

## – Lösung zur Praktikumsaufgabe 7 –

### Thema: *Netzoperationen, fork()*

5.

Listing 1: Beispiellösung der Aufgabe 5.

```
#!/bin/bash
#
# copy all tutorial files via scp into path $1
# demonstrates usage of wget

HOST="http://www.htw-dresden.de"
PRAKPATH=~robge/bsl/prak/

if [ $# -ne 1 ]; then
    echo "Usage: $0 <path>"
    exit 1
fi
if [ ! -w $1 ]; then
    echo "Pfad $1 existiert nicht (oder ist nicht schreibbar)."
    exit 2
fi
if [ ! `which wget` ]; then
    echo "Bitte installieren Sie zunaechst wget."
    exit 3
fi

cd $1
# wget options
# -nd : do not create a hierarchy of directories
# -r : get all files recursively ...
# -ll : ... but only one level deep
# -A : list of file name suffixes (patterns) to be accepted
wget -nd -r -ll -A bsl-prak*.pdf $HOST/$PRAKPATH
exit 0
```

Anmerkungen:

- da wget nicht unbedingt standardmäßig installiert sein muss, sollte man auf seine Existenz zunächst prüfen.
- Es werden zunächst mehr Dateien übertragen, die jedoch danach wieder transparent gelöscht werden.
- Die Datei robots.txt, die Suchhinweise für Webcrawler enthält, lässt sich offenbar nicht vermeiden.
- Die Pfadangabe im URL muss in Anführungszeichen gesetzt werden, ansonsten wird die Tilde lokal durch die Shell expandiert.
- Man kann entweder mit dem Switch -R alle diejenigen Dateien (als Suchmuster)

angeben, die *nicht* mit geladen werden sollen, oder mit dem Switch `-A` explizit alle angeben, die geladen werden sollen (ebenfalls als Suchmuster).

7.

Listing 2: Lösung der Aufgabe 5

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

#define N 3

int main(int argc, char* argv[])
{
    int c, ret;
    pid_t pid[N];

    for (c=0; c<N; c++) {
        pid[c] = fork();
        if (pid[c] == -1) {
            perror("fork");
            exit(EXIT_FAILURE);
        }
        if (pid[c] == 0) { /* Sohn */
            printf("PID des Sohnes %i = %i\n", c+1, getpid());
            exit(EXIT_SUCCESS);
        }
    }

    for (c=0; c<N; c++){
        ret = wait(NULL);
        if (ret == -1) {
            perror("wait");
            exit(EXIT_FAILURE);
        }
    }

    exit(EXIT_SUCCESS);
}
```