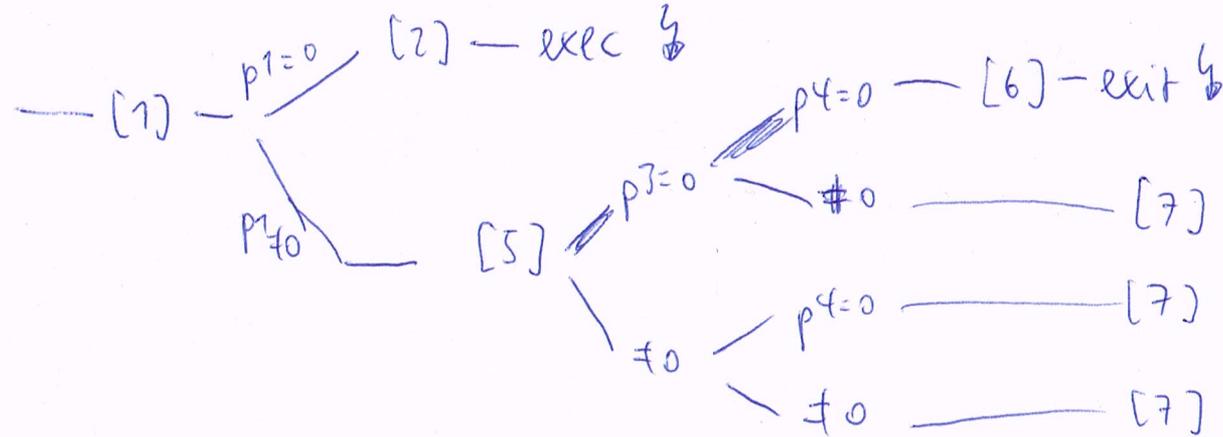


1. Prozessbaum

a) Betrachten Sie das folgende Programm forktest.c (Listing 1) und erstellen Sie (wie in Aufgabe 5 auf Lehrbrief-Seite 43 f., Lehrbrief-Version 08/2022) einen Prozessbaum, aus dem Sie ablesen können, wie oft die Ausgaben [1] bis [7] von den Prozessen erzeugt werden.

```
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#define print(s) printf (s); fflush(stdout);

int main () {
    int pid1,pid2,pid3,pid4;
    print ("[1] Start\n");
    pid1 = fork ();
    if (pid1 == 0) {
        print ("[2] vor dem exec\n");
        execlp ("true", "true", NULL);
        print ("[3] nach dem exec\n");
        pid2 = fork ();
        print ("[4]\n");
    } else {
        print ("[5] zweiter Zweig\n");
        pid3 = fork ();
        pid4 = fork ();
        if (pid3+pid4 == 0) {
            print ("[6] pid-Summe ist 0\n");
            exit (0);
        }
        print ("[7] Ende\n");
    }
}
```



Platz für Prozessbaum-Zeichnung ...

b) Laden Sie die Quellcodedatei forktest.c von folgender URL herunter:

<http://swf.hgesser.de/vb-b1/code/forktest.c>

und übersetzen und testen Sie das Programm mit

```
gcc -o forktest forktest.c
```

```
./forktest
```

Vergleichen Sie dann die Ausgaben mit Ihrem Ergebnis aus Aufgabe a).

Wenn Sie keine Entwicklungsumgebung verfügbar haben, fügen Sie den Code in einem Online-C-Compiler ein, z. B. hier:

<https://www.jdoodle.com/c-online-compiler/>

Falls Sie sich fragen, warum statt printf das Makro print verwendet wird, welches an jede printf-Ausgabe noch einen fflush-Aufruf anhängt, entfernen Sie in der #define-Zeile den fflush-Aufruf und testen im Online-Compiler (der Fehler tritt nur dort auf).