Python

A1: Bäume lassen sich leicht vorraussagen

A2: 0.5: Sehr kleiner baum, da fasst jeder fork() durch exit wieder „vernichtet“ wird

A3: Wie bei 1 nur umgekehrt. Wenn man ale einzelbäume hat lassen sich alle aktionen genau rekonstruieren

A4: Dachte alle children von c werden an den nächsten „verwandten“ also b weitergereicht.

Das ist aber nur mit der „-R“ option der fall

A5: Der finale Baum lässt sich ebenfalls genau bestimmen, in dem man ihn schritt für schritt aufbaut

A6: Sobald einmal exit aufgerufen wurde können die Aktionen nicht rekonstruiert werden.

C

A1: Da jeder prozess sein eigenes „x“ mit wert 100 hat, beeinflussen sie sich nicht gegenseitig

A2: Der fd ist für beide Prozesse gültig, beide Prozesse können in die datei schreiben. Parent schreibt immer zuerst?

A3: ja, mit sleep(), ist aber eher unschön

A4: es hat wohl nach der ersten exec() funktion eine erweiterung gebraucht. Damit alte programme noch funktionieren mussten neue funktionen erstellt werden

A5: wait() gibt die pid des kindprozesses zurück, auf den gewartet wurde. Bei fehler (also zb call in einem (child) prozess ohne eigenes child kommt -1 zurück

A6:waitpid() ist notwendig, wenn auf einen bestimmten kind prozess (von mehreren) gewartet werden soll

A7: child kann nix ausgeben, es wird errno gesetzt, das programm läuft aber weiter

A8: