Operating Systems
Jacobs University Bremen
Dr. Jürgen Schönwälder

Course: 320202
Date: 2015-02-12
Deadline: 2015-02-18

Problem Sheet #1

Problem 1.1: *vfork() vs. fork()* (1+1 = 2 points)

The `vfork()` variant of `fork()` was briefly discussed in class.

a) Explain how `vfork()` differs from `fork()`.

b) What does the following program return? Explain.

```
#include <sys/types.h>
#include <unistd.h>

int main()
{
    int f = 0;

    if (vfork() == 0) {
        f = 42;
        _exit(0);
    }

    return f;
}
```

Problem 1.2: *parallel* (8 points)

Your task is to write a program `parallel` that can execute commands in parallel. This is best explained by an example. If you want to compress several files (e.g., using gzip), you can run the following shell command:

```
$ gzip *.txt
```

The shell expands the pattern and then a child process is created that compresses all files sequentially. This is, however, not efficient if your computer has multiple cores. With your program `parallel`, one can write the following:

```
$ parallel gzip ::: *.txt
```

The shell again expands the pattern and then `parallel` is invoked. The `parallel` program analyzes the command line looking for the special argument `:::`. If found, `parallel` iterates over the arguments following `:::` and in each iteration it creates a child process executing the command constructed out of the arguments before the special argument `:::` following by the current argument.

To continue the example, lets assume that the current directory contains the files `a.txt`, `b.txt`, and `c.txt`. The `parallel` command shown above then leads to the execution of the three processes `gzip a.txt`, `gzip b.txt`, and `gzip c.txt` under the control of the `parallel` process.

Here is another example. It shows how to compile all .c files in the current directory concurrently:

```
$ parallel gcc -c ::: *.c
```

Bonus points: Implement an option -j that controls how many processes are executed in parallel: `parallel -j 1` will execute all commands sequentially, `parallel -j 3` will execute up to three commands concurrently. (You can earn up to 2 bonus points.)