

## Übungsblatt 9

### Aufgabe 1 (Interprozesskommunikation)

1. Was ist ein kritischer Abschnitt?
2. Was ist eine Race Condition?
3. Warum sind Race Conditions schwierig zu lokalisieren und zu beheben?
4. Wie werden Race Conditions vermieden?

### Aufgabe 2 (Synchronisation)

1. Welchen Vorteil hat Signalisieren und Warten gegenüber aktivem Warten (Warteschleife)?
2. Was ist eine Barriere?
3. Welche beiden Probleme können durch Blockieren entstehen?
4. Was ist der Unterschied zwischen Signalisieren und Blockieren?
5. Welche vier Bedingungen müssen gleichzeitig erfüllt sein, damit ein Deadlock entstehen kann?

<input type="checkbox"/> Rekursive Funktionsaufrufe	<input type="checkbox"/> Anforderung weiterer Betriebsmittel
<input type="checkbox"/> Wechselseitiger Ausschluss	<input type="checkbox"/> > 128 Prozesse im Zustand <b>blockiert</b>
<input type="checkbox"/> Häufige Funktionsaufrufe	<input type="checkbox"/> Iterative Programmierung
<input type="checkbox"/> Geschachtelte <b>for</b> -Schleifen	<input type="checkbox"/> Zyklische Wartebedingung
<input type="checkbox"/> Ununterbrechbarkeit	<input type="checkbox"/> Warteschlangen
6. Kommt es zum Deadlock?  
*Führen Sie die Deadlock-Erkennung mit Matrizen durch.*

$$\text{Ressourcenvektor} = ( 8 \ 6 \ 7 \ 5 )$$

$$\text{Belegungsmatrix} = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 3 & 1 & 0 & 4 \\ 0 & 2 & 1 & 1 \end{bmatrix}$$

$$\text{Anforderungsmatrix} = \begin{bmatrix} 3 & 2 & 4 & 5 \\ 1 & 1 & 2 & 0 \\ 4 & 3 & 5 & 4 \end{bmatrix}$$

## Aufgabe 3 (Shell-Skripte, Datenkompression)

1. Schreiben Sie ein Shell-Skript, das eine Datei `testdaten.txt` erzeugt.
  - Die Datei soll mit Nullen gefüllt werden.
  - Die Nullen liefert die virtuelle Gerätedatei `/dev/zero`.  
(Beispiel: `dd if=/dev/zero of=/pfad/zur/datei bs=512 count=1`)
  - Die Dateigröße soll mindestens 128 und maximal 512 kB sein.
  - Wie groß die Datei wird, soll mit `RANDOM` zufällig festgelegt werden.
2. Schreiben Sie ein Shell-Skript, das als Kommandozeilenargument einen Dateinamen einliest.
  - Die Datei soll das Shell-Skript dahingehend untersuchen, ob es sich um eine Datei, einen Link oder ein Verzeichnis handelt.
  - Wenn es sich um eine Datei handelt, soll der Benutzer mit Hilfe von `select` folgende Auswahlmöglichkeiten haben:
    - 1) ZIP
    - 2) ARJ
    - 3) RAR
    - 4) GZ
    - 5) BZ2
    - 6) Alle
    - 7) Beenden
  - Wählt der Benutzer einen Kompressionsalgorithmus, soll mit diesem die Datei komprimiert werden und der Dateiname entsprechend angepasst werden. Die Dateigröße der originalen und der komprimierten Datei soll das Skript zum Vergleich ausgeben. z.B:  

<code>Testdatei.txt</code>	<code>&lt;Dateigröße&gt;</code>
<code>Testdatei.txt.rar</code>	<code>&lt;Dateigröße&gt;</code>
  - Wählt der Benutzer die Auswahlmöglichkeit (`Alle`), soll das Skript die Datei mit allen Kompressionsalgorithmen komprimieren und die Dateigrößen der originalen und der komprimierten Dateien zum Vergleich ausgeben.  

<code>Testdatei.txt</code>	<code>&lt;Dateigröße&gt;</code>
<code>Testdatei.txt.zip</code>	<code>&lt;Dateigröße&gt;</code>
<code>Testdatei.txt.arj</code>	<code>&lt;Dateigröße&gt;</code>
<code>Testdatei.txt.rar</code>	<code>&lt;Dateigröße&gt;</code>
<code>Testdatei.txt.gz</code>	<code>&lt;Dateigröße&gt;</code>
<code>Testdatei.txt.bz2</code>	<code>&lt;Dateigröße&gt;</code>

3. Testen Sie das Shell-Skript mit der generierten Datei `testdaten.txt`. Was ist das Ergebnis?

## Aufgabe 4 (Shell-Skripte, Datei-Browser)

Schreiben Sie ein Shell-Skript, das via `select` einen Datei-Browser realisiert.

- Die Liste der Dateien und Verzeichnisse im aktuellen Verzeichnis soll ausgegeben und die einzelnen Einträge sollen auswählbar sein.
- Wird eine Datei ausgewählt, soll der Dateiname mit Endung, die Anzahl der Zeichen, Wörter und Zeilen sowie eine Information über den Inhalt der Datei ausgegeben werden. z.B:

```
<Dateiname>.<Dateiendung>  
Zeichen: <Anzahl>  
Zeilen: <Anzahl>  
Wörter: <Anzahl>  
Inhalt: <Angabe>
```

*Informationen zur Anzahl der Zeichen, Wörter und Zeilen einer Datei liefert das Kommando `wc`. Information über den Inhalt einer Datei liefert das Kommando `file`.*

- Wird ein Verzeichnis ausgewählt, soll das Skript in dieses Verzeichnis wechseln und die Dateien und Verzeichnisse im Verzeichnis ausgeben.
- Es soll auch möglich sein, im Verzeichnisbaum nach oben zu gehen (`cd ..`).