Exercise Sheet 9

Aufgabe 1 (Inter-Process Communication)

1. What is a critical section?
2. What is a race condition?
3. Why are race conditions hard to locate and fix?
4. How can race conditions be avoided?

Aufgabe 2 (Synchronization)

1. What is the advantage of signal and wait compared with busy waiting?
2. What is a barrier?
3. Which two problems can arise from blocking?
4. What is the difference between signaling and blocking?
5. Which four conditions must be fulfilled at the same time as precondition that a deadlock can arise?

   □ Recursive function calls
   □ Mutual exclusion
   □ Frequent function calls
   □ Nested `for` loops
   □ No preemption
   □ Hold and wait
   □ > 128 processes in `blocked` state
   □ Iterative programming
   □ Circular wait
   □ Queues

6. Does a deadlock occur?
   Perform the deadlock detection with matrices.

$$\text{Existing resource vektor} = \begin{pmatrix} 8 & 6 & 7 & 5 \end{pmatrix}$$

$$\text{Current allocation matrix} = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 3 & 1 & 0 & 4 \\ 0 & 2 & 1 & 1 \end{bmatrix} \qquad \text{Request matrix} = \begin{bmatrix} 3 & 2 & 4 & 5 \\ 1 & 1 & 2 & 0 \\ 4 & 3 & 5 & 4 \end{bmatrix}$$

Aufgabe 3 (Shell Scripts, Data Compression)

1. Program a shell script, which creates a file `testdata.txt`.
   - The file should be filled with zeros.
   - The zeros provides the virtual device file `/dev/zero`.
     (Examples: `dd if=/dev/zero of=/path/to/file bs=512 count=1`
   - The file size should be at least 128 and 512 kB maximum.
   - How large the file becomes, should be specified randomly via `RANDOM`.
2. Program a shell script, which reads a file name as command line argument.
   - The shell script should check the file to find out if it is a file, a link or a directory.
   - If it is a file, the user should have with `select` these options to choose from:

     ```
     1) ZIP
     2) ARJ
     3) RAR
     4) GZ
     5) BZ2
     6) All
     7) Exit
     ```

   - If the user selects a compression algorithm, the file should be compressed with this compression algorithm and the file name should be adjusted accordingly. The file size of the original file and the file size of the compressed file should be printed out both for comparison reasons. e.g.:

     ```
     testdata.txt        <filesize>
     testdata.txt.rar    <filesize>
     ```

   - If the user selects the option (`All`), the script should compress the file with all compression algorithms and print out the file size of the original file and the file sizes of the compressed files for comparison reasons.

     ```
     testdata.txt        <filesize>
     testdata.txt.zip    <filesize>
     testdata.txt.arj    <filesize>
     testdata.txt.rar    <filesize>
     testdata.txt.gz     <filesize>
     testdata.txt.bz2    <filesize>
     ```

3. Test the shell script with the generated file `testdata.txt`. What is the result?

Aufgabe 4 (Shell Scripts, File Browser)

Program a shell script, which implements a file browser via `select`.

- The list of files and directories in the current directory should be printed out and the individual entries should be selectable.
- If a file is selected, the file name with the extension, the number of characters, words and lines as well as an information about the file content is printed out. e.g.:

  ```
  <Filename>.<Extension>
  Characters: <Number>
  Lines:      <Number>
  Words:      <Number>
  Content:    <Information>
  ```

  Information about the number of characters, words and lines of a file returns the command `wc`*. Information about the contents of a file provides the command* `file`.
- If a directory is selected, the script should navigate into that directory and print out the files and directories in that directory.
- It should also be possible to move up the directory tree into the directory's parent directory (`cd ..`).