M.Sc. Michael Rethfeldt

1 „Hello World“

a. Get used to the commands *man*, *apropos* and *whatis*.
b. Get used to the C development environment in UNIX/Linux and write a program that prints *“Hello World!”* to the terminal.
   - Type “bash” in a new terminal to switch to the bash shell
   - You can use a text editor of your choice (e.g., nano, vim, gedit, …)
   - Navigate to the location of your C program source code file
   - Compile your program with *gcc -o <program name> <source code file name>*
   - Run your program with *./<program name>*

2 The function fork()

a. Explain the following program and run it in Linux.
b. What are the characteristics of the *fork*() function*?*
c. What is the *wait*() function used for? Explain the parameter mapping!

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>

int main(void)
{
      int status;
      pid_t fork_pid;

      if ( (fork_pid=fork() ) == 0 )
      {
            printf("* I am the son. *\n");
            exit(3);
      }
      else if (fork_pid == -1)
      {
            printf("fork() failed!\n");
            exit(2);
      }
      wait(&status);    // wait for termination of son process
      printf("wait status: 0x%x | 0x%x | 0x%x |\n", status,
             (status>>8) & 0xff, status & 0xff);
      return 0;
}
```

Fig. 1: aufgabe2.c

3 Scheduling behavior when using fork()

a. Extend the program from Fig. 1 in such a way that father and son process count to 1,000,000 while printing every 100,000th step to the terminal!
b. Explain the behavior observed!

4 Executing external programs

a. Extend the program from Fig. 1 in such a way that the son process uses the *execl*() function to execute the command "ls -lasi" within the home directory! Perform multiple test runs!
b. What do you observe (note, e.g., the variable *status*)?

5 Printing process parameters

a. Extend the program from task 4 by a printing of process and group identifiers of the running processes!
b. Which functions are responsible for retrieving the identifiers? Use the commands *apropos* and *man*!