welche der Kriterien 1)–4) erfüllt bzw. verletzt werden und begründen Sie jede Antwort.

Hinweis: Anforderung zwei beinhaltet normalerweise auch die Anzahl der Rechenkerne. Für diese Aufgabe wurde Regel zwei vereinfacht.

Abgabe: Als PDF-Datei über Ilias bis 30. November 2015, 23:59:00 Uhr.